Bringing Intelligence to Software Defined Networks: Mitigating DDoS Attacks

Zakaria Abou El Houda, Lyes Khoukhi, and Abdelhakim Senhaji Hafid

***Abstract*—As one of the most devastating types of Distributed Denial of Service (DDoS) attacks, Domain Name System (DNS) amplification attack represents a big threat and one of the main Internet security problems to nowadays networks. Many protocols that form the Internet infrastructure expose a set of vulnerabilities that can be exploited by attackers to carry out a set of attacks. DNS, one of the most critical elements of the Internet, is among these protocols. It is vulnerable to DDoS attacks mainly because all exchanges in this protocol use User Datagram Protocol (UDP). These attacks are difficult to defeat because attackers spoof the IP address of the victim and flood him with valid DNS responses coming from legitimate DNS servers. In this paper, we propose an efficient and scalable solution, called WisdomSDN, to effectively mitigate DNS amplification attack in the context of software defined networks (SDN). WisdomSDN covers both detection and mitigation of illegitimate DNS requests and responses. WisdomSDN consists of: (1) a novel proactive and stateful scheme (PAS) to perform one-to-one mapping between DNS requests and DNS responses; it operates proactively by sending only legitimate responses, excluding amplified illegitimate DNS responses; (2) a machine learning DDoS detection module to detect, in real-time, illegitimate DNS requests. This module consists of (a) Flow statistics collection scheme (FSC) to gather the features of flows in an efficient and scalable way using sFlow protocol; (b) Entropy calculation scheme (ECS) to measure randomness of network traffic; and (c) Bayes Network based Filtering scheme (BNF) to classify, based on entropy values, illegitimate DNS requests; and (3) DNS Mitigation scheme (DM) to effectively mitigate illegitimate DNS requests. The experimental results show that, compared to state-of-art, WisdomSDN can effectively detect/mitigate DNS amplification attack quickly with high detection rate, less false positive rate, and low overhead making it a promising solution to mitigate DNS amplification attack in a SDN environment.**

***Index Terms*—DDoS; SDN; Entropy; Bayes Classifier.**

I. Introduction

A. Overview

DNS amplification attack is a popular form of DDoS attacks that relies on the use of Open Resolver (publically accessible DNS servers) to overwhelm the victim (i.e., target of DNS amplification attack) with amplified DNS traffic. This attack is based on a recursive function of DNS servers. Usually, the DNS server accepts and responds to resolution requests from anyone without verifying its identity. Thus, attackers can exploit recursive functions to amplify the attack by spoofing the victim's IP address. The spoofed queries (i.e., DNS requests) sent by the attacker are of type ANY; they include all known information about a DNS zone in a single request. The amplification impact of this attack comes from the fact that small queries can generate massive amounts of UDP packets in response. This category of attack can be divided in two types: (a) amplification with repeated DNS requests that have the same content; and (b) amplification with varied DNS requests that have different contents. The query can be of type ANY that requests all records for a particular domain or different domains. The size of the response may be large to produce a high level of amplification. The amplification ratio, of a factor up to 4670 [1], is calculated as the ratio between the response size and the request size. According to a recent study, there are about 7.5 million external DNS servers in the Internet; more than 75% of these servers allow recursive name service to the public [2]. This can cause significant collateral damage on the victim, if attackers use many recursive servers to amplify and generate the attack. As example, on the 2nd of October 2016, a huge attack was conducted against the servers of Dyn, a company that controls many Internets DNS servers. As a consequence, many popular Internet services, e.g., Amazon, Twitter, GitHub [3], PayPal and others became unavailable for several hours [4]. This attack [4] is considered as the largest ever DDoS attack, exceeding a rate of 1 Tbit/s. Such incidents harm Internet service providers (ISPs) and cost millions of dollars of lost revenues for enterprises.

B. Description of DNS amplification attack

DNS amplification attack consists of: (1) an attacker (e.g., bot master); (2) a large number of compromised devices (called zombies); and (3) reflectors (i.e., Open Resolvers). Each zombie is ordered by the bot master to send a large number of DNS requests, in which the source IP address is replaced with the victim's IP address (i.e., spoofed), to Open Resolvers. Upon receipt of these illegitimate DNS requests, Open Resolvers make a recursive resolution and flood the victim with large number of amplified DNS responses (see Fig. 1).

In this paper, we propose an efficient, stateful, proactive and scalable solution in the context of SDN, called WisdomSDN. Recently, SDN has attracted tremendous attention from industry and academia as an emerging networking paradigm that facilitates network management and provides new approaches to manage and deploy networks dynamically [5]–[9]. SDN separates data and control planes; this separation allows for more control over the network and brings a new way to deal

Z. A. El Houda is with ICD/ERA, University of Technology of Troyes, France, and NRL, Department of Computer Science and Operational Research, University of Montreal, Canada, email:(zakaria.abou.el.houda@umontreal.ca).

L. Khoukhi is with GREYC, ENSICAEN, France, since September 2020, email:(lyes.khoukhi@ensicaen.fr).

A. S. Hafid is with NRL, Department of Computer Science and Operational Research, University of Montreal, Canada, e-mail:(ahafid@iro.umontreal.ca).

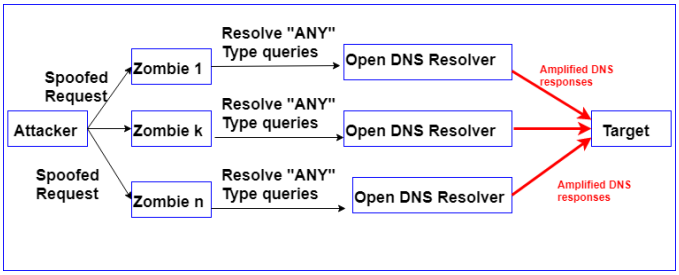


Fig. 1: DNS amplification attack.

with various forms of DDoS attacks [10]–[16]. While SDN can protect the network from DDoS attacks [17]–[22], it can be a victim of these attacks [23]. To address this problem, WisdomSDN promises to leverage the advantages of SDN to protect the victim from DNS amplification attack while maintaining the SDN secure i.e., protecting the resources of data plane (i.e., Ternary Content Addressable Memory (TCAM) of OF (OpenFlow) switches), the resources of control plane (i.e., the SDN controller resources) and the workload of OF channel. To this aim, WisdomSDN makes use of a novel proactive and stateful mapping scheme (PAS), based on in-OF switch (e.g., OpenvSwitch [24]) processing capabilities, to mitigate DNS amplification attack and protect SDN controller. PAS adopts a proactive and stateful paradigm to perform one-to-one mapping between DNS requests and DNS responses; this can effectively offload the SDN controller resources and the OF channel. Each OF switch filters DNS traffic according to the header fields (i.e., MAC address, IP address and UDP port). PAS checks the legitimacy of each DNS response by comparing its MAC address, IP address and UDP Port with the corresponding DNS request and drops automatically illegitimate DNS responses. Thus, PAS allows: (1) OF switches to be smart enough to react proactively and quickly to mitigate illegitimate DNS responses, and not wait for a reactive rule from the SDN controller; and (2) to effectively offload the control plane (i.e., SDN controller). However, if each OF switch maintains the one-to-one mapping of all DNS requests that it receives it will be overwhelmed. To address this issue and protect TCAM of OF switches that is limited in size, WisdomSDN makes use of a robust machine learning DDoS detection module that aims to detect, in real-time, illegitimate DNS requests. This module consists of: (1) FSC, to gather the features of flows in an efficient and scalable way using sFlow protocol; (2) ECS, to measure the disorder/randomness of network traffic; and (3) BNF, to automatically detect illegitimate DNS requests. To evaluate the effectiveness of WisdomSDN, we conducted a set of experiments in Mininet [25]. We launch the attack on a simulated SDN network environment in the context of 2 scenarios: (1) without WisdomSDN; and (2) with WisdomSDN. The results show that without WisdomSDN, the SDN controller and victim are flooded with illegitimate DNS traffic (DNS requests and DNS responses). However, with WisdomSDN, OF switches can effectively detect and mitigate illegitimate DNS requests and DNS responses. Also, we evaluated BNF through Receiver Operating Characteristic (ROC) curves and we compared it to the most prominent state-of-art schemes. The results show that BNF can effectively detect the attack with high detection rate and low false positive rate. To the best of our knowledge, WisdomSDN is the first contribution that mitigates this attack considering all possible attacks setup while maintaining the SDN secure.

