Received July 7, 2021, accepted July 30, 2021, date of publication August 11, 2021, date of current version August 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104115

Signature-Based Traffic Classification and Mitigation for DDoS Attacks Using Programmable Network Data Planes

MARINOS DIMOLIANIS, (Graduate Student Member, IEEE), ADAM PAVLIDIS, AND VASILIS MAGLARIS
School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece

Corresponding author: Marinos Dimolianis (mdimolianis@netmode.ntua.gr)

This research is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme <<Human Resources Development, Education and Lifelong Learning 2014-2020>> in the context of the project "COLLABORATIVE MITIGATION OF CYBERATTACKS IN PROGRAMMABLE NETWORKS" (MIS 5049178).

ABSTRACT Distributed Denial of Service (DDoS) attacks mitigation typically relies on source IP-based filtering rules; these may present scaling issues due to the vast amount of involved sources. By contrast, we propose a source IP-agnostic DDoS traffic classification and filtering schema that identifies malicious packet signatures via supervised Machine Learning methods and subsequently generates signature-based filtering rules. To accelerate packet processing, our schema utilizes XDP middleboxes operating as programmable Deep Packet Inspectors. Signatures are extracted from network traffic as unique combinations of the most significant packet features; these are subsequently fed to supervised Machine Learning algorithms that classify them as malicious or benign. Malicious signatures undergo a reduction process tailored to the attack vector in order to generate a concise set of filtering rules, thus expediting mitigation performance. Our schema was implemented as a proof-of-concept and evaluated for DNS volumetric attacks in terms of signature classification accuracy and packet filtering throughput. Experiments were based on benign and malicious traffic datasets recorded in production network environments. Our approach was compared to source-based mechanisms in terms of (i) malicious traffic identification, (ii) filtering rules cardinality, and (iii) packet processing throughput required in modern high speed networks. The experimental results demonstrate that our signature-based approach outperforms IP-based alternatives, achieving high detection accuracy and significant generalization capabilities.

INDEX TERMS Packet signatures, traffic classification, DDoS mitigation, supervised machine learning, data plane programmability, eXpress Data Path.

I. INTRODUCTION

Distributed Denial of Service (DDoS) attacks originate from compromised hosts and/or exploited vulnerable systems producing traffic from a large number of sources [1]. Such attacks are continuously increasing in frequency and magnitude [2].

Legacy DDoS protection mechanisms maintain statistics based on source IP or network flows to detect and ultimately mitigate malicious traffic. Maintaining flow/IP-based metrics requires data from lengthy time-windows that may hinder real-time identification of malicious traffic and the subsequent mitigation. Moreover, traditional filtering mechanisms rely on IP-based rules that increase proportionally to the number of alleged malicious sources. In massive attacks that may include millions of source IPs [1], such a filtering approach raises scalability issues [3], [4].

To counter the shortcomings of IP-based schemes, we propose a source IP-agnostic DDoS protection mechanism that classifies and mitigates network attacks based on packet signatures i.e. unique combinations of packet field values. Motivated by our early effort [5] on SYN Flood attacks, we consider DDoS Amplification attacks, commonly used to overwhelm network infrastructures. The proposed approach relies on the widely observed fact that such attacks may be characterized by a modest number of salient packet

The associate editor coordinating the review of this manuscript and approving it for publication was Guangjie Han.

characteristics [1]. Consequently, our schema attempts to dynamically reveal related packet characteristics (signatures) and use them as filters to block the attack traffic in a scalable fashion.

In a nutshell, the proposed mechanism continuously monitors the network traffic and extracts packet signatures based on the most important features tailored to an attack vector (e.g. DNS or NTP Amplification attacks). Packet signatures are classified via supervised Machine Learning (ML) algorithms, appropriately trained with benign and malicious traffic, focusing on distinct packet fields (features). Malicious signatures are further subjected to a reduction process before being employed as filtering rules to expedite mitigation performance. The reduced set of signatures is finally deployed on high-performance programmable scrubbing middleboxes.

The remainder of this paper is structured as follows: Section II contains background information and discusses related work; Section III offers an architectural overview; Section IV provides implementation details of the proposed Signature-based Traffic Classification and Mitigation schema; Section V provides experimental evaluations for volumetric DNS attacks regarding processing performance and classification accuracy based on traffic traces recorded in production networks. Finally, Section VI discusses future steps and directions.

II. BACKGROUND AND RELATED WORK

In subsection II-A, we present background information related to advances in programmable data planes, focusing on eXpress Data Path (XDP) as a key component in our architecture. Related efforts on DDoS protection are discussed in subsection II-B and our main contributions are summarized in subsection II-C.

A. PROGRAMMABLE DATA PLANES—eXpress DATA PATH

Recent advances in data plane programmability enable customized solutions tailored to various network applications. Approaches such as P4 [6] enable operators to reprogram the processing pipeline of a network element, employing novel switch architectures [7]. Implementing complicated algorithms in such time/memory constrained environments poses significant challenges; it is possible however to offload preliminary steps of anomaly detection schemas to P4 devices [8]–[10]. Similar programmable data plane approaches may rely on Linux-based servers combined with high throughput frameworks such as Data Plane Development Kit (DPDK) and PF_RING that bypass the Linux Kernel.

An alternative approach is the eXpress Data Path (XDP) [11], a softwarized data plane that harmonically co-exists with the Linux Kernel. XDP is executed prior to heavy networking stack operations and can be seamlessly ported in various Linux machines. It provides high-performance programmable packet processing in Commercial off-the-shelf (COTS) hardware, thus enabling deployment even within application servers to gather data on or filter malicious traffic. XDP has been widely adopted in production network environments for various applications e.g. Load-Balancing, Intrusion Detection and DDoS protection.

XDP programs, written in C, are executed either in software within the context of the network driver or offloaded directly in Network Interface Cards (NICs), e.g. Netronome SmartNICs [12]. Their execution is initiated upon the arrival of packets at the network interface. In turn, packet data can be parsed, extracted and stored in persistent memory referred to as Berkeley Packet Filter (BPF) Maps [11]. These are key-value stores defined when the XDP program is loaded. XDP returns an action for each packet which defines how it should be handled. The packets can be either (i) dropped - XDP_DROP, (ii) passed to the network stack - XDP_PASS, (iii) redirected to another interface - XDP_REDIRECT or (iv) transmitted back - XDP_TX. In this work, we employed XDP to design and implement high-performance yet programmable monitoring and filtering mechanisms. Note that, the design and implementation of XDP applications needs to account for specific limitations: (i) bounded loops, (ii) fixed-size data structures, (iii) 4096 BPF instructions per program, and (iv) limited support of kernel functions.

B. DDoS TRAFFIC CLASSIFICATION AND FILTERING

There are various efforts reported in the literature that attempt to classify and filter DDoS attacks. In subsection B-1, B-2 below, we present related flow-based and signature-based schemes accordingly. These efforts are summarized in Table 1.

1) FLOW-BASED MECHANISMS

In [13], a DDoS traffic classification schema based on a Multilayer Perceptron (MLP) was introduced. Traffic metrics related to flows and packet rates (UDP, ICMP) are collected and used as input to an MLP, tasked with classifying network traffic to benign/malicious.

In [14], an OpenFlow (OF) DDoS detection mechanism was presented. This periodically collects entries from OF-enabled network devices, extracts flow-related features and classifies them using Self-Organizing Maps (SOM). In [15], an SDN DDoS detection and mitigation schema was proposed. Sharp increases in the rate of Packet-In messages are considered as an indication of DDoS attacks; subsequently a mitigation pipeline is triggered. OpenFlow rules are collected from network devices and classified via an appropriate MLP that uses the same feature set as in [14]. Malicious flows are then blocked via appropriate mitigation entries in OF-enabled devices.

In [16], a large set of flow-related features is extracted from packets and sent to OF Controllers. These are used as input to a Stacked Autoencoder (AE), which provides traffic classification of the flow as benign or attack. Authors highlight processing limitations in Controller-based packet collection and feature extraction.

In [17], a two-level schema was introduced. Initially, entropy values are calculated for the number of destination IPs and ports, with sudden changes indicating an ongoing attack. The victim is identified and traffic destined towards its IP is redirected to an OF-enabled switch. This device acts as a second, more refined level of detection, that uses packet symmetry to identify malicious flows. Malicious flows are subjected to source IP-based aggregation in order to reduce the required blocking rules imposed by hardware limitations. Finally, filtering rules are deployed to the OF switch while benign traffic is redirected back.

In [18] *ATLANTIC*, an SDN framework for DDoS attack detection and mitigation, was proposed. Entropy changes for specific flow features within consecutive time-windows indicate the existence of an attack. Network flows responsible for entropy changes are fed in a traffic classification component based on K-means and Support Vector Machines (SVM). K-means is used initially to create clusters of common flows and SVM is further used to identify malicious flows. Subsequently, drop rules are installed for malicious flows.

A flow-based traffic classification mechanism was suggested in *LUCID*[19]. Flow values are collected from different time windows and represented as arrays; subsequently these arrays are fed to a Convolutional Neural Network (CNN) to identify time-dependent traffic patterns. Attack mitigation was not addressed in the *LUCID* paper.

2) SIGNATURE-BASED MECHANISMS

Signature-based traffic classification and filtering is commonly featured in Intrusion Detection/Prevention Systems (IDS/IPS), e.g. *Suricata* [20]. Network packets are monitored and their packet field values are compared to predefined sets of malicious signatures. Notably, the widely employed DDoS detection tool *FastNetMon* [21], relies on static rules to identify Amplification attacks. Although these approaches are able to instantly identify previously observed attack patterns, they are not able to detect zero-day threats.

By contrast, in [22] a tool for extracting zero-day attack signatures was proposed; upon the detection of an attack, their system analyzes both benign and attack packets. Signatures suddenly appearing in high frequency in the network traffic are attack indicators, while evenly distributed signatures usually characterize benign traffic.

In [23] *DeepDefense*, a DDoS detection schema based on Recurrent Neural Networks (RNN) was introduced. Traffic traces, collected within sliding time windows, are translated into arrays of packet features. These are fed to an RNN that segregates malicious from benign packets.

