DoH Deception: Evading ML-Based Tunnel Detection Models with Real-world Adversarial Examples

Emanuel C. A. Valente[1], **André A. Osti** [2], **Lourenço A. P. Júnior** [2], **Júlio C. Estrella**[1]

[1]University of São Paulo (USP)
São Carlos, SP – Brazil

[2]Aeronautics Institute of Technology (ITA)
São José dos Campos, SP – Brazil

emanuel.valente@usp.br, andre.osti@ga.ita.br, ljr@ita.br, jcezar@icmc.usp.br

Abstract. *Previous research on DNS over HTTPS (DoH) tunnel detection has focused on developing detection Machine Learning (ML) models, emphasizing accuracy and explainability. However, these models have neglected the threat of adversarial attacks, rendering them vulnerable and less robust. Our study reveals that most state-of-the-art DoH tunnel detection models are likely susceptible to adversarial black-box attacks. We adopt a novel approach by adapting the Zeroth Order Optimization (ZOO) attack to support DoH request features. The most constrained adaptation generated adversarial examples for 5 out of 6 DoH public tunnel tools. Our methods have successfully evaded the four most used state-of-the-art tunnel detection architectures. The technique relies on network flows and does not depend on the DoH request format. Thus, researchers can use it to create more robust DoH tunnel classifiers that target similar architectures in different security domains.*

1. Introduction

DNS (Domain Name System) is a vital element of the Internet. It improves web user experience and critical services by translating human-readable names into Internet Protocol (IP) addresses necessary for accessing websites and internal services. DNS queries are traditionally transmitted in plain text, exposing user data to potential eavesdropping and manipulation[Wang et al. 2021]. They are susceptible to interception and modification, potentially leading to traditional DNS attacks such as DNS spoofing, DNS tunneling, and privacy breaches. Therefore, the need to secure DNS communications against these implications drove the inception of DoH (DNS over HTTPS) [Borgolte et al. 2019].

In 2018, the IETF (Internet Engineering Task Force) proposed the DoH, a new specification of the DNS, to address the privacy and security issues associated with traditional DNS. This proposal, which provides confidentiality and authenticity by transmitting DNS queries over a secure channel, represents a significant advancement toward a more secure and private Internet infrastructure. However, despite its merits, DoH introduces new challenges and concerns related to DNS tunnels over DoH. DoH tunneling [Lyu et al. 2022] encapsulates DNS queries within HTTPS traffic, effectively carrying the DNS traffic within the encrypted HTTPS channel. These characteristics of the DoH protocol can be leveraged for malicious purposes, such as data exfiltration or command and control (C&C) attacks. In a C&C attack, an attacker establishes a covert communication channel with compromised systems within a target network, allowing them to

control these systems and potentially exfiltrate data remotely. As a result, cybersecurity researchers are increasingly turning to ML techniques to detect and mitigate the risks associated with DoH tunneling.

While the security community has developed sophisticated ML models for detecting DNS over HTTPS (DoH) tunnels with high accuracy [Hynek 2023], these models often lack robustness against adversarial ML attacks. Current research efforts primarily focus on achieving high accuracy and improving explainability [Zebin et al. 2022] in DoH tunnel detection models, emphasizing understanding feature importance rather than enhancing adversarial resilience. Existing research on attacking ML models that rely on network features has primarily focused on ML-based Network Intrusion Detection Systems (NIDS) [Debicha et al. 2023], with limited exploration of realistic adversarial examples [Catillo et al. 2024]. In the domain of DoH security, the emphasis has been mainly on adversarial attacks that manipulate DNS request parameters—such as reducing entropy [Žiža K and Vuletić 2023] rather than leveraging advanced ML adversarial techniques aimed at generating more sophisticated and generalized attacks.

In contrast to previous works on DoH security, this paper focuses on the evasion of DoH tunnel classifiers by examining the vulnerabilities of current state-of-the-art models through the application of ML adversarial attack techniques in realistic scenarios. Specifically, we have adapted the Zeroth Order Optimization (ZOO) attack [Chen et al. 2017], a black-box adversarial method, to incorporate features specific to DoH tunnel tools. This adaptation enables the generation of realistic adversarial examples, which can be used both to explore weaknesses in existing models and to inform the development of more robust defensive strategies.

This paper presents significant contributions to the field of DoH tunnel detection and other security domains, summarized as follows:

- We show that most state-of-the-art models for detecting DoH tunnels are potentially vulnerable to machine-learning black-box adversarial attacks;
- We introduce an attack methodology that generates realistic adversarial examples. This methodology can effectively instrument tunnel tools to attack DoH tunnel classifiers;
- Researchers can utilize our methods to develop more robust DoH tunnel detection models;
- Our method does not depend on the DoH request format, allowing researchers to target similar architectures in different security domains.

The remainder of this paper is structured as follows: Section 2 provides the necessary background information. Section 3 reviews the related works in the field. In Section 4, we detail the methodology, focusing on adapting attack algorithms to support DoH requests. The experimental results are discussed in Section 5. Finally, Section 6 presents the conclusions and outlines directions for future work.

2. Background

To better understand the subsequent sections of this paper, it is essential to define some key concepts and provide corresponding definitions. This section aims to cover the following ideas:

2.1. DNS Over HTTPS (DoH)

DNS over HTTPS (DoH), defined in RFC 8484 [Hoffman and McManus 2018], is an Internet protocol designed to enhance user privacy and security by encrypting DNS queries. Traditional DNS operates in plain text, leaving it vulnerable to eavesdropping and manipulation through attacks like man-in-the-middle. DoH mitigates these risks by transmitting DNS queries over HTTPS and concealing them within regular web traffic.