The main contributions of this paper can be summarized as follows:

- We propose a novel proactive and stateful scheme (PAS) to perform one-to-one mapping between DNS requests and DNS responses.
- We propose a flow statistics collection scheme (FSC) to gather the features of flows in an efficient way using sFlow protocol.
- We introduce an entropy calculation scheme (ECS) to measure the disorder/randomness of network traffic.
- We propose a Network based Filtering scheme (BNF) to classify, based on entropy values, illegitimate DNS requests.
- We propose a DNS Mitigation scheme (DM) to effectively mitigate illegitimate DNS requests.
- We evaluate the performance of WisdomSDN in terms of scalability, effectiveness and efficiency. The experiments results show that WisdomSDN can effectively mitigate DNS amplification attack with high detection rate, low false positive rate and minor overhead.

The rest of this paper is organized as follows. Section II presents related works. Section III presents an overview of WisdomSDN. Section IV introduces our system design. Section V presents PAS, our proactive and stateful scheme. Section VI presents our machine learning DDoS detection module that consists of: FSC, ECS, and BNF. Section VII describes DM. Section VIII evaluates WisdomSDN in terms of efficiency, scalability, detection rate and presents the simulation results. Finally, section IX concludes the paper and presents our future works.

II. Related Works

Several schemes have been proposed in the literature to mitigate DNS amplification attack. In the following: (1) we classify countermeasures into two groups; and (2) for each group, we present some of the most prominent works as well as their limitations.

A. Countermeasures in Legacy Networks

In [26], Huistra proposed a scheme to detect malicious DNS traffic by detecting all IP addresses that cause the attack, based on collected dataset from NetFlow. The scheme consists of two phases: (1) detection of suspicious IP addresses based on the quantity of requests generated by this IP address and stored in the flow-record; and (2) detection of any IP address that receives suspicious DNS responses based on the huge amount of received responses. However, the execution of the two phases, in this scheme [26], result in large response times. In [27], Rozekrans et al. proposed a defense mechanism, called response rate limiting (RRL), to limit the amount of generated responses by dropping the ones that exceed a predefined

threshold. This is performed by storing the requestor IP address when DNS server generates a response for a DNS request. When the number of responses exceeds the threshold, the server drops requests for this IP address. However, RRL only examines DNS responses and ignores the amount of incoming DNS requests. In [28], Sun et al. proposed a low-cost hardware based scheme to mitigate DNS amplification attack. The solution works well except that it is hardware-based, which makes it hard to update and extend. In [29], Guo et al. proposed a scheme that deploys filters at the border of networks in order to block incoming source IP addresses that are not belong to their networks. However, this scheme has "neighborhood policy" that requires all ISPs to participate in order to provide the complete list of IP addresses that do not sent from the network; moreover, the effectiveness of this scheme depends on its global deployment across the Internet. In [30], Kambourakis et al. proposed a scheme that stores all incoming DNS requests and DNS responses. Once an illegitimate DNS response is detected, a counter is incremented until it reaches a threshold. When the threshold is reached, an alert is generated and the attack is assumed to have happened. This scheme does not scale for large scale networks because it needs to store all DNS requests and responses.

B. SDN-Based Countermeasures

Our previous works [31], [32] did show their effectiveness in protecting permissioned blockchain from DNS amplification attack. In this paper, we extend these works to protect any type of applications and not consider only blockchain applications. Moreover, we combine a proactive and stateful scheme with a machine learning algorithm (i.e., BNF) in order to: (1) distinguish between legitimate and illegitimate responses and systematically eliminate the amplified illegitimate DNS responses; (2) decrease false positive rate while maintaining a high detection rate. In [33], Rodrigo et al. proposed a flow-based intrusion detection scheme using OF protocol to gather network traffic features. This scheme [33] focuses only on the attack in data plane without any analysis of the overhead to the control plane. Moreover, the performance analysis does not include the overall system performance. In [34], Mehdi et al. proposed an anomaly detection scheme in the context of SDN using OF protocol. However, this scheme [34] was designed only for small-scale setup; in large-scale environment, a high rate traffic from data plane to control plane may overload the SDN controller. In [35], Wang et al. proposed an entropy scheme based on OF switches; it focuses only on detection, but it cannot find the victim or the illegitimate hosts. In [36], Lim et al. proposed a DDoS attacks mitigation scheme for botnet-based attacks that runs on SDN controller. This scheme [36] requires a large amount of communications between the SDN controller and OF switches in order to protect the victim; moreover, it not only generates DDoS attacks against the SDN controller but also requires high latency to cooperate with the SDN controller. In [37], Zaalouk et al. proposed a scheme based on SDN to mitigate DNS amplification attack; it uses sFlow protocol to monitor DNS traffic. When the attack is detected, the orchestrator commands one of the SDN controllers to forward suspicious traffic to it in order to analyze the size of DNS response packets and to compute their average size. If the average of responses size exceeds the value of a threshold, it proceeds to the second phase of detection. In this phase, the orchestrator calculates the entropy of destination IP address. If the entropy value is low, then it is assumed that there is a DNS amplification attack. In this case, it applies a set of rules to limit the responses rate. However, this scheme [37] does not distinguish between legitimate and illegitimate responses since all DNS responses are sent to SDN controller and may cause DDoS attack against control plane as well as orchestrator. In [38], K. Giotis et al. proposed the use of sFlow protocol with OF protocol in order to detect DDoS attacks reducing the communication overhead between data plane and control plane. This scheme [38] works well; however, it has high false positive rate.

To address the weaknesses of these existing schemes [26]–[38], we propose WisdomSDN, an efficient, stateful, proactive, and scalable solution to detect and mitigate DNS amplification attack. In WisdomSDN, we use: (1) PAS, to exclude the amplified DNS responses; (2) a machine learning DDoS detection module, to detect, in real-time, illegitimate DNS requests; and (3) DM, to effectively mitigate illegitimate DNS requests. WisdomSDN uses FSC to separate flow monitoring from the forwarding logic; this makes it much more scalable compared to existing native OF schemes [33]–[36]. Using BNF, WisdomSDN is much accurate in comparison with the ones using sampling technology [31], [37], [38].

III. WisdomSDN: An Overview

In this section, we present an overview of WisdomSDN. More specifically, we explain how WisdomSDN can combine detection and mitigation of DNS amplification attack, allowing for a robust detection and an effective mitigation of this attack.

The majority of DNS requests use UDP as a transport protocol without providing any mechanism to verify the source IP address of DNS requests. Therefore, the network can be flooded with illegitimate DNS requests and amplified DNS responses. WisdomSDN is inspired from the techniques of Moving Target Defense [39]. These techniques allow to contain the attack in the space of the real source of the request and avoid propagating DDoS attacks to the potential victim.

Fig. 2 shows the flow diagram of WisdomSDN. When an OF switch receives a new packet, it first checks the type of incoming packet. If the packet is DNS request; then, WisdomSDN checks whether the maximum number of requests threshold per ingress port is reached. If the response is yes, it triggers, using sFlow protocol, an event to sFlow collector (i.e., sFlow-RT [40]) in order to collect network traffic features using FSC. Afterwards, ECS extracts network traffic features from collected information and calculates entropy values. Based on this calculation, BNF detects automatically illegitimate flows (see Section VI). If the flow is classified as illegitimate, then a mitigation action is performed using DM scheme. Otherwise, OF switch: (1) learns the authorized response from ingress port of incoming DNS request; and (2) installs the rule that allows only the DNS response that matches the corresponding