Finally, *Cloudflare*, currently one of the largest Content Delivery Networks (CDN) that also offers DDoS protection services, employs packet signatures to filter malicious traffic [24]. To the best of our knowledge, the exact methods for traffic classification and signature-based filtering are not publicly available and thus we cannot compare our approach with them.

3) KEY CONTRIBUTIONS

Our key contributions can be summarized as follows:

- Most of the reported efforts in the literature employ metrics aggregated by IP addresses or network flows for traffic classification [13]–[18]. In contrast, we focus on the most appropriate packet features to identify malicious signatures based on supervised Machine Learning algorithms. Due to their enhanced generalization capabilities, these can accurately identify zero-day (unseen) attacks (outperforming static approaches [20], [21]).
- We exploit common characteristics observed in the attack traffic to generate appropriate signature-based filtering rules. These are subjected to a reduction process that minimizes their number and expedites the mitigation performance.
- Our approach does not require collection of data over lengthy time-windows and corresponding time references as in [22], [23]. Instead, current packet field values are used, thus expediting detection and mitigation of attack traffic with no significant deterioration of classification accuracy.
- We propose a dynamic, tunable yet high-performance scrubbing mechanism based on programmable software data planes (XDP). Unlike proprietary monolithic solutions, our approach offers programmable monitoring and filtering functionalities without compromising on packet processing performance.
- We conducted detailed experiments focusing on volumetric DNS attacks; we employed high packet rates and real network data (benign and malicious) to illustrate the applicability of our mechanism in production network environments.

III. DESIGN PRINCIPLES AND ARCHITECTURAL OVERVIEW

In this section, we outline design principles and present a baseline overview of the proposed Signature-based Traffic Classification and Mitigation architecture.

A. DESIGN PRINCIPLES

The main design principles of our mechanism are summarized below:

- *Signature-based filtering*: We opt to surgically mitigate DDoS attacks focusing on distinct packet feature combinations (signatures) exhibited by offending traffic. Unlike traditional DDoS defense mechanisms that rely on blocking a massive number of IP sources, our approach attempts to generate IP-agnostic filtering rules.
- *Filtering rules reduction*: Filtering rules are stored within network devices (switches, routers, firewalls) that typically impose limits to the number of entries they can support. To reduce their number, source-IP based procedures [4], [17] employ IP aggregation techniques. Our signature reduction mechanism identifies instead a concise set of rules required to block an attack, with minimal effect on benign traffic.

TABLE 1. Taxonomy of traffic classification and filtering mechanisms.

Paper/Framework	Classification Mechanism	Input Data	Classifier	Source IP-agnostic Mitigation
Siaterlis et al. [13]	Flow-based	Flow length, duration, generation rate, UDP traffic ratio and ICMP traffic ratio	MLP	✕
Braga et al. [14]	Flow-based	Packets, bytes, duration per flow; pair-flows (%), growth of single flows/ports	SOM	✕
Cui et al. [15]	Flow-based	Packets, bytes, duration per flow; pair-flows (%), growth of single flows/ports	MLP	✕
Niyaz et al. [16]	Flow-based	Large set of features for TCP, ICMP and UDP flows	AE	✕
Giotis et al. [17]	Flow-based	Packet symmetry ratio per flow	Threshold-based	✕
Santos da Silva et al. [18]	Flow-based	Packets, bytes and duration per flow	SVM, K-Means	✕
Doriguzzi-Corin et al. [19]	Flow-based	Packet field values organized in flows	CNN	✕
Suricata [20]	Signature-based	Packet field values (L3-L7)	Well-known attack signatures	✓
FastNetMon [21]	Signature-based	Packet field values (for DNS, NTP Amplification attacks)	Well-known attack signatures	✕
Afek et al. [22]	Signature-based	Frequencies of packet field values (Evaluation focused on L7 attacks)	Threshold-based	✕
Yuan et al. [23]	Signature-based	Packet field values (L2-L7)	RNN	✕
Cloudflare DDoS Protection [24]	Signature-based	Not Publicly Available	Not Publicly Available	Not Publicly Available
This work	Signature-based	Packet field values (L3-L7)	RF, MLP	✓

- *Traffic classification based on supervised Machine Learning (ML) algorithms:* Our approach is trained using packet characteristics from normal (benign) traces and past attack incidents. The learning process can be tailored to specific network environments, thus enhancing classification accuracy. To that end, the employed features should be carefully selected and tuned depending on anticipated attack vectors.
- *High performance scalable network functions based on programmable middleboxes:* Typically, traffic monitoring and filtering functionalities are implemented by monolithic appliances. In contrast, we opted to use COTS hardware (i.e. low-cost NICs) as data plane programmable appliances powered by the XDP framework. This enables online packet handling without imposing control plane processing overhead. XDP-enabled appliances can be instantiated on-demand and scaled according to traffic and application requirements, thus providing a suitable mechanism for cloud-based scrubbing services.

B. ARCHITECTURAL OVERVIEW

In Fig. 1, we present a high-level overview of the proposed architecture for DDoS protection, applicable either in transit provider networks or customer/edge network domains. Our mechanism consists of four separate components that offer: (a) *Signature Extraction*, (b) *Signature Classification*, (c) *Signature Reduction* and (d) *Anomaly Mitigation*. In what follows, we outline the DDoS detection and mitigation workflow referring to steps *i* – *vi* illustrated in Fig. 1.

Benign and malicious traffic originating from various Internet sources traverses through a network infrastructure equipped with programmable devices. Network traffic is continuously monitored (step *i*) in the data plane by the *Signature Extraction* component. This component employs high-performance programmable mechanisms (e.g. XDP) to extract appropriate packet fields, i.e. signatures, pertaining to different attack vectors. Note that these fields should be selected after careful examination of benign and malicious traffic for a specific exploited protocol. Our methodology for selecting the most important packet fields (features) will be presented in subsection IV-A; note that the proposed method is not limited to a specific attack vector.

Extracted monitoring data (signatures) are organized per destination IP address and relayed (step *ii*) to the *Signature Classification* component, a control plane module that categorizes them as either benign or malicious. This component relies on classification methods based on supervised ML algorithms that have been trained with attack and benign traffic. Malicious signatures identify ongoing attacks targeting specific IP addresses (victims). Classified signatures are subsequently employed for mitigation rule generation (step *iii*) via the *Signature Reduction* component that expedites mitigation performance. This reduction process is formulated as a multi-objective (Pareto) optimization problem. Specifically, combinations of the most important packet features are explored to identify a smaller feature set that minimizes

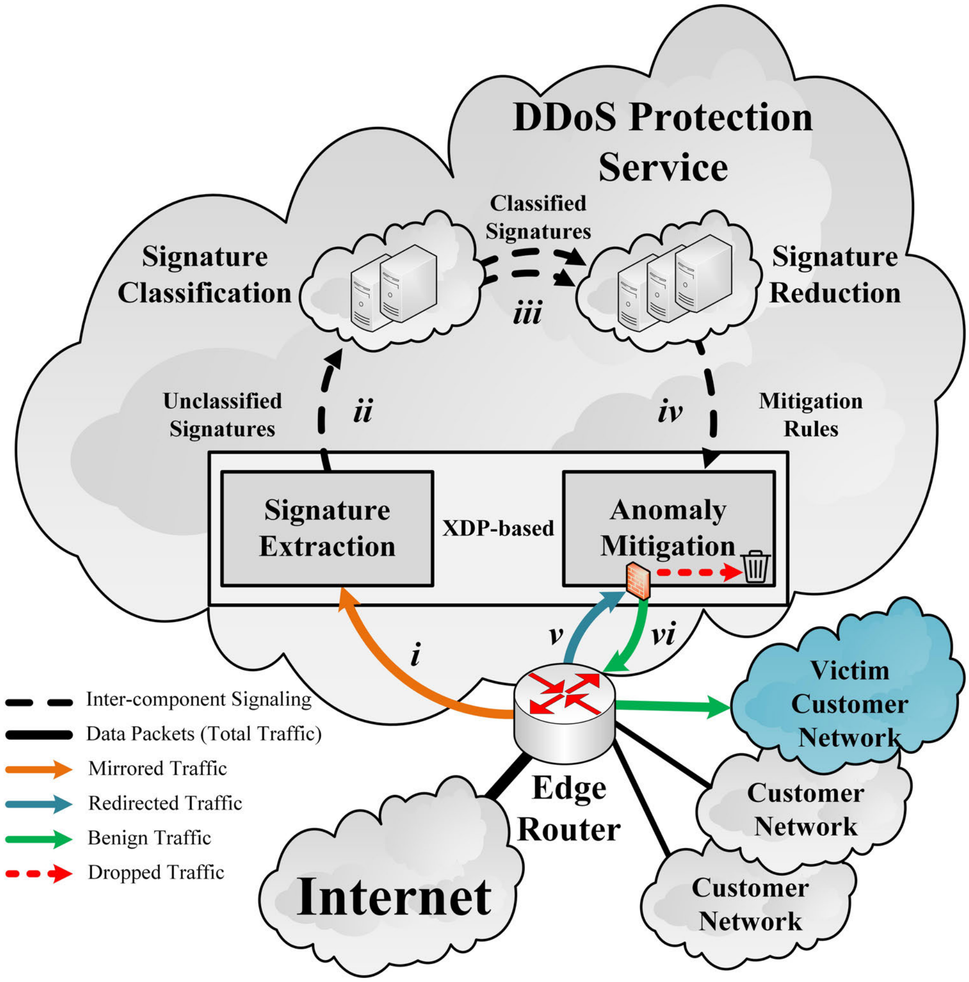

FIGURE 1. High-Level overview of the DDoS protection architecture.

the number of malicious signatures for an acceptable level of benign traffic drops. The selection of a Pareto optimal pair is based on DDoS Protection service operator preferences.