When a user initiates a DNS request using DoH, their DNS client establishes a secure HTTPS connection with a DoH-compatible resolver. The DNS query and its corresponding response are encapsulated within this encrypted channel, ensuring confidentiality and data integrity. This process appears indistinguishable from other HTTPS traffic on the network, as shown in Figure 1. The adoption of DoH has seen significant growth, with major web browsers incorporating this technology.

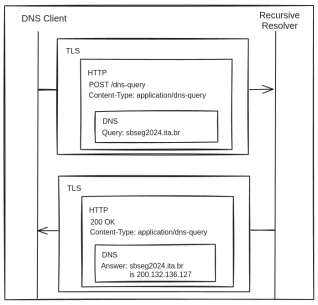

Figure 1. DoH workflow

2.2. DoH tunnels

DNS tunneling is a technique that exploits the DNS protocol to encapsulate non-DNS traffic, such as HTTP, SSH, or any private or malicious binary protocol, within DNS queries and responses [Yassine et al. 2018]. This technique enables data transmission over the Internet, diverging from the original design of the DNS protocol. It can serve legitimate purposes, such as bypassing network restrictions when other forms of Internet access are blocked. It also supports malicious intents, such as data exfiltration or command and control (C&C) communications in cyber attacks. Despite the advent of DoH, these tunneling techniques still apply within the HTTPS protocol.

Figure 2 illustrates a typical DoH tunneling attack flow. First, a malicious user, the attacker, registers a domain and embeds it in malicious software (malware) through obfuscation or encryption. The malware then retrieves the exfiltrated data or the command-and-control (C&C) responses from the target machine and splits it into segments. It adds each segment to a DNS query. Periodically, the malware sends these DNS queries to the DoH resolver server (step 1), which wraps them in HTTPS requests as a DoH payload. The DoH resolver locates the malicious user's registered domain name through recursive DNS queries (step 2). When the authoritative malicious server receives the DNS query containing the exfiltrated data, the malicious user analyzes the DNS query data to extract the intended information (step 3). Finally, the attacker's server can respond with a new malware C&C message encrypted within the DNS response's resource record.

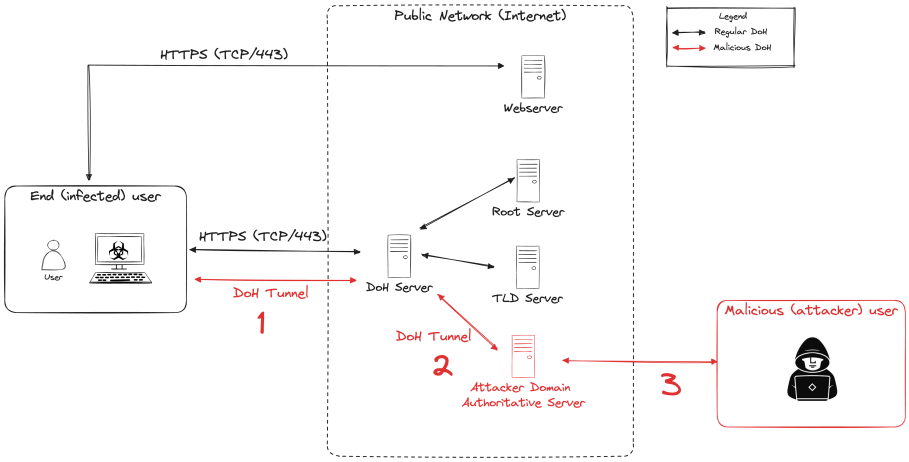

Figure 2. DNS tunnels - technical framework

2.3. ML Adversarial attacks & Adversarial Examples

Adversarial Machine Learning (ML) attacks aim to manipulate ML models by introducing malicious inputs known as adversarial examples or adversarial samples. These examples force the model to make incorrect predictions despite appearing benign to humans. A common approach is taking a correctly classified input (e.g., a dog image) and adding a subtle, human-imperceptible perturbation. While seemingly insignificant to human perception, this perturbation can drastically alter the model's output, leading to misclassification (e.g., classifying the dog as a car).

This described attack is an evasion or test-time attack named for its occurrence during the model's inference stage without impacting its training process. The attacker aims to "evade" the model's correct prediction by introducing a slight, calculated disturbance in the input. In a targeted attack, the attacker manipulates explicitly the input to be misclassified as a predetermined incorrect class. Conversely, in an untargeted attack, the attacker's goal is merely to cause incorrect classification, regardless of the class.

Equation (1) indicates a mathematical definition for an ML Adversarial Attack. It can be stated that for a K-way multiclass classification problem, we define the model as $F_{model} : R^d \rightarrow 1, ..., K$ that maps an input x to a predicted class label y.

$$y = F_{model}(x) \tag{1}$$

Adding a small perturbation δ to the input x, we get a new input x_{adv},

$$x_{adv} = x + \delta \tag{2}$$

This perturbation ensures that the new predicted class label y' differs from the original, as given by Equation (3).

$$y' = F_{model}(x_{adv}) \text{ , such that } y' \neq y \tag{3}$$

The perturbation $x_{adv} = x + \delta$ is named adversarial example. In essence, ad-

versarial examples highlight the need for robustness in ML models. In the scope of this research, adversarial examples are defined as requests or network flows (i.e., features) derived from adversarial attacks aimed at evading DoH tunnel detection models. Examples of such features are in Table 1.