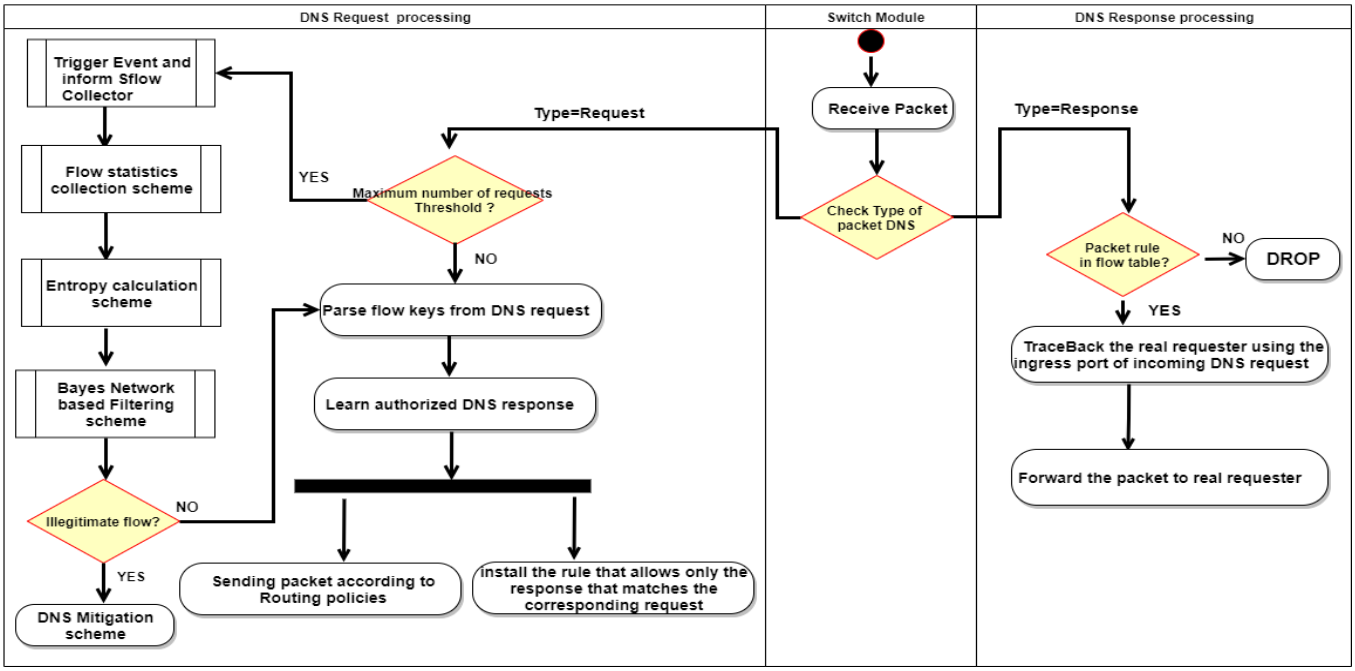


Fig. 2: Flow diagram of WisdomSDN.

DNS request with a short timeout (i.e., *idle timeout and hard timeout*) in order to avoid the storage complexity to maintain the one-to-one mapping. Thus, any amplified illegitimate DNS response will not be sent to neither the victim nor the SDN controller; this excludes amplified illegitimate DNS responses and offload the SDN controller and the OF channel. Upon receipt of DNS response, OF switch checks whether there is any corresponding DNS request; if it is the case, it forwards the DNS response following the ingress port of incoming DNS request; otherwise, it drops the DNS response.

IV. System Design

A. Design Overview

When designing WisdomSDN, we did consider the following objectives. First, WisdomSDN should give a full protection from DNS amplification attack. Unlike existing schemes [26]–[38] that try to analyze the network state; then, detect the attack. WisdomSDN aims to act proactively, by maintaining one-to-one mapping between DNS requests and DNS responses, in order to avoid sending illegitimate DNS traffic to victim (i.e., target of the attack); this is ensured via PAS. To protect TCAM of OF switches which is limited in size, WisdomSDN makes use of a robust machine learning DDoS detection module that aims to detect, in real-time, illegitimate DNS requests. Finally, the attack should be effectively mitigated, using DM, and the whole system has to be as scalable as possible.

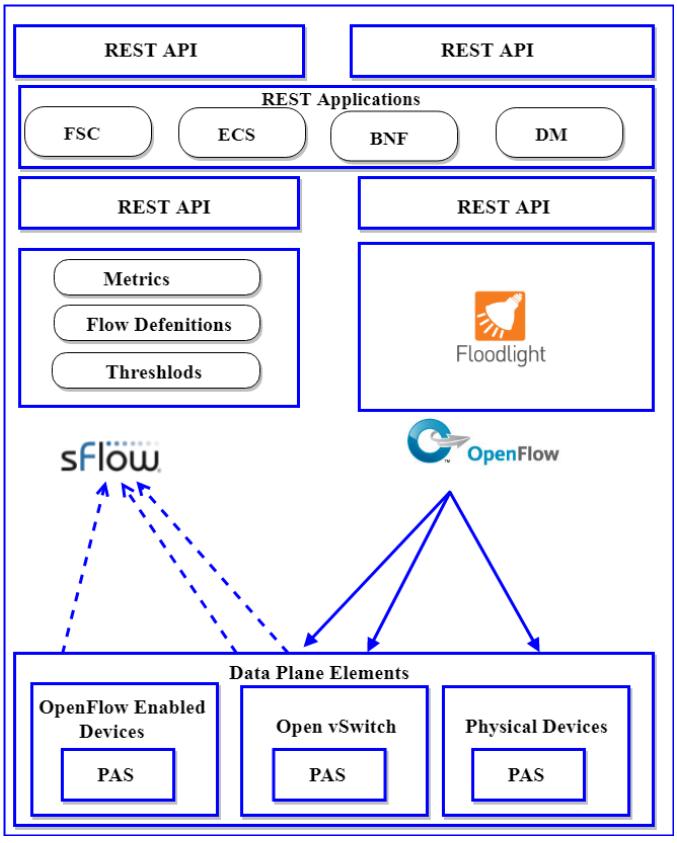


Fig. 3: System Architecture.

B. System Architecture

Fig. 3 shows the architecture of WisdomSDN. WisdomSDN has three phases: (1) PAS, a novel proactive and stateful scheme to perform one-to-one mapping between DNS requests and DNS responses; it operates proactively by sending only legitimate responses, excluding amplified illegitimate DNS responses. PAS is implemented in data plane (i.e., OF switch); (2) a machine learning DDoS detection module that aims

to detect, in real-time, illegitimate DNS requests. This module is implemented on the top of the SDN controller (i.e., application plane) and consists of (a) FSC, a novel Flow statistics collection scheme to monitor, using sFlow protocol, network traffic features in an efficient and scalable way; it defines the monitoring metrics (e.g., the attributes of flows aggregation and thresholds); (b) ECS, an entropy calculation scheme to measure disorder/randomness of network traffic; and (c) BNF, a real-time detection scheme, to classify, based on entropy values, illegitimate DNS requests; (3) DM, a DNS mitigation scheme to effectively mitigate illegitimate DNS requests enabling the network to recover quickly in short time. DM installs new OF rules into OF switches under attack in order to monitor the speed of illegitimate DNS requests. WisdomSDN separates flow monitoring from the forwarding logic; this makes it much scalable and more efficient than OF native schemes. The REST API [41] is used in the process of detection/mitigation to offer the interoperability in order to manage any SDN controller (e.g., OpenDaylight [42], Floodlight [43]).