Finally, the reduced set of signatures is conveyed (step *iv*) to the *Anomaly Mitigation* component, that acts as a traffic scrubbing mechanism in the data plane. Data packets destined to the victim IP are redirected to this component (step *v*) via appropriate traffic diversion techniques [25]. Malicious packets are dropped while benign traffic is returned back to the router (step *vi*) to be forwarded to the destination IPs.

Extraction, classification and reduction of signatures, as well as mitigation rule generation, are performed continuously in distinct intervals (time-windows). Selected intervals should be small (e.g. 10 seconds) to enable rapid propagation of information and ultimately prompt accurate traffic scrubbing.

IV. IMPLEMENTATION DETAILS

Our methodology for packet feature selection and implementation details of the components in Fig. 1 are presented in the following subsections.

A. PACKET HEADER FEATURE SELECTION METHODOLOGY

Packet header fields forming signatures are of paramount importance for our mechanism. They are used to (i) classify packets to malicious/benign and (ii) create filtering rules for blocking the offending traffic.

In DDoS Amplification attacks, vulnerable protocols and services are exploited in a very specific manner for generating massive amounts of traffic. This traffic exhibits packet characteristics that typically deviate from benign network traffic. In order to identify the most important characteristics pertaining to a specific attack vector, we select the relevant packet header fields (features) of each abused protocol. For that purpose, we employ the methodology described below.

We start with an initial set of n features $F = \{F_1, F_2, \ldots F_n\}$, that includes (i) packet header fields of an abused protocol (e.g. DNS) and (ii) IP packet Total Length and UDP datagram Length fields (these values may differ in cases of IP fragmentation of large UDP packets). The former may reveal packet field values that are employed for generating large payloads in such attacks. The latter may correspond to large values, typical for DDoS Amplification [26].

The packet header field selection algorithm uses both benign and malicious traffic for an attack vector to train a Random Forest (RF) classifier [27] based on a training dataset T of examples with F features. The RF training process provides (i) the Out-Of-Bag (OOB) score, a metric that shows the accuracy achieved on examples that were not included in the training process of each decision tree [28] and (ii) the importance of each feature [29]. High values of OOB score illustrate that the employed fields can be used to accurately classify benign and malicious packets. The feature selection pseudocode is:

```
Packet Header Field Selection Algorithm
Input: Training Dataset $T$, Packet Features $F = \{F_1, F_2, \ldots F_n\}$
Output: Packet Features $F' = \{F_1, F_2, \ldots F_m\}$
($OOB_n$, $F_{\text{ranked}}$) ← Random Forest ($T$, $F$)
$F_{\text{ranked}}$ ← sort_descending($F_{\text{ranked}}$)
for each q ∈ [1, n) do:
    m = n − q
    $F'$ = TOP m entries from $F_{\text{ranked}}$
    $OOB_m$ ← Random Forest ($T$, $F'$)
    if ($OOB_n - OOB_m$) ≥ $\varepsilon$ then
        return $F'$
end for
```

The RF feature importance metric enables the selection of $m < n$ important features according to the above iterative process, see also [30]. As a result, we obtain a reduced set of features $F' = \{F_1, F_2, \ldots F_m\}$ that are used for packet monitoring, traffic classification, signature reduction and attack mitigation purposes. Note that in our experiments out of $n = 20$ features, we were able to obtain $m \leq 8$ important features (see Table 2 and 3 in subsection V-B).

The elimination of non-important features (selecting m most important ones) has the following benefits for our schema: (i) increased packet throughput of *Signature Extraction* and *Anomaly Mitigation* components of Fig. 1 since fewer packet fields are required to be parsed and stored; (ii) enhanced accuracy and shorter training times of supervised learning algorithms; (iii) lower complexity of the *Signature Reduction* component due to the lower dimensionality of its input (this will be elaborated in subsection IV-D).

B. SIGNATURE EXTRACTION (SE)

SE is a high-performance monitoring mechanism based on the XDP framework. It collects mirrored network traffic, extracts appropriate packet fields and conveys monitoring data to the *Signature Classification* (SC) component, as illustrated in Fig. 2.

The combination of packet feature values can be represented by the signature vector $X = [x_1 x_2 \ \ldots \ x_m]^{\mathrm{T}}$, where x_i is the value for packet field i. Each unique signature X corresponds to a row in the Monitoring Data table of Fig. 2. Every observed packet signature pertains to a counter stored within an appropriate BPF Map (i.e. hash table).

SE consists of various instances, each associated with a specific attack vector. They all contain a *Data Extractor* and a *Data Exporter* module:

- The *Data Extractor* is a kernel space XDP (data plane) program that extracts and stores packet header values for the preselected fields F', including the destination IP address. Destination IPs are required for the identification of the victim and subsequent traffic scrubbing (redirection and filtering).
- The *Data Exporter* is a user space program that periodically retrieves the contents (i.e. signatures) of the BPF Map and conveys them to the SC component.

Note that the SE component could be implemented using any approach that provides access to packet fields such as sFlow [31]. We opted for XDP since it provides cost-effective high-throughput monitoring of all packets (no sampling) and does not exhibit limitations on the available packet fields to be collected.

C. SIGNATURE CLASSIFICATION (SC)

SC collects monitoring data and classifies them using supervised Machine Learning (ML) methods to identify malicious signatures. It consists of the *Data Handler* and the *ML Classifier* module. The *Data Handler* module collects the different signatures X relayed by the SE component and preprocesses them (if needed) in a data normalization step. In turn, the set of X is used as input to the *ML Classifier* module which classifies them as benign/malicious. This module is trained with malicious and benign traffic datasets related to a specific protocol (e.g. DNS attacks and benign DNS traffic).

Malicious signatures correspond to ongoing attacks targeting specific IP addresses (victims). The mitigation process for the victim IP addresses is initiated by conveying malicious and benign signatures to the *Signature Reduction* (SR) component to generate filtering rules (see the following subsection IV-D).

Note that we experimented with two widely used supervised ML algorithms (see subsection V-C); however, our schema can employ alternate classification algorithms.

D. SIGNATURE REDUCTION (SR)

SR receives both malicious and benign signatures from the SC component and reduces the number of malicious signatures to expedite the mitigation performance of the *Anomaly Mitigation* (AM) component. As mentioned, malicious signatures will be used to generate filtering rules. These are stored in memory resources (i.e. BPF Maps in the XDP context) that enable packet matching in the data plane. Their number significantly affects the deployment and lookup time in the BPF Map, which is ultimately related to the AM packet processing performance (throughput).

The SR component searches for a concise set of signatures that can block offending traffic, with minimal effect on the benign traffic. This was formulated as a multi-objective (Pareto) optimization problem, in which we search for feature

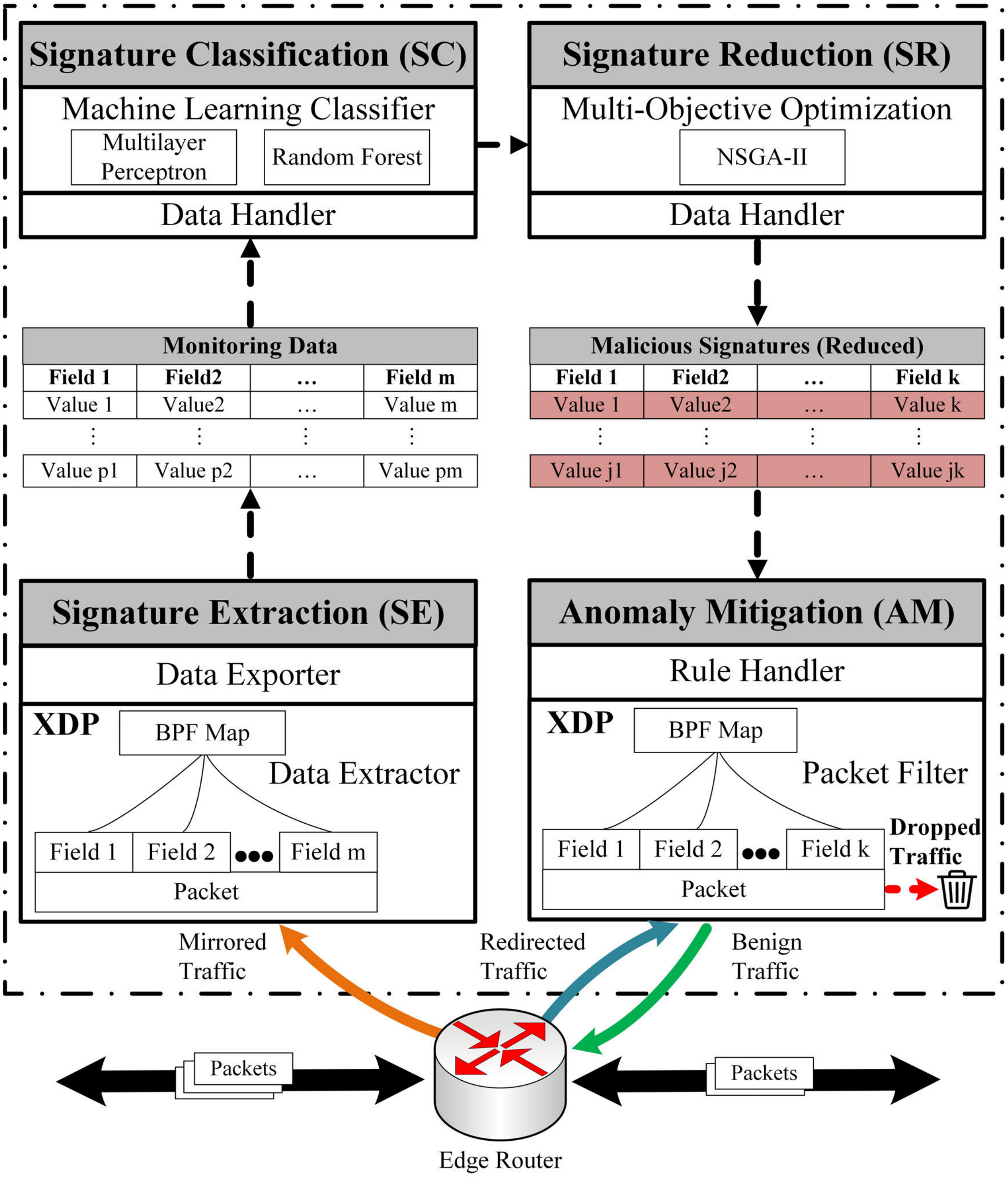

FIGURE 2. DDoS protection service component interactions.

subsets $F'' = \{F_1, F_2, \ldots F_k\}$ of the feature set $F' = \{F_1, F_2, \ldots F_m\}$, $k < m$, to identify operating points that simultaneously minimize:

1) the number of malicious signatures (filtering rules)
2) the percentage of benign traffic drops

Let M' and B' be the sets of malicious and benign signatures respectively based on features from F'. For each subset F'', we similarly define M'' and B'' using only the features in F''. Objective (i) is calculated as the number (cardinality) of unique signatures in M''. Objective (ii) is the number of benign packets that correspond to the signatures in $M'' \cap B''$ divided by the number of benign packets that correspond to signatures in B''. This provides the percentage of benign traffic that would be dropped (False Positive Rate) if we used as filtering rules the signatures in M''. Note that the intersection $M' \cap B'$ is an empty set; however, the intersection $M'' \cap B''$ may result to non-empty sets in the reduced feature space F'', corresponding to False Positive cases (see subsection V-C *Signature Classification Accuracy*).