2.3.1. White Box Attacks

A white-box attack assumes complete knowledge of the target model, including its architecture and parameters. Model developers more commonly use this type of attack to perform in-house robustness tests. Modern white-box attacks are formulated as optimization problems. For example, the adversarial ML community frequently uses the method described in [Carlini and Wagner 2017] because of its high effectiveness, flexibility, robustness, and fine-tuning. This method can serve as a general definition of a contemporary white-box machine learning (ML) attack, as represented by Equation (4):

$$J(x, x_{adv}, y_t) = \alpha \cdot Distortion(x, x_{adv}) + \beta \cdot loss(f(x_{adv}), y_t) \tag{4}$$

Here, $J(x, x_{adv}, y_t)$ denotes the objective function to be minimized, where x is the original input and x_{adv} is the perturbed input. The term $Distortion(x, x_{adv})$ measures the perturbation, typically using the L_2 or L_∞ norm. The $loss(f(x_{adv}), y_t)$ represents the misclassification loss of the target model f on the perturbed input with respect to the target class y_t. The parameters α and β are weights that balance the contributions of the two objectives.

2.3.2. Black Box Attacks

A black-box attack assumes that an attacker can only observe the model prediction of a data input (i.e., inference) and knows no other information. The target model is a black-box function. This work will rely on black box attacks, specifically, the Zeroth Order Optimization (ZOO) attack [Chen et al. 2017], based on [Carlini and Wagner 2017]. The goal of the ZOO attack is to minimize the following objective function:

$$J(x, x_{adv}, y_t) = \alpha.Distortion(x, x_{adv}) + c.f(x_{adv}, y_t) \tag{5}$$

Although this research focuses only on benign and malicious DoH requests, we will rely on targeted attacks because untargeted attacks in our primary experiments had a low success rate. Therefore, we can formalize Equation (5) as follows:

$$J(x, x_{adv}, y, c) = \|x - x_{adv}\|_2^2 + c \cdot \max\left(0, \max_{i \neq y}(f_i(x_{adv})) - f_y(x_{adv}) + \kappa\right) \tag{6}$$

In the minimization process, we calculate the constant c using binary search to find an optimal trade-off between minimizing distortion and maximizing attack success. The algorithm uses the Adam or Newton optimizer to calculate the steps for updating

the noise δ by estimating gradients through finite differences and iteratively improving the adversarial example x_{adv}. In practice, the Adam optimizer usually works better with fine-tuned parameters, but Newton is more stable when close to the optimal solution. For this work, we rely on the Adam optimizer for the proposed ZOO adaptions, presented in Section 4.

2.3.3. Unrealistic vs. Realistic Adversarial Examples

In adversarial attacks against ML models, realistic adversarial examples refer to perturbations applied to input data that preserve the plausibility and integrity of the problem space. These perturbations are consistent with real-world data and could plausibly occur within the targeted domain. For instance, in network intrusion detection systems (NIDS), a realistic adversarial example [Catillo et al. 2024] involves modifications to network traffic that continue to resemble legitimate traffic. Such perturbations might include adjustments to packet timings or sizes that do not interfere with the underlying communication protocol but are still capable of misleading the detection model.

Conversely, unrealistic adversarial examples are those perturbations that, although effective in deceiving ML models within the feature space, do not correspond to any plausible or physically possible scenario in the real world. These perturbations can produce alterations that result in impossible or nonsensical features, such as a network packet with a fractional number of packets, a mathematically inconsistent ratio of packet sizes, or even a negative packet time, which cannot exist in actual network traffic. Unrealistic adversarial examples often emerge from traditional feature-space attacks that overlook the constraints of the problem space, thereby generating inputs that could not be produced by any legitimate network agent, such as a network script, tool, or malicious user.

2.4. DoH datasets

The CIRA-CIC-DoHBrw-2020[1][MontazeriShatoori et al. 2020] dataset stands out as the most commonly used dataset for training DoH tunnel classifiers. It considers benign and malicious DoH traffic and non-DoH traffic. The benign DoH traffic comprises DNS resolutions in the top 10,000 websites from the Alexa rankings. On the other hand, malicious DoH traffic comprised tunnels generated by publicly available DNS tunneling tools: *dns2tcp*, *DNSCat2*, and *Iodine*. The researchers used the DoHLyzer[2] tool to extract and analyze the captured DoH traffic for feature extraction. Table 1 lists the 28 possible features from the traffic.

The DoH-Tunnel-Traffic-HKD[3] [Mitsuhashi et al. 2022] dataset incorporates emerging and missing DNS tunnel tools, *dnstt*, *TCP over DNS*, and *tuns*, that previous works overlooked. In 2024, [Niktabe et al. 2024] addressed the imbalance problem of the CIRA-CIC-DoHBrw-2020 dataset to create the BCCC-CIRA-CIC-DoHBrw-2020[4].