V. Proactive And Stateful Scheme (PAS)

PAS is a novel proactive and stateful scheme that is inspired by the techniques of one-to-one mapping between DNS requests and DNS responses [28]. First, the SDN controller pushes the OF rules (see Algorithm 1) to OF switches in order to process proactively all DNS traffic (i.e., DNS requests and DNS responses). In PAS, a DNS response is considered as legitimate if it has the same reversed fields' values (i.e., MAC addresses, IP addresses, and UDP ports) of a pre-sent DNS request; otherwise, this DNS response will be considered as illegitimate and systematically eliminated. This allows OF switches to be smart enough to react quickly, to avoid any attempt of external DNS amplification attack that aims to flood the victim's network with amplified DNS traffic, and not wait for a reactive rule from the SDN controller. However, when an attacker is within the victim's network (see Figs. 4(c), and 4(d)), he can easily spoof the source IP address of the victim in order to direct the amplified DNS responses to that victim. To alleviate this issue, each DNS response received, by each OF switch, is forwarded to the original port (i.e., ingress port) from which the corresponding DNS request came. Then, if the attacker spoofed the source IP address in the prior DNS request, he will receive the returned amplified DNS traffic (see Figs. 5(b), 5(c), and 5(d)). Otherwise, legitimate sources will receive legitimate DNS responses. Thus, PAS totally ensures the protection of the victim from any external or internal DNS amplification attack. PAS uses short timeout (i.e., *idle timeout and hard timeout*) in order to avoid the storage complexity to maintain the one-to-one mapping. Algorithm 1 illustrates steps executed by each OF switch upon receipt of a packet. Fig. 5(a) shows that with WisdomSDN and since there is no DNS request that is sent from a host in the network, the default rule (Proto=UDP, port_src=53, prior=0, action=DROP) is triggered in order to drop illegitimate DNS responses; unlike without WisdomSDN, where the victim is flooded with illegitimate amplified DNS responses (see Fig. 4(a)). Figs. 5(b), 5(c) and 5(d) show that PAS traces the real path of the DNS requests provenance, using Ingres Port in the OF switch, and sends the corresponding responses using the same path as the requests. Thus, the attacker will receive the returned amplified DNS responses, and the victim will not receive any illegitimate DNS traffic; unlike without WisdomSDN, where the victim is flooded with illegitimate amplified DNS responses (see Figs. 4(a), 4(b), 4(c) and 4(d)). To protect TCAM of OF switches, which can be the target for attackers, from the huge amount of DNS requests, we propose a machine learning DDoS detection module that consists of FSC, ECS and BNF. In what follows, we investigate how this module can effectively detect illegitimate DNS requests.

Algorithm 1 (PAS): Proactive And Stateful Scheme

```
Input  : DNS packet
Output: Action to carry out
for each DNS packet do
    Check Type of packet DNS
     if this.dns_packet.type==Request then
         Expect_solution=new Packet()
         Expect_solution.eth_src ←this.dns_packet.eth_dst
         Expect_solution.eth_dst←this.dns_packet.eth_src
         Expect_solution.ip_src←this.dns_packet.ip_dst
         Expect_solution.ip_dst←this.dns_packet.ip_src
         Expect_solution.udp_dst←this.dns_packet.udp_src
         Install rule that allows only this Expect_solution
         This.dns_packet::Forward_Routing_Policies()
    else
         if this.dns_packet.type==Response then
              if Match(this.dns_packet)::In_Flow_Table then
                   This.dns_packet::Action(Output:IN_PORT)
              else
                   This.dns_packet::ACTION(DROP)
              end
         else
              return;
         end
    end
end
```

VI. Machine Learning DDoS Detection Module

This module aims to protect TCAM of OF switches while maintaining the SDN secure; it consists of the following: (1) FSC to gather network traffic features in an efficient and scalable way using sFlow protocol; (2) ECS to extract network traffic features; and (3) BNF to, based on ECS calculation, detect network anomalies.

1) FSC: In SDN environment, two commonly approaches are used to collect network traffic features (e.g., count number of received packets). The first approach is based on OF protocol while the second one is based on flow sampling (e.g., sFlow). In OF based approach, collection of network traffic features can be initiated when the control plane (i.e., SDN controller) sends a state request *(ofp_flow_stats_request)* to data plane devices (i.e., OF switches); these latter respond

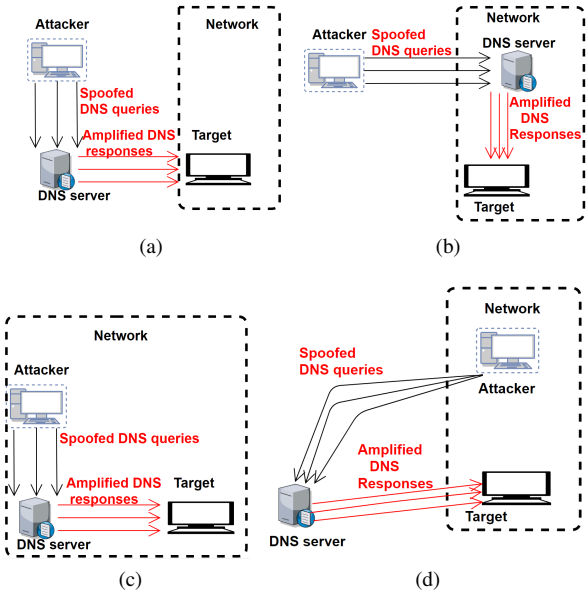


Fig. 4: Different attacks setup without WisdomSDN.

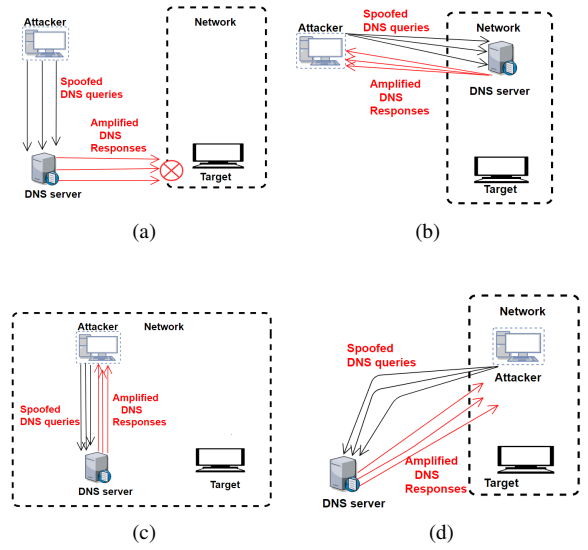


Fig. 5: Different attacks setup with WisdomSDN.

with one or more reply messages *(ofp_flow_stats_reply)* by sending network traffic features. To this aim, each OF switch needs to maintain a large number of flow entries. However, this can exhaust TCAM in OF switches. Moreover, a large size of reply messages sent by OF switches to SDN controller can exhaust the bandwidth between data plane and control plane, congest the OF channel and generate a DDoS attack on control plane. Thus, OF based approach is not efficient to detect high rate attacks (i.e., DNS amplification attack).

To address the issues of OF based approach, we decided to monitor DNS flows based on flow sampling approach using sFlow protocol. This is more scalable and efficient; moreover, it does neither overload the OF channel nor consume bandwidth between control plane and data plane. FSC performs flow aggregation which makes it more adequate to detect high rate DNS amplification attack. In FSC, at each monitoring interval ΔT, the sFlow collector (i.e., sFlow-RT) receives network traffic features from sFlow agents embedded in data plane (i.e., OF switches). Then, ECS computes the entropy values of network traffic. Table 1 shows the list of notations used to describe ECS. In this work, we define a flow as a seven tuple: $\{MAC_{src}, MAC_{dst}, IP_{src}, IP_{dst}, Port_{src}, Port_{dst} = 53, Proto = UDP\}$

TABLE I: Notations.

Notations	Definition
MAC_{src}	The source MAC address of a packet
MAC_{dst}	The destination MAC address of a packet
IP_{src}	The source IP address of a packet
IP_{dst}	The destination IP address of a packet
$Port_{src}$	The UDP source $Port$ of a packet
$Port_{dst}$	The UDP destination $Port$ of a packet
IP_{proto}	The transport protocol of a packet (UDP)
S_j	OF switch S_j
ΔT	The monitoring interval
$F_{i,j}$	Flow f_i at a local OF switch S_j
$p_{i,j}$	The probability of flow f_i over all flows at local OF switch S_j
N	The total number of flows at local OF switch S_j
R	Set of real numbers
I	Set of positive integers

2) ECS: The concept behind ECS comes from Shannon's information theory [44]. ECS is used to extract network traffic features and calculates entropy values of each flow. When the victim's network is under attack, the number of packets for a given flow that have the same source IP address, denoted IP_{src}, will sharply increase causing a more concentrated distribution of source IP address; while normal state leads to a more dispersed probability distribution of source IP address. The higher entropy values, the more dispersed probability distribution of the source IP address, while low entropy values means the concentration of distribution of source IP address. If the source sends similar requests (e.g., requests from the same source IP address IP_{src}), then its entropy will be low; this means that we are very likely in the presence of illegitimate flow. Therefore, we use ECS to measure the changes of network traffic. A flow is characterized by a sequence of packets that have similar properties reaching the same OF switch S_j during each monitoring interval ΔT.