The proposed optimization problem points to Pareto optimal solutions (referred to as Pareto-optimal front). However, due to stringent time constraints for attack mitigation, related algorithms would typically stop prior to Pareto-optimal front identification. We opted for a fast evolutionary approach based on Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [32]. The algorithm starts with arbitrary subsets $F'' \subset F'$ and iteratively attempts in each step to further reduce the objectives. At each iteration (generation), new subsets of F' are generated based on random combinations of F'' that correspond to the best solutions produced so far in previous iterations. The algorithm stops when a time limit is reached thus generating suboptimal subsets.

As stated above, the proposed approach will generate several solutions near the Pareto-optimal front. Naturally, only one of the can be ultimately selected for mitigating the attack. This selection should be tuned per customer network profile to depict network operator preferences e.g. acceptable percentage of dropped benign traffic (False Positive Rate). Finally, from the selected solution, signatures of M'' are conveyed to the AM component to generate filtering rules.

E. ANOMALY MITIGATION (AM)

AM is a high-performance programmable firewall based on the XDP framework. It consists of two modules: the *Rule Handler* and the *Packet Filter*. The former receives a list of malicious signatures associated with a victim IP, installs them as filtering rules in a BPF Map and triggers traffic redirection for the targeted victim IP. The latter is an XDP kernel space program similar to the *Data Extractor* module of the SE component. The *Packet Filter* receives traffic destined to the victim IP and extracts the packet fields based on the reduced set of signatures F''. The extracted packet fields values are subsequently compared to the filtering rules within the BPF Map. If the combination of packet fields (i.e. signature) of the received packet is contained in the BPF Map, the packet is dropped (XDP_DROP). Otherwise, the packet is considered benign and transmitted back (XDP_TX) to the edge router to be normally forwarded to the victim IP. For implementation options related to traffic redirection and reinjection see [25].

Note that SE can be implemented with alternate monitoring solutions (e.g. sFlow) that can extract packet characteristics. However, the AM component is tightly coupled with programmable data planes solutions, such as XDP, able to perform inline packet filtering based on selected packet fields.

V. EXPERIMENTAL EVALUATION

We selected as a case study volumetric DNS attacks, one of the most common DDoS Amplification attack vectors [33]. We evaluate our schema in an experimental testbed, employing real datasets and synthetic network traces as detailed in subsection V-A below. In short, our experiments attempt to: (i) identify and select the most important features for DNS traffic classification, (ii) assess the signature classification accuracy of our supervised learning mechanism and (iii) compare the proposed signature-based approach to source IP/flow-based alternatives. These are presented accordingly in subsections V-B, V-C and V-D.

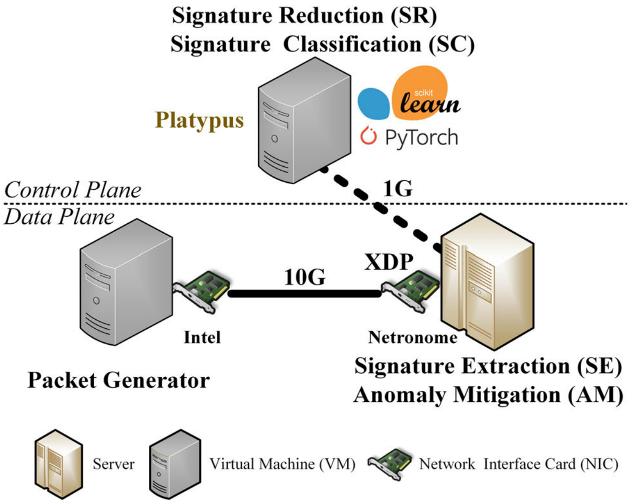

FIGURE 3. Proof-of-concept testbed setup.

A. DATASETS/TESTBED

Our proof-of-concept testbed is illustrated in Fig. 3. The experimental setup was used to evaluate packet monitoring, signature classification, signature reduction, and packet filtering capabilities. The SE and AM components were implemented within the XDP framework in the data plane. They were deployed on a physical machine (XDP-enabled node) equipped with a Netronome Agilio CX 2 × 10G SmartNIC [12]. For packet generation purposes, we used a Virtual Machine (VM), equipped with an Intel X520 NIC 2 × 10G, able to generate packets at high rates using the PF_RING ZC framework. The SC component was implemented using the *scikit-learn* and *PyTorch* Python libraries while the SR component was based on the *Platypus* framework [34]. They were both deployed as control plane modules on a VM equipped with 12 vCPUs and 12GB RAM.

Real network traces were used to assess the signature classification accuracy of our schema, whereas synthesized traffic was used for stress testing packet filtering capabilities. As benign traffic, we used DNS responses from: (i) a 10G transit link between WIDE and DIX-IE (an experimental Internet Exchange), henceforth WIDE-G [35], (ii) a 1G transit link between WIDE and an upstream provider, henceforth WIDE-F [35], and (iii) Thapar University Campus Network, henceforth TU Campus [36]. As malicious traffic, we used the *Booters* datasets. These datasets, henceforth individually referred to as $B_1, B_2, \ldots B_7$ or collectively as *Booters*, contain seven different DNS-based Amplification attacks generated by DDoS-for-Hire services. The attacks [26] were captured during a controlled experiment conducted between the University of Twente and SURFnet, the Dutch Research and Education Network.

All *Booters* attacks apart from B_4 and B_5 used *type ANY* DNS responses, a commonly used method for DNS Amplification attacks that returns every available Resource Record (RR) for a given fully qualified domain name. In B_4

and B_5 attacks, the attackers attempted to use *type A* requests. Specifically, B_4 contains multiple responses for the domain *packetdevil.com*, a domain name that resolves into a very large number of IP addresses in the DNS response payload. By contrast, B_5 corresponds to a *type A* attack, that could not generate responses with heavy payload.

B. PACKET HEADER FIELD SELECTION FOR DNS AMPLIFICATION ATTACKS

In this subsection, we evaluate the packet header field selection algorithm for three different combinations of benign and malicious DNS traffic. Initially, we selected the 20 packet fields (features) as presented in Table 2.

Employing the features of Table 2, we trained three different Random Forest (RF) classifiers consisting of 100 decision trees with default parameters of the *scikit-learn* library for tree structure and stopping [29]; each one includes all *Booters* traffic and a particular benign dataset (WIDE-G, WIDE-F, TU Campus). The selected features except for *dns.qry.name* correspond to numerical values and were fed directly to the RF classifiers; *dns.qry.name* was transformed to a numerical value via hash encoding. In Fig. 4, we depict the importance of each feature for the different combinations of datasets, as computed by the *scikit-learn* library. The reported values correspond to the average feature importance for multiple training iterations.

In order to identify the most important features, we employed for each dataset combination the iterative process described in subsection IV-A. The threshold ε (line 7 in *Packet Header Field Selection Algorithm* pseudocode) was set equal to zero. In Table 3, we present the most important features that the algorithm produced for each dataset.

One of the dominant features in all cases is the type of the query (*dns.qry.type*) since most attacks in the *Booters* dataset rely on DNS *type ANY* messages to generate large volumes of malicious traffic. The length of the IP packet and the UDP datagram are also important features; benign DNS traffic mainly consists of small packets while DNS Amplification attacks consist of large responses. Similarly, *dns.count.answers* and *dns.count.add_rr* can also be used to identify malicious traffic, as these counters significantly increase in attack cases. Furthermore, some of the attacks used the same *dns.qry.name* (*root-servers.net* for B_1, B_2, B_3, and *anonsc.com* for B_6, B_7) to generate large DNS packets, thus the hashed *dns.qry.name* may also enhance the accuracy of the resulting classification. Interestingly, *dns.flags. recdesired*, *dns.flags.recavail* and *dns.flags. authoritative* are of high importance for the *Booters*+WIDE-F dataset combination. This follows from the fact that most DNS responses in WIDE-F dataset (benign) were generated by iterative queries on authoritative DNS servers, while in *Booters* (malicious) by recursive queries in non-authoritative servers.

As expected, *dns.flags.response*, *dns.flags.z*, *dns.count.queries*, *dns.qry.class*, *dns.flags.opcode* are of low importance for DNS traffic classification. These had almost the same value for every packet, malicious or benign. In addition, based on our experimental observations the features *dns.flags.authenticated*, *dns.flags.truncated*, *dns.flags.rcode*, *dns.id*, *dns.count.auth_rr* and *dns.flags.checkdisable* do not improve the Out-Of-Bag (*OOB*) score of the RF classifiers and have been removed.

In summary, the proposed packet field (feature) selection algorithm identifies a small set of features out of the 20 initially chosen. These are used to accurately classify both benign and malicious DNS traffic patterns. The classification

TABLE 2. Packet header fields (features) for DNS traffic classification.