Since we aim to include as many public DoH tunnel tools as possible in our

[1]https://www.unb.ca/cic/datasets/dohbrw-2020.html
[2]https://github.com/ahlashkari/DoHLyzer
[3]https://github.com/doh-traffic-dataset/DoH-Tunnel-Traffic-HKD/
[4]https://www.yorku.ca/research/bccc/ucs-technical/cybersecurity-datasets-cds/

Table 1. Extracted Features from Captured Traffic

Param	Feature	Param	Feature
F1	Number of flow bytes sent	F15	Mode Packet Time
F2	Rate of flow bytes sent	F16	Variance of Packet Time
F3	Number of flow bytes received	F17	Standard Deviation of Packet Time
F4	Rate of flow bytes received	F18	Coefficient of Variation of Packet Time
F5	Mean Packet Length	F19	Skew from median Packet Time
F6	Median Packet Length	F20	Skew from mode Packet Time
F7	Mode Packet Length	F21	Mean Request/response time difference
F8	Variance of Packet Length	F22	Median Req/resp time difference
F9	Std. Deviation of Packet Length	F23	Mode Req/resp time difference
F10	Coef. of Variation of Packet Length	F24	Variance of Req/resp time difference
F11	Skew from median Packet Length	F25	Std. Deviation of Req/resp time difference
F12	Skew from mode Packet Length	F26	Coef. of Variation of Req/resp time difference
F13	Mean Packet Time	F27	Skew from median Req/resp time difference
F14	Median Packet Time	F28	Skew from mode Req/resp time difference

research, we will merge both the DoH-Tunnel-Traffic-HKD and BCCC-CIRA-CIC-DoHBrw-2020 datasets into a single aggregated dataset for all the experiments, described in Section 5.

2.5. Target models methods

To test the ZOO adaptations and the quality of the produced adversarial examples in our experiments, instead of considering the actual DoH tunnel models in our experiments, we will rely on the four most used model architectures from the [Hynek 2023] thesis, described in Table 2. For instance, we considered the gradient-boosting ML algorithm in the first target model (TM1). The primary reasons for this approach are as follows:

- **Consistency in Feature Space**: The baseline and the actual models use the same feature space derived from network flows, ensuring that perturbations impact the models similarly;
- **Absence of Defense Mechanisms**: The models under consideration do not incorporate adversarial defenses, simplifying the transfer of attack efficacy from baseline to actual models;
- **Transferability of Attacks**: The attacks with the same architecture tend to be transferable because similar architectures share vulnerabilities [Papernot et al. 2016];
- **Lack of details of the actual DoH tunnel detection models**: The authors must provide more information for the complete reproducibility of the experiments for the target models. Most of the considered models do not offer all the necessary material to reproduce the model.

Adhering to these assumptions allows us to systematically demonstrate the applicability of our adversarial techniques across different model implementations.

3. Related Work

Detecting DNS over HTTPS (DoH) tunnels is critical in cybersecurity. The DoH protocol addresses many vulnerabilities inherent to traditional DNS, such as eavesdropping and

Table 2. Target Models

Target Model	Author	Method	Dataset
TM_1	[Singh and Roy 2020]	GB	D
TM_2	[Alenezi and Ludwig 2021]	XGB	D
TM_3	[Zebin et al. 2022]	BS	D
TM_4	[Zhan et al. 2022]	RF	C

Model methods: GB - Gradient Boosting, XGB - Extreme Gradient Boosting, BS - Balanced Stacked, RF - Random Forest. Datasets: D - CIRA-CIC-DoHBrw-2020, C - Custom.

tampering. However, identifying malicious uses of this protocol remains a significant challenge, as attackers exploit DNS to scan networks, track end users, and exfiltrate data through command and control techniques [Toulas 2024]. The scientific community has actively developed machine learning (ML) models to enhance the accuracy and efficiency of DoH tunnel detection, especially in distinguishing benign DNS traffic encapsulated within HTTPS.

A comprehensive survey, [Wang et al. 2021], discusses DNS tunnel detection techniques developed between 2006 and 2020, covering rule-based and ML model-based methods. This work served as a baseline for further ML model-based works in DNS tunnel detection and provided the background foundation for developing our work. As adversarial attacks were still in development when these studies were presented, they did not consider adversarial examples in their design. Consequently, all the DoH tunnel classifiers studied by Wang are vulnerable to adversarial attacks, such as evasion attacks.

Hynek's PhD thesis [Hynek 2023] examines the security implications of encrypted DNS, demonstrating that ML can detect DoH traffic with very high accuracy. His work highlights recent studies in ML-based DoH tunnel detection. For this research, we use these studies to build the target models' architecture, selecting the four most commonly used ML architectures, as explained in Section 2.5. Although all the DoH classifiers presented high accuracy, they did not consider adversarial examples during the model lifecycle, making them vulnerable to ML attacks.

[Goodfellow et al. 2015] introduced the concept of adversarial examples, demonstrating how small perturbations to input data can significantly evade deep learning models. Their work underscores the importance of understanding adversarial vulnerabilities across various model types, including traditional ML algorithms. Building on this, [Papernot et al. 2016] explored the transferability of adversarial examples, showing that attacks on one model can often be successfully applied to others. [Szegedy et al. 2014] further examined the unique properties of neural networks in the context of adversarial examples, offering foundational insights into how these attacks exploit model weaknesses. While adversarial ML has been considered in the context of ML-based Network Intrusion Detection Systems (NIDS) [Debicha et al. 2023], similar techniques have not yet been extensively applied to DoH tunnel detection models.

The study by [Sánchez Sánchez et al. 2024] offers substantial insights into the challenges and potential solutions associated with vulnerabilities in ML models. Their

research, which primarily addresses the robustness of IoT device identification systems against ML evasion attacks, parallels our work in several aspects. Notably, their examination of the [Carlini and Wagner 2017] attack using the L2 norm revealed a high success rate in evading identification models, underscoring the effectiveness of this method in compromising security classifiers. However, given that the Carlini and Wagner attack is inherently a white-box attack, it does not align with our goal of reproducing real-world scenarios, where attackers typically lack access to model internals. Consequently, our methodology employs the Zeroth Order Optimization (ZOO) attack. ZOO builds on the foundational principles of the C&W attack, particularly in its objective function and strategy for generating adversarial examples. The ZOO attack is designed explicitly for black-box settings, as Section 2.3.2 explains.