Let $F_{i,j}$ denotes flow f_i at local OF switch S_j; it is defined as follows:

$$F_{i,j}(IP_{src_i}, S_j) = \{< IP_{src_i}, S_j, t > | S_j \in S, i, j \in I, t \in R\} \quad (1)$$

where IP_{src_i} is the source IP address of f_i, t is the current timestamp, and $S = \{S_j, j \in I\}$ are the set of OF switches.

Let $|F_{i,j}(IP_{src_i}, S_j, t)|$ be the count number of packets of flow $F_{i,j}$ at time t. The variation of the number of packets

for flow f_i at local OF switch S_j during ΔT is defined as follows:

$$N_{F_{i,j}}(IP_{src_i}, S_j, t+\Delta T) = |F_{i,j}(IP_{src_i}, S_j, t + \Delta T)| - |F_{i,j}(IP_{src_i}, S_j, t)| \quad (2)$$

The probability $p_{i,j}$ of flow f_i over all flows at local OF switch S_j is expressed as follows:

$$p_{i,j}(IP_{src_i}, S_j) = \frac{N_{F_{i,j}}(IP_{src_i}, S_j, t+\Delta T)}{\sum_{i=1}^{N} N_{F_{i,j}}} \quad (3)$$

where $\sum_{i=1}^{N} p_{i,j}(IP_{src_i}, S_j) = 1$.

Let IP_{src} be the random variable that represents the number of flows during time interval ΔT. We define the entropy of flows at local OF switch S_j as follows:

$$H(IP_{src}) = -\sum_{i=1}^{N} p_{i,j}(IP_{src_i}, S_j) \log_2 p_{i,j}(IP_{src_i}, S_j) \quad (4)$$

Lemma 1: The upper and lower bound of $H(IP_{src})$ are, respectively, 0 and $\log_2 N$ (see inequality (5)).

$$0 \leq H(IP_{src}) \leq \log_2 N \quad (5)$$

Proof: Let $f(x) = \log_2 x, x \geq 0$. $f(x)$ is a monotonically increasing concave function. Let X be the random variable for flow distribution at an OF switch. Applying Jensen's inequality [33] to $f(x)$, we have $Ef(X) \leq f(EX)$. Let $P(X) \doteq \{p_1, p_2, ..., p_n\}$ be the distribution of flows at the OF switch. Then, $\forall\, p_i, 0 \leq p_i \leq 1, \sum_{i=1}^{N} p_i f(x_i) \leq f(\sum_{i=1}^{N} p_i x_i)$, where $\sum_{i=1}^{N} p_i = 1$. Especially $H(IP_{src}) = \sum_{i=1}^{N} p_i \log_2(\frac{1}{p_i}) \leq \log_2(\sum_{i=1}^{N} p_i \frac{1}{p_i}) = \log_2 N$. Then, $H(IP_{src}) \leq \log_2 N$. Since $\log_2 \frac{1}{p_i} \geq 0, \forall\, p_i, 0 \leq p_i \leq 1$. Then, $H(IP_{src}) \geq 0$, and further Eq.5 holds. ■

In order to normalize the entropy values, to have a measurement metric which is totally independent from the number of distinct entropy values, we divide them by the maximum value which is $\log_2 N$, as is demonstrated in Eq. (5). Therefore, the normalized entropy values are in [0, 1] and are defined as follows:

$$H(IP_{src})' = \frac{H(IP_{src})}{\log_2 N} \quad (6)$$

Lemma 2: When the network is under DNS amplification attack, the upper bound of entropy variation at local OF switch S_j decreases sharply in comparison to the normal, non-attack, case.

Proof: As we proved in Eq. (5), the entropy of flows reaches the upper bound, $\log_2 N$, when the probability distribution of source IP address is almost even. The variation number of packets for each flow f_i is almost stable, specifically, $p_1 = p_2 = p_3 = ... = p_N$, and it reaches the lower bound, $H(IP_{src}) = 0$, when the probability distribution of source IP address is almost uneven. Especially, $p_i = 1, 0 \leq i \leq N, p_k = 0, \forall 0 \leq k \leq N, k \neq i$. As the entropy is a monotonic function [44]; therefore, when the victim's network is under DNS amplification attack, the distribution of source IP address moves toward the extreme unbalanced point. As result, the upper bound of the entropy variation decreases sharply. ■

Theorem 1: We divide the state of the network into two segments: normal state (i.e., non-attack), and under DNS amplification attack. We denote $H_{leg}(IP_{src})$ and $H_{illeg}(IP_{src})$ as, respectively, the entropy value of each flow at local OF switch S_j in normal state and under attack. When the attacker starts an attack towards a specific victim (i.e., target of DNS amplification attack), the number of packets that have the same IP_{src} will increase quickly, which leads to a significant decrease of entropy, especially, $H_{leg}(IP_{src}) \gg H_{illeg}(IP_{src})$. High entropy values lead to a dispersed probability distribution of flows, whilst low entropy values indicate a concentrated probability distribution.

Proof: Let $\Omega(x) = x \log_2 x$, $x \geq 0$. $\Omega(x)$ is a monotonically increasing convex function. Let X be the random variable for flow distribution at an OF switch S_j. Applying Jensen's inequality to $\Omega(x)$, we have $E\Omega(X) \geq \Omega(EX)$. Let $\Psi(X) \doteq \{\Psi_1, \Psi_2, ..., \Psi_n\}$ be the distribution of flows at an OF switch S_j, $\Psi^{leg}(X^{leg}) \doteq \{\Psi_1^{leg}, \Psi_2^{leg}, ..., \Psi_n^{leg}\}$ be the distribution of normal case and $\Psi^{illeg}(X^{illeg}) \doteq \{\Psi_1^{illeg}, \Psi_2^{illeg}, ..., \Psi_n^{illeg}\}$ be the distribution of attack case. As the network is stable in the normal case; therefore, $H_{leg}(IP_{src})$ is also stable. Then, $\forall i, 0 \leq i \leq N$, $\Psi_i^{leg} \ll \Psi_i^{illeg}$. More, $EX^{leg} \ll EX^{illeg}$ and $\Omega(EX^{leg}) \ll \Omega(EX^{illeg})$. Therefore, $E(\Omega(X^{leg})) \ll E(\Omega(X^{illeg}))$. Hence, we have: $-\sum_{i=1}^{N} \Psi_i^{leg} \log_2 \Psi_i^{leg} \gg -\sum_{i=1}^{N} \Psi_i^{illeg} \log_2 \Psi_i^{illeg}$. Finally, the result is $H_{leg}(IP_{src}) \gg H_{illeg}(IP_{src})$. ■

The attribute (i.e., header fields of the packet) used to aggregate incoming flows at a local OF switch S_j depends on the scenario of the attack under consideration (e.g., DNS amplification attack). For example, many attackers sends multiple illegitimate DNS requests from the same UDP port source with the same DNS request type (e.g., ANY), as legitimate machines send legitimate DNS requests from random UDP port source with different DNS request types (e.g., A, MX, NS, etc.). Consequently, we use $\{IP_{src}$, $Port_{src}$ and $ANY\}$ as attributes to aggregate DNS flows. Similarly, we can also define the UDP port source entropy, $H(Port_{src})'$ and ANY entropy, $H(ANY)'$. Finally, we represent the network traffic features at the k^{th} time period as:

$$X_k = \{H(IP_{src})'_k, H(Port_{src})'_k, H(ANY)'_k\} \quad (7)$$

3) BNF: BNF aims to detect illegitimate DNS requests that overload the network and may exhaust TCAM in OF switches. First, we describe the flow representation. Then, we discuss BNF criterion of classification.

a) Flow representation: We represent each sample in BNF by a vector $x = (x_1, x_2, x_3)$ where x_1, x_2, x_3 are values taken, respectively, by random variables $H(IP_{src})'$, $H(Port_{src})'$ and $H(ANY)'$. When the network suffers from DNS amplification attack, the number of DNS requests sent from the same IP_{src}, with the same type (e.g., ANY) and

from the same $Port_{src}$, will increase sharply. This leads to a significant decrease of entropy values of, respectively, $H(IP_{src})'$, $H(Port_{src})'$ and $H(ANY)'$. Therefore, the vector X_k can better represent the DNS amplification attack characteristics.