Packet Fields	Short Description
ip.length	IP packet size in bytes
udp.length	UDP datagram size in bytes
dns.id	identifies uniquely a DNS transaction
dns.flags.response	specifies whether the message is a query (0) or a response (1)
dns.flags.opcode	specifies the kind of the query e.g. standard DNS query
dns.flags.authoritative	specifies whether the responding DNS server is authoritative (1) or not (0) for the requested domain name
dns.flags.truncated	specifies whether the message is truncated (1) or not (0)
dns.flags.recdesired	specifies whether recursion is desired (1) or not (0)
dns.flags.recavail	specifies whether recursive query support is available (1) in the name server or not (0)
dns.flags.z	reserved field for future use
dns.flags.authenticated	indicates in a response that all data included in the answer and authority portion of the response has been authenticated by the server (1) or not (0)
dns.flags.checkdisable	indicates in a query that non-authenticated data is acceptable to the resolver sending the query (1) or not (0)
dns.flags.rcode	indicates the response code for the specified request e.g. the name server refused to respond
dns.count.queries	number of entries in the question section
dns.count.answers	number of Resource Records (RRs) in the answer section
dns.count.auth_rr	number of name server RRs in the authority records section
dns.count.add_rr	number of RRs in the additional records section
dns.qry.name	variable length field terminated by the zero length byte, specifying the requested domain name
dns.qry.type	specifies the type of the query
dns.qry.class	specifies the class of the query e.g. IN for the Internet class

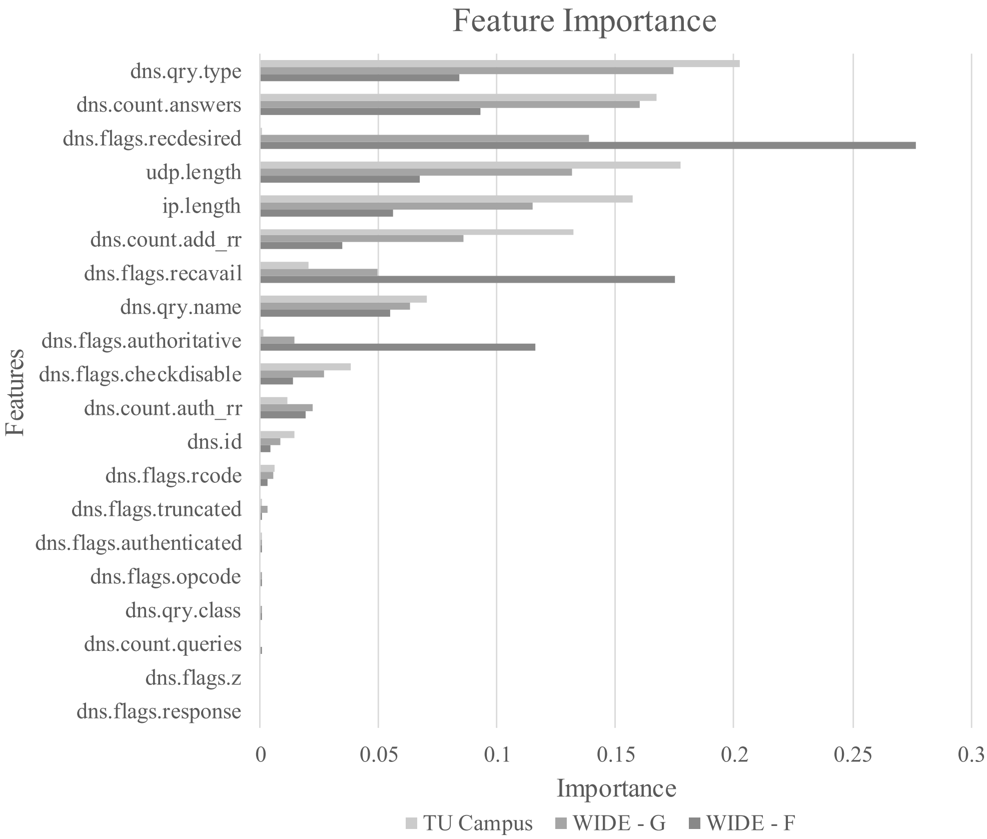

FIGURE 4. Feature importance provided by random forest classifiers for DNS traffic.

TABLE 3. Most important packet fields for DNS traffic classification.

***Booters*+WIDE-G**	***Booters*+WIDE-F**	***Booters*+TU Campus**
dns.qry.type	*dns.flags.recdesired*	*dns.qry.type*
dns.count.answers	*dns.flags.recavail*	*udp.length*
dns.flags.recdesired	*dns.flags.authoritative*	*dns.count.answers*
udp.length	*dns.count.answers*	*ip.length*
ip.length	*dns.qry.type*	*dns.count.add_rr*
dns.count.add_rr	*udp.length*	*dns.qry.name*
dns.qry.name	*ip.length*	-
-	*dns.qry.name*	-

results are based on diverse and realistic traffic scenarios sourced from heterogeneous network environments.

C. SIGNATURE CLASSIFICATION ACCURACY

In this subsection, we evaluate the signature classification accuracy of the proposed mechanism, using two different supervised learning methods. We implemented two classifiers: (i) Random Forests (RF) with 100 decision trees and (ii) an N x $(2N + 1)$ x 1 Multilayer Perceptron (MLP), with sigmoid activation functions, as suggested in [13]; N is the number of features (see Table 3 above). The MLP was trained with examples of batch size equal to 4096 and MLP weights were updated based on *Adam* method [37] with learning rate $\alpha = 0.01$. We used a single epoch with a validation dataset comprising 30% of the training dataset. The training procedure was conducted separately for each unique combination of the following:

- Each classifier (RF, MLP)
- Each benign dataset (WIDE-G, WIDE-F, TU Campus)
- Each set $A_i = \{Booters - B_i\}$, where $i = 1 \ldots 7$, e.g. $A_4 = \{B_1, B_2, B_3, B_5, B_6, B_7\}$

There are 42 different dataset combinations. Each trained model is evaluated against a mix of traffic (test dataset) based on the excluded attack dataset B_i and benign traffic from the same origin (e.g. WIDE-G). Specifically, for WIDE-G and WIDE-F, we employed two 15-minute traces for training and eight 15-minute traces as test dataset. Similarly, for TU Campus we used two 1-hour traces for training and eight 1-hour traces as test dataset respectively.

For MLP we employed undersampling techniques on the attack datasets as they contain more signatures than benign datasets. Training data for MLP were also normalized in the range of [0,1] to enhance classification capabilities. In Fig. 5, we illustrate the *True Negative Rate* (*TNR*) of all

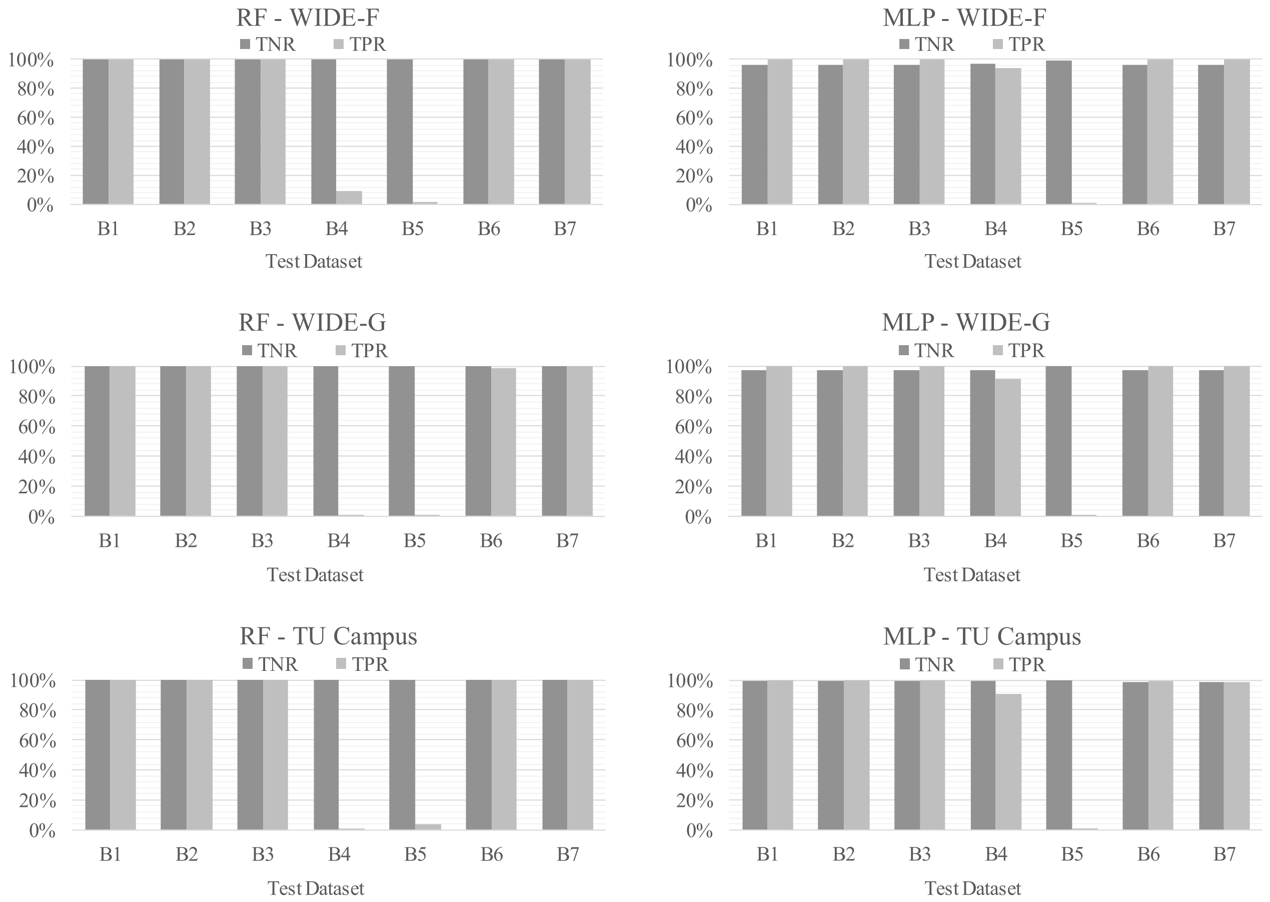

FIGURE 5. True Negative and True Positive Rates for various training scenarios using *Booters* combined with the benign datasets WIDE-F, WIDE-G, and TU Campus.

combinations, which is the percentage of benign traffic that was classified as benign and the *True Positive Rate* (*TPR*), which is the percentage of attack traffic classified as malicious.