The pioneering study by [Žiža K and Vuletić 2023] investigated the impact of modifying DNS exfiltration tool parameters on detection accuracy. The authors successfully deceived classifiers by generating DoH tunnel exfiltration requests with reduced character entropy. However, this approach has several limitations. It primarily targets traditional DNS tunnels, which operate over an unencrypted channel, rather than DoH, which utilizes an encrypted channel. Additionally, the method does not leverage network features crucial for identifying DoH tunnels in ML models, limiting its applicability to more advanced DoH detection scenarios.

Our primary goal is to highlight the risk that state-of-the-art DoH tunnel classifiers are likely susceptible to adversarial attacks. The second goal is to address the discussed limitations of poor application of ML adversarial methods by adapting ML adversarial techniques to support security domain problems, especially DoH tunnel classifiers. Our research adapts the Zeroth Order Optimization (ZOO) attack, a black-box adversarial attack, to include DoH requests from DoH tunnel tools as input for the adapted algorithm. This approach generates realistic adversarial examples, which means that researchers can:

1. Use these examples to produce more robust models;
2. Enable DoH tunnel tools to evade DoH tunnel classifiers more effectively;
3. Apply our methodology to other security domains for testing against adversarial examples during the model lifecycle.

4. Methodology

This section details our research methodology. It involves adapting the Zeroth Order Optimization (ZOO) attack to support DoH request features. We propose three adaptations to increase the constraints in the ZOO attack to support more realistic scenarios, such as perturbing specific DoH tunnel tools' feature limits while keeping others unmodified. Figure 3 illustrates the details of our methodology. In the following subsections, we will describe our method.

4.1. ZOO-DoH

ZOO-DoH represents the first version of the ZOO Attack adaptation. Unlike the original ZOO algorithm, which uses images as input, this adapted version uses features extracted from network flows by the DoHLyzer tool, as described in Table 1. We also made internal modifications to remove the image optimizations designed to attack Deep Neural Networks (DNNs). Therefore, the input to the objective function remains as shown in

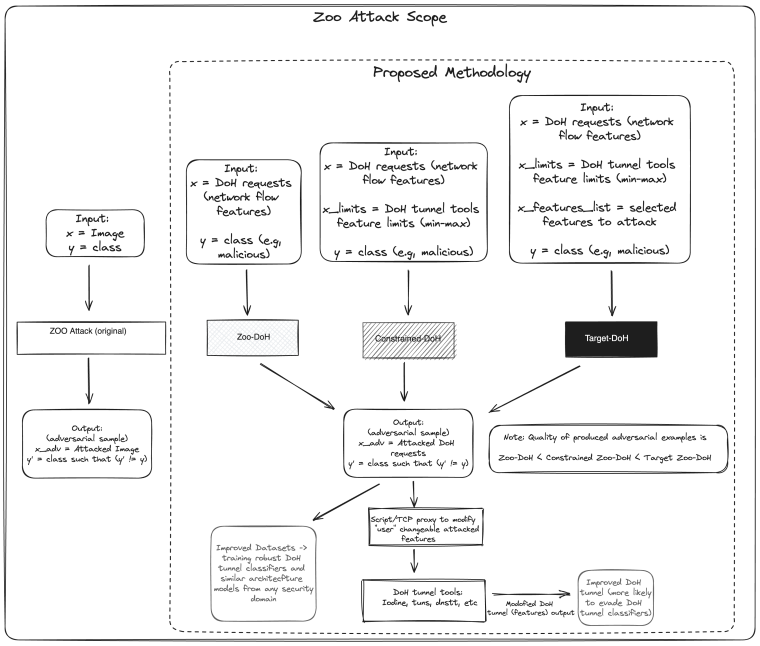

Figure 3. Methodology - ZOO Adaptations to support DoH features

Equation (6). We adopt the authors' original proposal to minimize the objective function, as explained in Section 2.3.2.

Given that this adaptation of the ZOO attack does not impose value boundaries on the features associated with DoH tunnel detection tools, we anticipate a high success rate in generating adversarial examples. However, this lack of constraints will likely result in unrealistic feature values, such as negative or excessively large time and packet size measurements, thereby compromising the practical applicability of the generated adversarial examples.

4.2. Constrained ZOO-DoH

In the second adaptation of the ZOO algorithm, *Constrained ZOO-DoH*, different from the previous adaptation, we account for the range of extracted features from DoH tunnel tools from the merged dataset. The values range constrains the algorithm to clip the values between these feature limits during the minimization process. This is necessary because the ZOO algorithm would apply broader limits without constraining feature value limits during the attack.

We calculate the feature limits (min, max) of DoH tunnel tools by extracting the quantiles of the values. To eliminate potential outliers, we choose 10% quantiles for the minimum values and 90% for the maximum values. Consequently, the objective function now relies on a new parameter, x_{limits}, containing the feature range list mentioned earlier, as shown in Equation (7).

$$J(x, x_{adv}, y, c, x_{limits}) = \|x - x_{adv}\|_2^2 + c \cdot \max\left(0, \max_{i \neq y}(f_i(x_{adv})) - f_y(x_{adv}) + \kappa\right) \quad (7)$$

where x_{adv} is subject to x_{limits}.