b) Criterion of classification: BNF is a binary classifier that consider two classes of DNS requests: (1) legitimate DNS requests, denoted by leg, and (2) illegitimate DNS requests, denoted by $illeg$. The class of X_k, denoted by c, can be either leg or $illeg$ and is defined as follows:

$$c = \underset{c \in \{leg, illeg\}}{\arg\max}\ p(c|X_k)$$

Since $p(leg|X_k) + p(illeg|X_k) = 1$; thus, the selection criterion becomes:

$$X_k \text{ is illegitimate iff } p(illeg|X_k) \geq 0.5 \tag{8}$$

According to Bayes theorem [45], the probability of vector X_k to belong to class c is defined as follows:

$$p(C = c|X = X_k) = \frac{p(C = c).p(X = X_k|C = c)}{p(X = X_k)} \tag{9}$$

Using the total probability theorem, we conclude:

$$p(C = c|X = X_k) = \frac{p(C = c).p(X = X_k|C = c)}{\sum_{c \in \{leg, illeg\}} p(C = c)p(X = X_k|C = c)} \tag{10}$$

Therefore, the selection criterion can be expressed as follows:

$X_k$is illegitimate iff:

$$p(C = c|X = X_k) = \frac{p(C = c).p(X = X_k|C = c)}{\sum_{c \in \{leg, illeg\}} p(C = c)p(X = X_k|C = c)} \geq 0.5 \tag{11}$$

$H(IP_{src})'$, $H(Port_{src})'$ and $H(ANY)'$ are conditionally independent variables given class c. Let $p_k(leg)$ and $p_k(illeg)$ denote, respectively, the conditional probabilities that vector X_k is legitimate and illegitimate.

Using Eq. (11), the selection criterion becomes, X_k is illegitimate iff:

$$\frac{\prod_{k=1}^{n} p_k^{X_k}(illeg)(1-p_k(illeg))^{1-X_k} p(illeg)}{\prod_{k=1}^{n} p_k^{X_k}(illeg)(1-p_k(illeg))^{1-X_k} p(illeg) + \prod_{k=1}^{n} p_k^{X_k}(leg)(1-p_k(leg))^{1-X_k} p(leg)} \geq 0.5 \tag{12}$$

When $p(leg) = p(illeg)$, the selection criterion becomes, $X_k$is illegitimate iff:

$$\frac{\prod_{k=1}^{n} p_k^{X_k}(illeg)(1-p_k(illeg))^{1-X_k}}{\prod_{k=1}^{n} p_k^{X_k}(illeg)(1-p_k(illeg))^{1-X_k} + \prod_{k=1}^{n} p_k^{X_k}(leg)(1-p_k(leg))^{1-X_k}} \geq 0.5 \tag{13}$$

BNF is trained and then used to classify the k^{th} vector X_k as either legitimate or illegitimate. By combining ECS and BNF, WisdomSDN can accurately detect illegitimate DNS requests in real time with low false positive rate while maintaining a high detection rate (see Section VIII).

4) Machine Learning DDoS Detection Algorithm: After having explained our machine learning DDoS detection Module, in this section we summarize this calculation via an algorithm (Algorithm 2). This algorithm is implemented on the top of the SDN controller as a REST application and allows, using sFlow protocol, the monitoring of each incoming flow; it defines some monitoring metrics (e.g., address groups, attributes of flows aggregation, and thresholds) and commands the sFlow collector to deploy these monitoring metrics within the data plane using sFlow protocol. Using the collected data, the algorithm detects illegitimate flows.

Algorithm 2 Machine Learning DDoS Detection Algorithm.

Input : Aggregated DNS requests from OF Ingress port
Output: legitimate or illegitimate

1. Define address groups, the attribute to aggregate flows denoted as att (i.e., $IP_{src}, Port_{src}, ANY$) and initialize the monitoring interval ΔT.
2. Identify flow f_i, $\forall\, 0 \leq i \leq N$, and set the count number of packets for each flow f_i to zero.
3. When the monitoring interval ΔT is over, the entropy values are calculated as follows:

for *each flow f_i at a local OF switch S_j* **do**

$$N_{F_{i,j}}(att_i, S_j, t + \Delta T) = |F_{i,j}(att_i, S_j, t + \Delta T)| - |F_{i,j}(att_i, S_j, t)|$$

end

for $i \leftarrow$*1 to N* **do**

$$p_{i,j}(att_i, S_j) = \frac{N_{F_{i,j}}(att_i, S_j, t+\Delta T)}{\sum_{i=1}^{N} N_{F_{i,j}}}$$

$$H(IP_{src}) += -p_{i,j}(IP_{src_i}, S_j) \log_2 p_{i,j}(IP_{src_i}, S_j)$$

$$H(Port_{src}) += -p_{i,j}(Port_{src_i}, S_j) \log_2 p_{i,j}(Port_{src_i}, S_j)$$

$$H(ANY) += -p_{i,j}(ANY_i, S_j) \log_2 p_{i,j}(ANY_i, S_j)$$

end

4. *Calculate the normalized entropy values as follows:*

$$H(IP_{src})' = \frac{H(IP_{src})}{\log_2 N}$$
$$H(Port_{src})' = \frac{H(Port_{src})}{\log_2 N}$$
$$H(ANY)' = \frac{H(ANY)}{\log_2 N}$$

5. *Calculate the network traffic features at the k^{th} time period as:*

$$X_k = \{H(IP_{src})'_k, H(Port_{src})'_k, H(ANY)'_k\}$$

6. **if** *p(illeg $|X_k$) ≥ 0.5* **then**
   notifies DM
   Go to step 2

else
   Go to step 2

end

VII. DNS Mitigation (DM) Scheme

When BNF detects illegitimate DNS requests (i.e., illegitimate traffic features vector X_k), a mitigation rule is executed in order to protect TCAM of OF switches. DM installs new OF rules into OF switches under DNS amplification attack;

these rules have a high priority to match illegitimate flows and monitor their speed. OF protocol was designed without any support to QoS features; however, OF 1.3 introduces *m*eters [46] to OF. DM specifies for each flow entry in OF table a *meter*; *meter* entries with different *Meter_id* are deployed to monitor the speed of the classified illegitimate traffic features vector X_k by BNF; if the packet rate surpasses the band (i.e., rate limiter); then, DM systematically drops the illegitimate DNS requests.

VIII. Evaluation of WisdomSDN

In this section, we present the evaluation of WisdomSDN. First, we introduce the experimental environment. Then, we evaluate the performance of WisdomSDN.

A. Experimental Environment

PAS is implemented in OF switch (i.e., OpenvSwitch); while Machine Learning DDoS Detection Module (i.e., FSC, ECS and BNF) and DM are implemented as applications on the top of the SDN controller. PAS systematically blocks amplified illegitimate DNS responses. Using FSC, ECS extracts network traffic features; then, BNF detects automatically illegitimate DNS requests. Finally, DM mitigates the illegitimate DNS requests. In order to emulate a real network environment, we use Mininet, a popular SDN emulation tool. Mininet uses Linux containers and virtual OF switches (e.g., OpenvSwitch) to allow realistic virtual networks of switches and hosts to be constructed using a virtual machine (VM). In our testing environment, the network monitor (i.e., sFlow-RT) and the SDN controller (i.e., Floodlight) are installed on the host-system. We run our experiments on a PC with CPU Intel Core i7-8750H-2.2 GHz and 16GB RAM.

Fig. 6 shows the experimental environment (i.e., components of the testbed); it consists of: (a) SDN controller (i.e., Floodlight); (b) sFlow collector (i.e., sFlow-RT): to perform the monitoring of network traffic features (sFlow-RT can perform monitoring of 7500 switch ports in data center networks); (c) REST applications: it executes FSC, ECS, BNF and DM; and (d) 6 OF switches, that are connected together via 1 Gbps links. Each OF network contains more than 20 hosts; multiple hosts in our setup topology are simulated to act as Open Resolvers and send amplified DNS responses (we tested several attacks setup: when the attacker is external (see Figs. 4(a) and 4(b)), and then when it is internal (see Fig. 4(c) and 4(d)) and other hosts are legitimate hosts. In order to test the scalability of WisdomSDN, we varied the rate of attack from 100 to 500 Mbps. Table 2 shows the parameter values for our experimentations.