As illustrated in Fig. 5, RF is a reliable method to identify both benign (WIDE-G, WIDE-F, TU Campus) and attack traffic (*Booters*) patterns, provided it is trained with diverse attack data. However, RF is not able to recognize attacks that significantly deviate from the training attack pattern. This is clearly illustrated when the model is trained with A_4, which does not include B_4 of the test dataset. Recall that B_4 contains large DNS responses with multiple *type A* RR for a domain name, while the training dataset (A_4) contains attack traces with *type ANY* DNS responses.

Similar to RF, MLP can identify benign and attack traffic with high accuracy for all combinations of training data. However, MLP identified B_4 as an attack, illustrating significant generalization capabilities on detecting "unseen" (zero-day) attacks.

Note that B_5 was not identified by any classifier as an attack trace. As already mentioned it corresponds to a failed attack that did not produce heavy payload, thus exhibiting similarities to benign traffic. Interestingly, all classification mechanisms in our experiments discovered attack data within the benign datasets (WIDE-F, WIDE-G). A closer inspection of the original network traces revealed modest attack traffic, i.e. consecutive *type ANY* responses from specific IP sources to the same destination IP. These data were manually removed and are not included in Fig. 5.

An interesting topic pertaining to ML algorithms are the training and test runtimes. With regards to the former, i.e. training runtime, has limited impact to our mechanism since the training process is conducted offline and the values are in any case in the order of seconds for both models. Qualitatively, training runtimes for MLP were on average 11 times faster than RF. The most important metric for us is the test runtime since it corresponds to real-time signature classification. These values were in the order of milliseconds with MLP runtimes being on average 17 times faster than RFs. Such values are negligible compared to the overall time-window during which our mechanism identifies and mitigates DDoS attacks. This time-window (several seconds) includes packet monitoring, signature classification and filtering rule deployment. To our knowledge, such time-windows are consistent with production solutions offered by major security service providers [38].

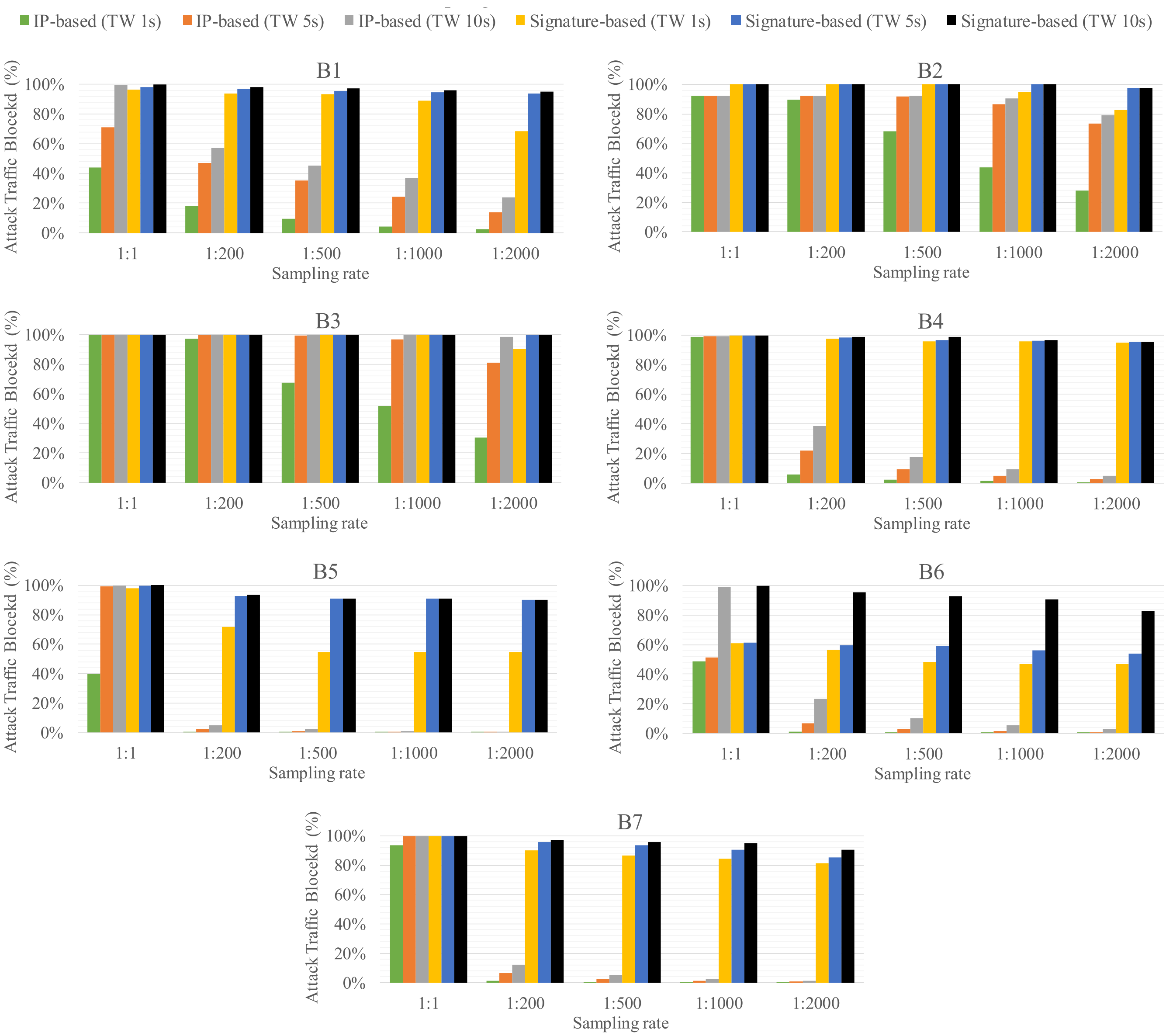

FIGURE 6. Comparison between source-based and signature-based protection mechanisms for *Booters* datasets.

In summary, the proposed approach provides accurate classification of DNS Amplification attacks and benign traffic. This was validated for 42 different training/testing scenarios utilizing real data from heterogeneous network environments. Notably, MLPs achieved detection of ''unseen'' attack traffic patterns (not used in the training process), illustrating better generalization capabilities compared to RF classification algorithms. However, RF is still a reliable classification method, provided that it is trained with diverse attack data.

D. IP-BASED VS SIGNATURE-BASED PROTECTION MECHANISMS

In the following subsections, we compare our signature-based schema to legacy IP-based mechanisms e.g. [13]–[18]. We evaluate both approaches considering their (i) ability to identify and filter malicious traffic, (ii) filtering rules cardinality and (iii) packet filtering performance. These are described in subsections D-1, D-2 and D-3 respectively.

1) MALICIOUS TRAFFIC IDENTIFICATION AND FILTERING

Typically, DDoS protection mechanisms collect monitoring data within time-windows (TW) and utilize them to classify network traffic. Based on this classification, filtering rules are generated and used to block the attack traffic. In this subsection, we compare our signature-based protection mechanism to the optimal IP-based approach, that identifies all malicious IP sources.

In our comparisons, we analyzed network traffic from the first time-window of each attack dataset B_i and extracted the malicious DNS signatures (based on WIDE-F features[1]) and source IP addresses. Subsequently, we calculated from the whole attack dataset B_i the traffic (in bytes) that corresponds to the extracted DNS signatures and IP sources divided by the total attack traffic. This illustrates the percentage of the attack traffic that is dropped by each approach based on monitoring data from the first time-window of the attack. In Fig. 6, we present for every B_i the dropped attack traffic (%) considering various time-windows and packet sampling rates. Short TWs (e.g. 1s) allow for rapid detection and mitigation. Sampling rate 1:1 corresponds to our XDP-based monitoring approach (SE), while lower values correspond to sparse packet sampling, typically employed in monitoring mechanisms e.g. sFlow [39].

Our signature-based approach outperforms the source IP-based alternative for all attack scenarios and combinations of time-windows (TW) and sampling rates. This is attributed to the fact that the attack traffic is characterized by a few amount of DNS signatures, typically distributed to multiple IP addresses. Decreasing the sampling rate significantly reduces the effectiveness of the source-based mechanism especially for highly distributed attacks (e.g. B_1, B_4, B_6, B_7). In contrast, our approach is not affected and is able to successfully block most of the attack traffic (e.g. TW 1s - B_3: 90%) even for the lowest sampling rate 1:2000. As expected, increasing the time-window duration enables both mechanisms to observe more data and thus filter more attack traffic. Notably, our signature-based approach is able to filter a greater portion of the attack traffic (for packet sampling cases lower than 1:1) than the IP-based counterpart, while using data from shorter time-windows (grey bars – IP 10s vs yellow–signatures 1s bars).

In summary, packet signatures are associated with larger amounts of attack packets compared to source IP addresses. This supports the observation that signature-based schemes may provide faster detection and more efficient filtering of DDoS Amplification attacks than conventional source IP-based mechanisms.

2) FILTERING RULES CARDINALITY

In this subsection, the number of filtering rules required by IP-based schemes is compared to our signature-based approach. Specifically, we extracted the total number of unique sources for each *Booter* dataset (B_i) and the DNS signatures (WIDE-F features[2]) that characterize all the malicious traffic. Subsequently, we employed our *Signature Reduction* (SR) component to calculate the reduced number of signatures that can match and block the malicious traffic (DNS signatures - reduced). SR, for all *Booters* and benign datasets combinations, concluded that *dns.qry.name* and *dns.qry.type* could be used to block all the offending traffic without blocking benign traffic portions.