We anticipate this ZOO adaptation will achieve a moderate success rate while generating more realistic adversarial examples than the previous version. However, the resulting adversarial examples may only partially capture real-world scenarios, as certain features, such as packet size-based features, are targeted for manipulation while others, like time-based features, must remain unchanged. This limitation highlights the need for further refinement, which we will address in the subsequent adaptation.

4.3. Targeted ZOO-DoH

The last adaptation of the ZOO attack constrains the algorithm even more during the minimization process. In this version, *Target ZOO-DoH*, we modify the previous version by adding a list to set up specific attack features while leaving others unmodified. We adopt this approach because previous ZOO-DoH adaptations can generate adversarial examples that do not accurately reflect real-world scenarios, such as modifying DoH features that DNS tunnel tools cannot utilize. The Equation (8) shows the resulting objective function, considering the list of features to attack $x_{feature\_list}$.

$$J(x, x_{adv}, y, c, x_{limits}, x_{feature\_list}) = \|x - x_{adv}\|_2^2 + c \cdot \max\left(0, \max_{i \neq y}\left(f_i(x_{adv})\right) - f_y(x_{adv}) + \kappa\right) \tag{8}$$

where x_{adv} is subject to x_{limits} only for $x_{feature\_list}$.

We expect this ZOO adaptation will achieve a low success rate but generate more realistic adversarial examples than the previous adaptation. Although we expect only a limited number of adversarial examples to compromise the tunnel classifier successfully, this outcome is sufficient to guide the enhancement of real tunnel tools, enabling them to produce optimized feature values that can evade detection. This approach assumes that the user or tunneling tool can modify a subset of model features (i.e., the $x_{feature\_list}$). The most practical features to target are those that the user or tunnel application, such as time-dependent or packet size-based features, can easily modify. The application of these adversarial examples in DoH tunnel tools is discussed in detail in Section 5.

5. Experiments and Results

This section details the experiments and discusses the results from simulated attacks for the three ZOO algorithm adaptations.

5.1. Experiment Details

We tested the three proposed adaptations of the ZOO algorithm against the merged dataset, as explained in Section 4. We defined three experiments: E_1, E_2, and E_3, which correspond to *ZOO-DoH*, *Constrained ZOO-DoH*, *Target ZOO-DoH*, respectively. The inputs for our experiments consist of extracted features from the following DNS tunnel tools: dns2tcp, dnscat2, Iodine, dnstt, TCP over DNS, and tuns, available in the merged dataset. Additionally, we used the same dataset to train all the classifiers, specifically the DoH tunnel detection models.

We employed the sci-kit-learn[5] toolkit with default parameters[6] to build the target model methods. We utilized the XGBoost[7] library framework for building the Extreme Gradient Boosting (XGB) model. Table 3 details the trained classifiers and their respective metrics. We use the Adversarial Robustness Toolbox (ART)[8] to implement the ZOO adaptations. ART is a Python library for Machine Learning Security hosted by the Linux Foundation AI & Data Foundation.

The experiments are defined as follows:

Table 3. Experiments - DoH tunnel classifiers details

Target Model	Method	F1 Score - Benign/Malicious	Accuracy (%)
TM_1	GB	0.9990/ 0.9990	99.90
TM_2	XGB	0.9993/0.9993	99.93
TM_3	BS	0.9170/0.9213	91.92
TM_4	RF	0.9963/0.9964	99.64

E_1 : In this experiment, we run *ZOO-DoH* with the merged dataset against all target model methods;

E_2 : In the second experiment, we run the *Constrained ZOO-DoH* adaptation attack with the merged dataset, including the DoH feature range (min-max) list, as explained in the Methodology Section;

E_3 : We executed the *Target ZOO-DoH* adaptation using the same input data as in Experiment E_2 while including the subset F13-F24 from the feature list, resulting in a total of 12 features. This subset specifically includes time-dependent features, which were selected based on our pre-experiments, where we demonstrated their reproducibility in real-world scenarios using the dnstt tunnel tool and a TCP proxy, as illustrated in Figure 3.

For all the experiments to succeed, they must meet the following two conditions:

$C1$: The adapted algorithm must minimize the respective objective function, thereby generating adversarial examples;

$C2$: The last ZOO adaptation, Target ZOO-DoH adaptation, must generate at least one adversarial example for each considered DoH tunnel tool.

Upon satisfying these conditions, we validate that the adaptations work for all the model architectures considered and that real DoH tunnel tools can use the generated adversarial examples to evade DoH tunnel classifiers, thereby underscoring the significance and impact of our research.

The experiments conducted in this study have certain limitations. The results are influenced by the constraints of the datasets, including specific network conditions (e.g., latency, jitter), the use of particular tunnel tools, and the response times of specific DoH servers. Furthermore, replicating these experiments in real-world scenarios presents additional challenges since the datasets were derived from controlled environments.

[5]`https://scikit-learn.org`.

[6]The used parameters are available in the git repository, indicated in the section 6.

[7]`https://xgboost.readthedocs.io`.

[8]`https://adversarial-robustness-toolbox.readthedocs.io`

5.2. Discussion

The results of the experiments are presented in Figure 4 and 5. The first figure shows the mean attack success rate for all target model methods organized by DoH tunnel tools. The second illustrates the same information but with more details for each target model method.