TABLE II: Parameter values.

	Average traffic rate (Mbps)	Sampling rate	Attack rate(pkts/s)
Exp.1	100	1/64	200-500
Exp.2	500	1/256	1000-2500

For the attack script, Scapy's Python library [47] module of Python that makes easy to forge, send, receive and manipulate network packets, is used to forge DNS request. NodeJs [48] is used to create the DNS server using dnsd package; in order to emulate a real DNS amplification attack, the DNS server is implemented to send amplified DNS responses. Tcpdump [49], Wireshark [50] and Iperf [51] are used to capture and analyzes the network traffic in real-time. For each scenario presented in Figs. 4 and 5, we tested several cases of attacks and made test measurements considering both cases: (1) without wisdomSDN; and (2) with wisdomSDN.

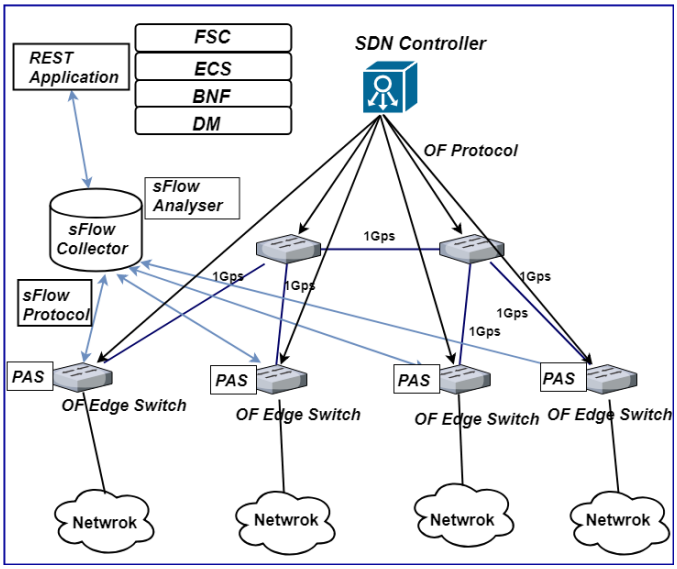
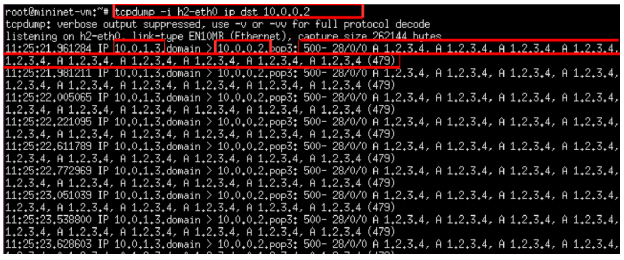
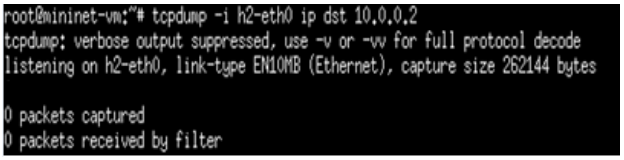


Fig. 6: Experimental Environment.

Fig. 7(a) shows that, without wisdomSDN, the victim (host h2) is flooded by amplified illegitimate DNS responses.



(a)

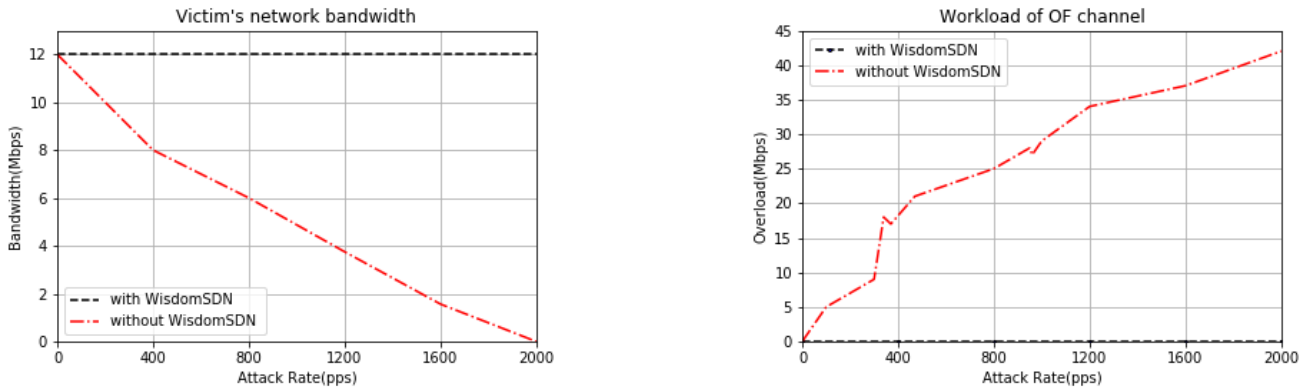


(b)

Fig. 7: Capture on victim's network without and with wisdomSDN.

After deploying WisdomSDN, we re-launched the attack. Fig. 7(b) shows that, the victim does not receive any amplified illegitimate DNS responses.

Figs. 8(a) and 8(b) show the protection of the victim's bandwidth resources and OF channel by WisdomSDN. Fig.



(a) (b)

Fig. 8: Performance evaluation of WisdomSDN in terms of: a) bandwidth consumption; b) OF channel workload.

8(a) shows the victim's bandwidth resources consumption with and without WisdomSDN. Without WisdomSDN, with the increase of attack rate, the victim's network bandwidth consumption decreases sharply (victim's network bandwidth goes to zero when the attack rate reaches 2000 packets per second (pps)). With WisdomSDN, the SDN controller set flow rules to OF switch (see Algorithm 1) to process proactively DNS packets and to: (1) systematically drop the amplified DNS responses; (2) perform one-to-one mapping between DNS requests and DNS responses; and (3) send each DNS response according to the original port, if the source IP address of DNS request has been spoofed, the attacker will receive the illegitimate DNS response. Thus, the victim is totally protected and his network bandwidth is not used by the traffic generated by the attack (12Mbps of available bandwidth) even if the attack rate reaches 2000 pps. PAS not only saves the victim's network bandwidth, but also reduces the load of OF channel. Fig. 8(b) shows OF channel's workload variations with and without WisdomSDN. Without WisdomSDN, with the increase of attack rate, the load of OF channel increases sharply. With WisdomSDN, we use a proactive mechanism (i.e., PAS) and not a reactive one. In PAS, each OF switch makes filtering of illegitimate DNS responses from all DNS traffic without sending amplified traffic to SDN controller. Thus, saving the load of OF channel.

Fig. 9 shows that, without WisdomSDN, the attack rate reaches more than 2000 DNS requests per second. Thus, even if the victim is protected using PAS, the OF switch can be overwhelmed. This occurs when the attacker is within the network of the victim and tries to overwhelm TCAM of OF switches with a large number of illegitimate DNS requests. However, when our machine Learning DDoS detection module is deployed, the traffic, generated by the attack. is effectively monitored using FSC; when ECS and BNF classify the flow as illegitimate, it is stopped and DM automatically mitigates the traffic attack (rate limit).

Fig. 10 shows that the time taken to mitigate the attacks is less than 13 seconds. Thus, we can effectively and quickly recover the network in short time.

To examine the effectiveness of ECS, we set a simulation

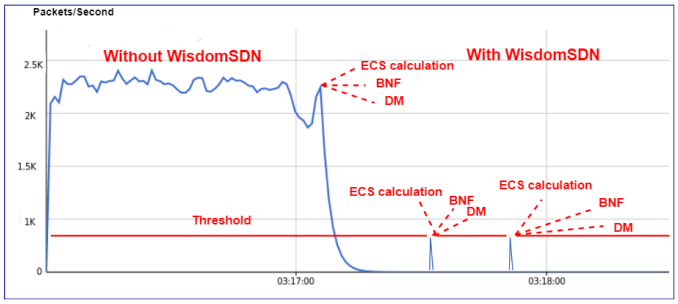


Fig. 9: DNS request's traffic before and after enabling WisdomSDN.