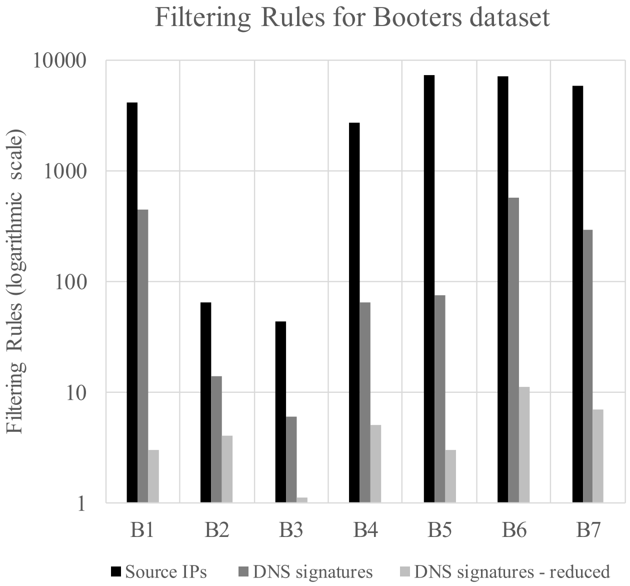

FIGURE 7. Comparison between source-based and signature-based filtering rules for *Booters* datasets.

In Fig. 7, we compare (in logarithmic scale) the number of the source IP filtering rules to the signatures that would be required to fully block the seven DNS attacks of the *Booters* datasets without signature reduction (DNS signatures in Fig. 7) and with signature reduction (DNS signatures – reduced in Fig. 7).

As illustrated in Fig. 7, the number of the required rules is decreased considerably (on average ∼91% for DNS signatures and ∼99% for DNS signatures – reduced). The benefits are: (i) we do not rely on source-based filters that are tough to maintain due to the extremely large cardinality of unique IPs; (ii) we are not affected by dynamic IP changes during an attack, e.g. introduced in case of rotating attackers and (iii) we significantly reduce the memory consumed in the filtering process.

In total, our signature-based approaches require significantly less filtering rules to mitigate the total attack traffic than IP-based alternatives. As mentioned, this benefits our schema since large memory utilization results to increased lookup times in software data planes (BPF Maps - XDP). Hardware-based implementation may also face similar issues due to memory constraints (scarce TCAM resources).

3) MITIGATION PERFORMANCE

In this subsection, the packet filtering performance (throughput) of our approach is compared to source IP filtering alternatives. We implemented three different XDP-based mechanisms: (i) DROP_IP, an IP-based firewall that filters packets based on their source IP address, (ii) AM, that filters packets according to DNS signatures of eight features (WIDE-F features) and (iii) AM_REDUCED, that filters

[1] The total blocked attack traffic using WIDE-G and TU Campus feature sets is on average for all scenarios ∼ 0.06% greater than WIDE-F.

[2] The total number of DNS signatures for all *Booters* using WIDE-G and TU Campus feature sets is on average ∼0.6% less than WIDE-F and thus not included in Fig. 7.

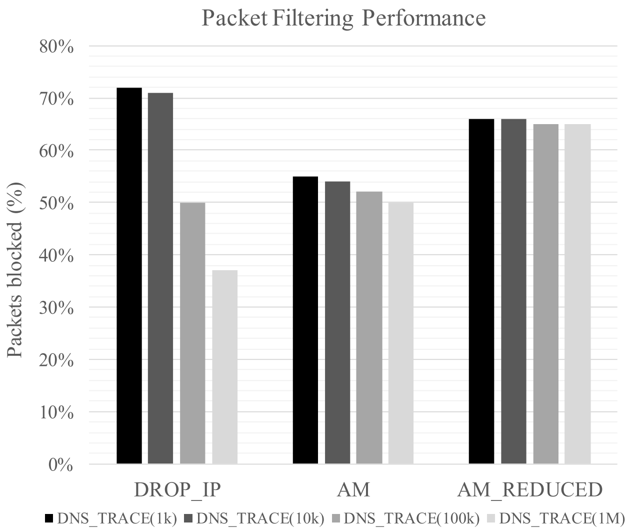

FIGURE 8. Packet throughput for IP-based and signature-based filtering mechanisms.

packets according to DNS signatures (reduced) of two features (*dns.qry.name*, *dns.qry.type*).

For stress testing, we employed synthesized network traces *DNS_TRACE*(n). These contain n unique IP sources, $n/30$ unique combinations of DNS signatures of eight features and $n/850$ unique DNS signatures (reduced) of two features. The proportions were based on the experiments of the previous subsection.[3]

We replayed various synthesized DNS traffic traces at high-speed rates (10 Million packets per second - Mpps) and measured (using the NIC drivers counters [40]) the packets filtered by each XDP mechanism. In Fig. 8, we present the percentage of blocked packets to the transmitted packets for various traffic traces.

DROP_IP performs better than AM and slightly better than AM_REDUCED for the *DNS_TRACE*(1,000) and (10,000); however, it faces scaling issues as the number of IP sources further increases. Specifically, DROP_IP packet processing performance (throughput) decreases from 72% to 37% as the number of IPs increases from 1,000 to 1,000,000. This validates that the number of entries in a BPF Map are significantly affecting its lookup time [3]. In contrast, both our AM and AM_REDUCED are scaling better in terms of packet throughput as the number of sources increases, since few DNS signatures are used to drop the attack traffic. Notably, AM_REDUCED achieves on average ∼10% higher packet processing rate than the AM, presenting the added performance gain provided by reducing the number of DNS signatures. This is mainly attributed to the fewer number of entries contained in the BPF Map and fewer packet fields required to be parsed and processed compared to AM.

Overall, our signature-based approach outperforms the source IP-based alternative due to the fact that the attack traffic can be described by a modest number of signatures. This is even more beneficial in massive attack scenarios where our approach achieves almost two times greater packet filtering performance than IP-based alternatives, utilizing the same set of resources.

VI. CONCLUSION

In this paper we presented an integrated schema for DDoS protection that employs packets signatures for traffic classification and filtering. It leverages on XDP to create performant monitoring and filtering middleboxes, tailored to different attack vectors. These operate either (i) as programmable Deep Packet Inspectors (DPI) to extract monitoring data or (ii) as flexible firewalls that block malicious traffic. Our approach does not rely on IP-sources but employs appropriate traffic signatures. This was based on the widely observed fact that volumetric DDoS attacks, especially UDP-based, may be characterized by a modest number of salient characteristics, thus enabling efficient Machine Learning algorithms (RF, MLP). Note that we did not consider temporal correlations since these may require network data from lengthy time-windows, thus hindering near real-time anomaly detection and mitigation.

In our proof-of-concept, we experimented with benign DNS traffic and malicious DNS Amplification attacks recorded in production network environments. The experimental results were promising and drew interesting conclusions: (i) we were able to automatically identify the most important features for DNS traffic classification for various network traffic profiles; (ii) XDP-based middleboxes were able to expediently monitor and filter network traffic; (iii) RF and MLP illustrated high classification accuracy, with the latter achieving significant generalization capabilities on detecting unknown attacks; (iv) our signature-based approach outperformed traditional IP-based schemes in terms of malicious traffic identification, filtering rules cardinality, and packet processing throughput required in modern high speed networks.

Our experimental evaluation focused on volumetric DNS attacks; however, the proposed approach is based on a generic packet feature selection methodology, and can be seamlessly extended to DDoS Amplification attacks. This follows from the fact that such attacks abuse vulnerable protocols and services in a very specific manner to generate massive amounts of traffic targeting the selected victim. Indicatively, they may exploit messages generated by SNMP *GetBulk*, NTP *monlist* and SSDP *SEARCH* requests [1]. Selecting the most important packet features (i.e. signatures) that are related to the aforementioned attack vectors will enable implementation of protection mechanisms similar to the one proposed in this paper.

As future work, we will consider classification mechanisms that can jointly recognize various attack vectors via multi-task learning techniques as in [41]. Additionally,

[3] Recent DNS Amplification attacks that targeted our University Campus, exhibited a greater proportion of IP attack sources to DNS signatures than the ones mentioned above. Thus we anticipate that our signature-based mitigation mechanism will perform even better with network traffic profiles evolution.

we will investigate application-layer attacks with emphasis on encrypted network traffic [42], [43], which may require state information maintenance. This can be potentially offloaded within XDP, protecting valuable resources in firewalls, routers and hosts. XDP-based middleboxes are modular and may be easily adapted within the NFV paradigm, thus suitable for federated collaborations. Such work will center on collaborative detection and cost-effective mitigation of malicious traffic across network federations, e.g. extending our previous effort [44].

ABBREVIATIONS

Acronym	Definition
AE	Autoencoder.
AM	Anomaly Mitigation.
BPF	Berkeley Packet Filter.
CDN	Content Delivery Network.
CNN	Convolutional Neural Network.
COTS	Commercial-Off-The-Shelf.
CPU	Central Processing Unit.
DDoS	Distributed Denial of Service.
DNS	Domain Name System.
DPDK	Data Plane Development Kit.
ICMP	Internet Control Message Protocol.
IDS	Intrusion Detection System.
IP	Internet Protocol.
IPS	Intrusion Prevention System.
ML	Machine Learning.
MLP	Multilayer Perceptron.
NFV	Network Function Virtualization.
NIC	Network Interface Card.
NSGA-II	Non-dominated Sorting Genetic Algorithm-II.
NTP	Network Time Protocol.
OF	OpenFlow.
OOB	Out-of-Bag.
RAM	Random Access Memory.
RF	Random Forest.
RNN	Recurrent Neural Network.
RR	Resource Record.
SC	Signature Classification.
SDN	Software-Defined Networks.
SE	Signature Extraction.
SNMP	Simple Network Management Protocol.
SOM	Self-Organizing Map.
SR	Signature Reduction.
SSDP	Simple Service Discovery Protocol.
SVM	Support Vector Machine.
TNR	True Negative Rate.
TPR	True Positive Rate.
TU	Thapar University.
TW	Time-Window.
UDP	User Datagram Protocol.
VM	Virtual Machine.
WIDE	Widely Integrated Distributed Environment.
XDP	eXpress Data Path.