The attack was successful across all target model methods in experiment E_1. As anticipated, the *ZOO-DoH* adaptation, lacking constraints, classified all generated adversarial examples (i.e., the modified malicious DoH tunnel tool requests) as benign (i.e., regular DoH requests). Since the algorithm could perturb any feature, it successfully converged across all classifiers. However, despite this success, the generated adversarial examples are unsuitable for real-world scenarios, as discussed in section 3. Upon unnormalizing the adversarial features, we encountered unrealistic values, such as negative time intervals and excessively large packet sizes. Consequently, we consider this experiment as a validation that our initial adaptation functioned correctly while also underscoring the necessity for an algorithm that imposes constraints to prevent the generation of such unrealistic values.

In Experiment E_2, we incorporated the feature limitations of DoH tunnel tools into the Constrained *ZOO-DoH* algorithm. As illustrated in Figure 5, the attack's success rate was lower than in Experiment E_1, which aligns with our expectations. The adaptation of the ZOO algorithm could not achieve convergence when applied to the TCP over DNS tool. The underlying reasons for this lack of convergence will be explored next, mainly as similar outcomes were observed in Experiment E_3. Furthermore, the attack perturbed features that are not easily replicable within DoH tunnel tools, specifically features F7-F9 and F27-F28. This poses significant challenges in replicating the attack under real-world conditions, as these perturbed features are difficult to reproduce in practical DoH tunnel implementations.

In the final experiment, E_3, the Target *ZOO-DoH* algorithm successfully generated adversarial examples across all target model architectures, thereby validating the most constrained adaptation of the ZOO approach. Despite being limited to perturbing only specific features relevant to DoH tunnel tools (F13-F24), the algorithm demonstrated a high success rate for the tuns DoH tunnel tool in target model architectures 1, 2, and 4. However, the Target ZOO adaptation was unsuccessful for target model architecture 3, the Balanced Stacked (BS) model. Although the BS model architecture appeared to be the most robust based on structural considerations alone, it is crucial to emphasize that this robustness cannot be generalized to all adversarial attacks. The observed results are specific to the constraints of our experiments, including the particular dataset employed.

Both experiments E_2 and E_3 demonstrated that the respective ZOO adaptations could not generate adversarial examples for the DNS over TCP tunnel tool. Upon analyzing the distribution of each dataset, no significant differences were identified among the DoH tunnel tools. Consequently, we cannot attribute the unsuccessful attacks on this tool to variations in data distribution. This suggests that the failure of the ZOO algorithm to converge was likely due to the specific constraints imposed during these experiments.

Additionally, we successfully reproduced the adversarial examples generated in Experiment E_3 using the dnstt tunnel tool. This was achieved by employing a TCP proxy

to modify the targeted features, as shown in Figure 3. This successful replication validates our methodology and confirms the practical applicability of our theoretical assumptions.