REFERENCES

[1] C. Rossow, "Amplification hell: Revisiting network protocols for DDoS abuse," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, May 2014, pp. 1–15, doi: 10.14722/ndss.2014.23233.

[2] *Global DDoS Summary—NETSCOUT Cyber Threat Horizon*. Accessed: Aug. 12, 2021. [Online]. Available: https://horizon.netscout.com/?atlas=summary

[3] S. Miano, R. Doriguzzi-Corin, F. Risso, D. Siracusa, and R. Sommese, "Introducing SmartNICs in server-based data plane processing: The DDoS mitigation use case," *IEEE Access*, vol. 7, pp. 107161–107170, Aug. 2019, doi: 10.1109/access.2019.2933491.

[4] F. Soldo, K. Argyraki, and A. Markopoulou, "Optimal source-based filtering of malicious traffic," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 381–395, Apr. 2012, doi: 10.1109/TNET.2011.2161615.

[5] M. Dimolianis, A. Pavlidis, and V. Maglaris, "SYN flood attack detection and mitigation using machine learning traffic classification and programmable data plane filtering," in *Proc. 24th Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Mar. 2021, pp. 126–133, doi: 10.1109/ICIN51074.2021.9385540.

[6] 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, pp. 87–95, Jul. 2014, doi: 10.1145/2656877.2656890.

[7] *Tofino 2 Barefoot*. Accessed: Aug. 12, 2021. [Online]. Available: https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch/tofino-2-series.html

[8] A. C. Lapolli, J. A. Marques, and L. P. Gaspary, "Offloading real-time DDoS attack detection to programmable data planes," in *Proc. IFIP/IEEE Symp. Integr. Netw. Manage.*, Apr. 2019, pp. 19–27.

[9] M. Dimolianis, A. Pavlidis, and V. Maglaris, "A multi-feature DDoS detection schema on P4 network hardware," in *Proc. 23rd Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2020, pp. 1–6, doi: 10.1109/ICIN48450.2020.9059327.

[10] D. Ding, M. Savi, F. Pederzolli, M. Campanella, and D. Siracusa, "In-network volumetric DDoS victim identification using programmable commodity switches," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1191–1202, Jun. 2021, doi: 10.1109/TNSM.2021.3073597.

[11] T. Høiland-Jørgensen, J. D. Brouer, D. Borkmann, J. Fastabend, T. Herbert, D. Ahern, and D. Miller, "The eXpress data path: Fast programmable packet processing in the operating system kernel," in *Proc. 14th Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2018, pp. 54–66, doi: 10.1145/3281411.3281443.

[12] *Netronome Agilio SmartNICs*. Accessed: Aug. 12, 2021. [Online]. Available: https://www.netronome.com/products/agilio-cx/

[13] C. Siaterlis and V. Maglaris, "Detecting incoming and outgoing DDoS attacks at the edge using a single set of network characteristics," in *Proc. 10th IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2005, pp. 469–475, doi: 10.1109/ISCC.2005.50.

[14] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE Local Comput. Netw. Conf.*, Oct. 2010, pp. 408–415, doi: 10.1109/LCN.2010.5735752.

[15] Y. Cui, L. Yan, S. Li, H. Xing, W. Pan, J. Zhu, and X. Zheng, "SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks," *J. Netw. Comput. Appl.*, vol. 68, pp. 65–79, Jun. 2016.

[16] Q. Niyaz, W. Sun, and A. Y Javaid, "A deep learning based DDoS detection system in software-defined networking (SDN)," 2016, *arXiv:1611.07400*. [Online]. Available: http://arxiv.org/abs/1611.07400

[17] G. Kostas, A. George, and M. Vasilis, "A scalable anomaly detection and mitigation architecture for legacy networks via an OpenFlow middlebox," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 1958–1970, 2016, doi: 10.1002/sec.1368.

[18] A. Santos Da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, "ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2016, pp. 27–35, doi: 10.1109/NOMS.2016.7502793.

[19] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del-Rincon, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 876–889, Jun. 2020, doi: 10.1109/TNSM.2020.2971776.

[20] *eBPF and XDP Suricata*. Accessed: Aug. 12, 2021. [Online]. Available: https://suricata.readthedocs.io/en/latest/capture-hardware/ebpf-xdp.html

[21] *FastNetMon DDoS Detection Tool*. Accessed: Aug. 12, 2021. [Online]. Available: https://fastnetmon.com/

[22] Y. Afek, A. Bremler-Barr, and S. L. Feibish, "Zero-day signature extraction for high-volume attacks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 691–706, Apr. 2019, doi: 10.1109/TNET.2019.2899124.

[23] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2017, pp. 1–8, doi: 10.1109/SMARTCOMP.2017.7946998.

[24] G. Bertin. *Introducing the p0f BPF Compiler*. Accessed: Aug. 12, 2021. [Online]. Available: https://blog.cloudflare.com/introducing-the-p0f-bpf-compiler/

[25] L. Serodio, *Traffic Diversion Techniques for DDoS Mitigation Using BGP Flowspec Distributed Denial of Service (DDoS) Attacks*. Accessed: Aug. 12, 2021. [Online]. Available: https://archive.nanog.org/sites/default/files/wed.general.trafficdiversion.serodio.10.pdf

[26] J. J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Z. Granville, and A. Pras, "Booters—An analysis of DDoS-as-a-service attacks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 243–251, doi: 10.1109/INM.2015.7140298.

[27] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[28] *What is Out of Bag (OOB) Score in Random Forest*. Accessed: Aug. 12, 2021. [Online]. Available: https://towardsdatascience.com/what-is-out-of-bag-oob-score-in-random-forest-a7fa23d710

[29] *RandomForestclassifier Scikit-Learn*. Accessed: Aug. 12, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[30] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognit. Lett.*, vol. 31, no. 14, pp. 2225–2236, Oct. 2010, doi: 10.1016/j.patrec.2010.03.014.

[31] P. Phaal, S. Panchen, and N. McKee. *RFC 3176 - InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks*. Accessed: Aug. 12, 2021. [Online]. Available: https://tools.ietf.org/html/rfc3176

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[33] *Network-Layer DDoS Attack Trends for Q4 2020*. Accessed: Aug. 12, 2021. [Online]. Available: https://blog.cloudflare.com/network-layer-ddos-attack-trends-for-q4-2020/

[34] *Platypus Multiobjective Optimization in Python*. Accessed: Aug. 12, 2021. [Online]. Available: https://platypus.readthedocs.io/en/latest/

[35] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," in *Proc. Annu. Conf. USENIX Annu. Tech. Conf.*, Jun. 2000, pp. 263–270. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/1267724.1267775

[36] M. Singh, M. Singh, and S. Kaur. *10 Days DNS Network Traffic From April-May 2016*. Accessed: Aug. 12, 2021. [Online]. Available: https://data.mendeley.com/datasets/zh3wnddzxy/1

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, May 2014, pp. 1–15.

[38] *Imperva DDoS Mitigation SLA*. Accessed: Aug. 12, 2021. [Online]. Available: https://www.imperva.com/blog/new-3-second-ddos-mitigation-sla-is-3x-faster-and-the-industrys-best/

[39] *sFlow: Sampling Rates*. [Online]. Available: https://blog.sflow.com/2009/06/sampling-rates.html

[40] *Netronome Flow Processor (NFP) Kernel Drivers*. Accessed: Aug. 12, 2021. [Online]. Available: https://www.kernel.org/doc/html/latest/networking/device_drivers/ethernet/netronome/nfp.html

[41] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Toward effective mobile encrypted traffic classification through deep learning," *Neurocomputing*, vol. 409, pp. 306–315, Oct. 2020, doi: 10.1016/j.neucom.2020.05.036.

[42] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 445–458, Feb. 2019, doi: 10.1109/TNSM.2019.2899085.

[43] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "FS-Net: A flow sequence network for encrypted traffic classification," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1171–1179, doi: 10.1109/INFOCOM.2019.8737507.

[44] A. Pavlidis, M. Dimolianis, K. Giotis, L. Anagnostou, N. Kostopoulos, T. Tsigkritis, I. Kotinas, D. Kalogeras, and V. Maglaris, "Orchestrating DDoS mitigation via blockchain-based network provider collaborations," *Knowl. Eng. Rev.*, vol. 35, pp. 1–17, Apr. 2020, doi: 10.1017/S0269888920000259.

MARINOS DIMOLIANIS (Graduate Student Member, IEEE) was born in Athens, Greece, in 1993. He received the Engineering degree from the National Technical University of Athens (NTUA), Athens, in 2017, where he is currently pursuing the Ph.D. degree. He has proven experience both in the research industry and operational network environments. His research interests include computer networks and network programmability to network security and intelligent network management.

ADAM PAVLIDIS was born in Athens, Greece, in 1991. He received the Engineering and Ph.D. degrees from the National Technical University of Athens (NTUA), Athens, in 2015 and 2020, respectively. He has involved in multiple national and European projects. He has also considerable experience in designing and managing production network environments. His current research interests include novel technologies related to network monitoring, software-defined networking, and security applicable to large-scale network infrastructures.

VASILIS MAGLARIS was born in Athens, Greece, in 1952. He received the Engineering degree from the National Technical University of Athens (NTUA), Athens, in 1974, and the Ph.D. degree from Columbia University, New York, NY, USA, in 1979. From 1979 to 1981, he was with the Network Analysis Corporation, Great Neck, NY, USA, working on electronic communications advanced projects. From 1981 to 1989, he was with the Faculty of Electrical Engineering and Computer Science, Polytechnic University, now part of New York University, New York, NY. In 1989, he joined as a Faculty Member with the School of Electrical and Computer Engineering, NTUA, teaching and performing research on Internet technologies. In 2020, the NTUA Senate conferred upon him the title of Professor Emeritus, enabling him to continue his teaching and research activities beyond his retirement, in 2019.