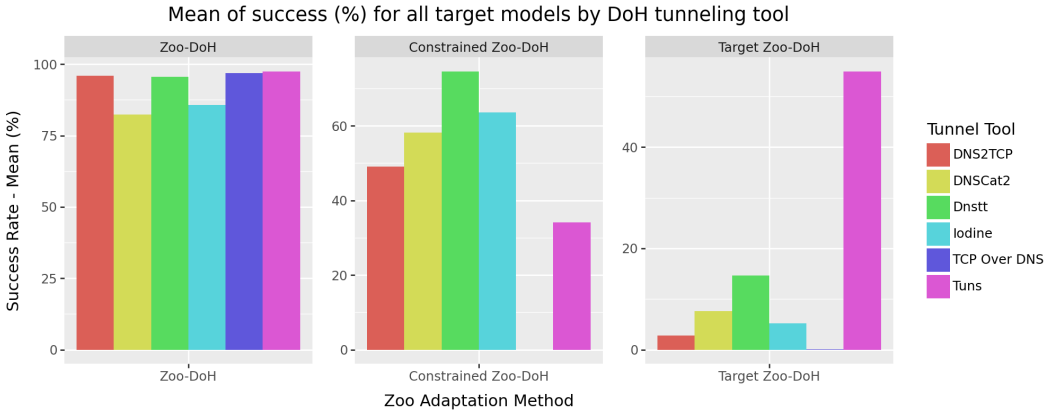

Figure 4. ZOO Attack Results

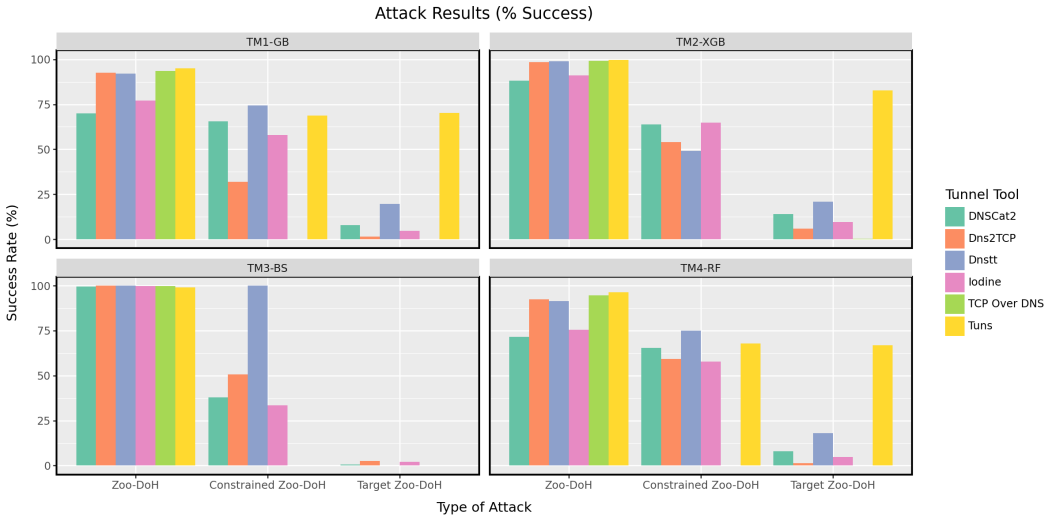

Figure 5. ZOO Attack Detailed Results by Tunnel Tools

6. Conclusion and Future Work

This work proposes three Zeroth Order Optimization (ZOO) Attack adaptations to support DoH tunnel detection requests. Our methodology leverages real-world data from tunnel detection tools to train classifiers and generate adversarial examples. Our results demonstrate that all three ZOO-DoH adaptations successfully evaded all target model methods. Given the assumptions in Section 2.5, we conclude that all state-of-the-art models are likely susceptible to adversarial black-box attacks.

Furthermore, researchers can use the ZOO adaptations to solve other robustness issues in any security domain. The only requirement is that the researcher knows the range of their tools and which features the attack algorithm can modify. In general, the research community can use our methodology to design more robust models, highlighting the practical implications of our research.

In future work, we plan to enhance our methodology to facilitate the reproduction of adversarial examples in real-world tools. Additionally, we intend to extend the applicability of the proposed methods to other security domains, such as ML-based intrusion detection systems (NIDS) [Debicha et al. 2023], and to refine the attack strategy by incorporating feature relevance into the model decision process. We aim to release the corresponding datasets, enabling the security community to train and develop more robust models. Furthermore, we plan to explore and adapt additional adversarial attack algorithms as part of our ongoing research efforts.

The source code to reproduce this work and additional experimental setup and parameter details are available via the repository at `https://github.com/e-valente/doh-deception`.

References

Alenezi, R. and Ludwig, S. A. (2021). Classifying dns tunneling tools for malicious doh traffic. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–9.

Borgolte, K., Chattopadhyay, T., Feamster, N., Kshirsagar, M., Holland, J., Hounsel, A., and Schmitt, P. (2019). How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem. In *Proceedings of the 47th Research Conference on Communications, Information and Internet Policy (TPRC)*. TPRC.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, Los Alamitos, CA, USA. IEEE Computer Society.

Catillo, M., Pecchia, A., Repola, A., and Villano, U. (2024). Towards realistic problem-space adversarial attacks against machine learning in network intrusion detection. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*, ARES '24, New York, NY, USA. Association for Computing Machinery.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 15–26, New York, NY, USA. Association for Computing Machinery.

Debicha, I., Bauwens, R., Debatty, T., Dricot, J.-M., Kenaza, T., and Mees, W. (2023). Tad: Transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems. *Future Generation Computer Systems*, 138:185–197.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples.

Hoffman, P. E. and McManus, P. (2018). DNS Queries over HTTPS (DoH). RFC 8484.

Hynek, K. (2023). *The Impact of Encrypted DNS on Network Security*. Dissertation, Czech Technical University in Prague, Prague, Czech Republic.

Lyu, M., Gharakheili, H. H., and Sivaraman, V. (2022). A survey on DNS encryption: Current development, malware misuse, and inference techniques. *CoRR*, abs/2201.00900.

Mitsuhashi, R., Jin, Y., Iida, K., Shinagawa, T., and Takai, Y. (2022). Malicious dns tunnel tool recognition using persistent doh traffic analysis. *IEEE Transactions on Network and Service Management*, pages 1–1.

MontazeriShatoori, M., Davidson, L., Kaur, G., and Habibi Lashkari, A. (2020). Detection of doh tunnels using time-series classification of encrypted traffic. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 63–70.

Niktabe, S., Lashkari, A. H., and Roudsari, A. H. (2024). Unveiling DoH tunnel: Toward generating a balanced DoH encrypted traffic dataset and profiling malicious behavior using inherently interpretable machine learning. *Peer-to-Peer Networking and Applications*, 17(1):507–531.

Papernot, N., McDaniel, P., and Goodfellow, I. (2016). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples.

Singh, S. K. and Roy, P. K. (2020). Detecting malicious dns over https traffic using machine learning. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, pages 1–6.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks.

Sánchez Sánchez, P. M., Huertas Celdrán, A., Bovet, G., and Martínez Pérez, G. (2024). Adversarial attacks and defenses on ml- and hardware-based iot device fingerprinting and identification. *Future Generation Computer Systems*, 152:30–42.

Toulas, B. (2024). Hackers use dns tunneling for network scanning, tracking victims. Accessed: 2024-06-14.

Wang, Y., Zhou, A., Liao, S., Zheng, R., Hu, R., and Zhang, L. (2021). A comprehensive survey on dns tunnel detection. *Comput. Netw.*, 197(C).

Yassine, S., Khalife, J., Chamoun, M., and Ghor, H. E. (2018). A survey of dns tunnelling detection techniques using machine learning. In *International Conference on Big Data and Cyber-Security Intelligence*.

Zebin, T., Rezvy, S., and Luo, Y. (2022). An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Transactions on Information Forensics and Security*, 17:2339–2349.

Zhan, M., Li, Y., Yu, G., Li, B., and Wang, W. (2022). Detecting dns over https based data exfiltration. *Computer Networks*, 209:108919.

Žiža K, T. P. and Vuletić, P. (2023). Dns exfiltration detection in the presence of adversarial attacks and modified exfiltrator behaviour. *Int. J. Inf. Secur. (2023).*