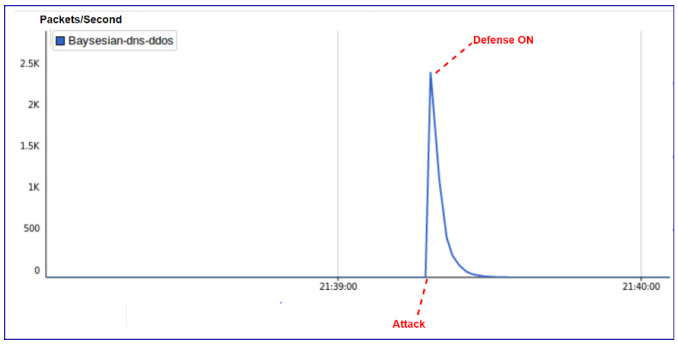
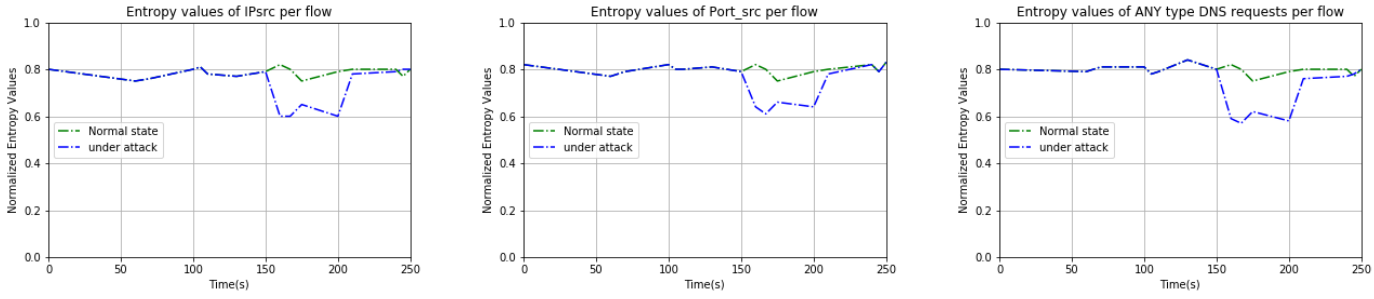


Fig. 10: Time of DM mitigation.

interval of $250s$; then, we launch the attack during the interval of $150-200s$. Figs. 11(a), 11(b) and 11(c) show, respectively, the normalized entropy values of IP source address, UDP port source and ANY. These normalized entropy values decreases rapidly in the interval of attack $150-200s$. Thus, ECS can better represents the attack and indicates that we are very likely in the presence of illegitimate flow. Therefore, BNF can detect the attack with high detection rate and low false positive rate.

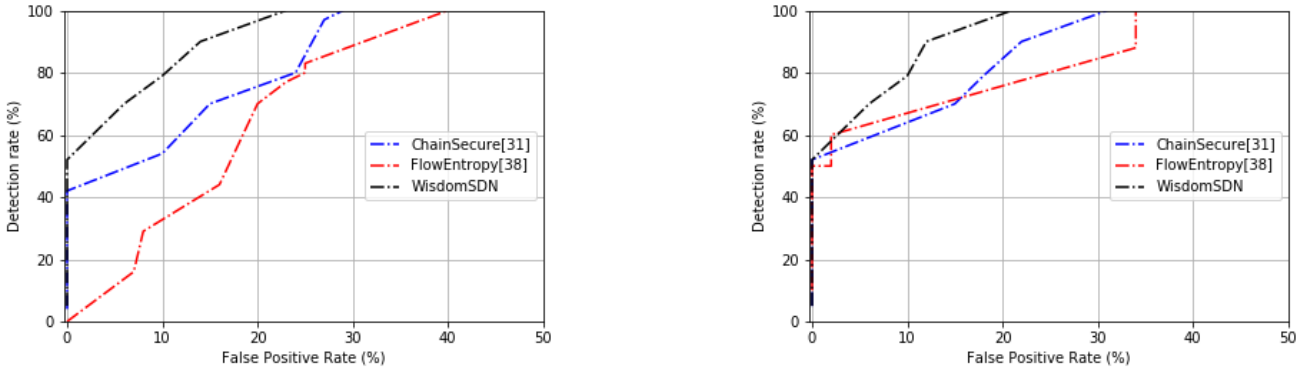
B. Performance Evaluation

The performance of BNF is measured using ROC curves. The ROC curve shows the True Positive Rate (TPR) called



(a) (b) (c)

Fig. 11: Normalized entropy values of: a) source IP address (IP_{src}); b) UDP port Source ($Port_{src}$); and c) ANY DNS requests.



(a) (b)

Fig. 12: ROC curves for the: a) 100 Mbps case; b) 500 Mbps case.

sensitivity or Detection Rate (DR) according to the False Positive Rate (FPR). To evaluate the detection rate of BNF in high traffic rate, we conducted two experiments (i.e., $100Mbps$ and $500Mbps$) and we compared BNF in terms of detection rate and FPR with ChainSecure [31] and FlowEntropy [38]. To measure the performance of BNF, we use confusion matrix (see Table 3); it provides several metrics that can help to study the performance of BNF. We also define DR and FPR as follows:

TABLE III: Confusion matrix.

	Classified as illegitimate	Classified as legitimate
illegitimate flows	TP (True Positives)	FN (False Negatives)
legitimate flows	FP (False Positives)	TN (True Negatives)

$$Se = DR = \frac{TP}{TP + FN}, 1 - Sp = FPR = \frac{FP}{TN + FP}$$

where, TP represent the illegitimate flows that are correctly classified as illegitimate, FN represent the illegitimate flows that are identified as legitimate, FP represent the legitimate flows that are classified as illegitimate, and TN represent the legitimate flows that are correctly identified as legitimate. Fig. 12(a) shows that WisdomSDN achieves around 100% detection rate for $100Mbps$ case while it has just 23% of FPR, while [31] and [38] achieve the same detection rate but with respectively 31% and 40% of FPR. Fig. 12(b) shows that WisdomSDN achieves around 100% detection rate for 500 Mbps case while it has just 21% of FPR, while [31] and [38] achieve the same detection rate but with respectively 30% and 34% of FPR.

IX. Conclusion

SDN is an emerging technology that brings numerous benefits by decoupling the control plane from data plane. On one hand, the separation of the control plane from the data plane allows for more control over the network and brings new capabilities to deal with large forms of DDoS attacks. On the other hand, this separation introduces new challenges regarding the security of the control plane. This paper aims to deal with DNS amplification attack while maintaining the SDN secure (i.e., protecting the resources of data plane (i.e., Ternary Content Addressable Memory (TCAM) of OF switches). For this aim, first, we proposed PAS, a proactive and stateful scheme that performs a one-to-one mapping between

DNS request and DNS response in order to: (1) protect the victim from DNS amplification attack; and (2) protect the resources of the SDN controller. Then, we proposed a machine learning DDoS detection module that consists of FSC, ECS and BNF in order to detect illegitimate DNS requests and protect TCAM of OF switches. Finally, DM is designed to mitigate illegitimate DNS requests. In our simulations, we set a fixed idle and hard timeouts of flow rules. For large values, flow rules stay in OF table for a long time which can exhaust TCAM of OF switches, while too small values, may lead to the dropping of legitimate DNS responses. For future work, we intend to design a novel optimization algorithms to set dynamically those timeouts. This optimization can be based on the capacity of OF table, traffic rate and workload of both data plane and control plane. Moreover, we aim also to extend this work considering inter-domain mitigation based on a decentralized architecture (e.g., Blockchain [52]) ensuring two levels of mitigation (i.e., intra-domain and inter-domain DDoS mitigation [53]).

REFERENCES

[1] C. Rossow, "Amplification hell: Revisiting network protocols for ddos abuse," in *In Proceedings of the 2014 Network and Distributed System Security Symposium, NDSS*, 2014.

[2] "Dns expertise." Accessed: June. 1, 2019. [Online]. Available: http://dns.measurement-factory.com/surveys/sum1.html

[3] S. Sharwood, "Github wobbles under ddos attack." Accessed: June. 1, 2019. [Online]. Available: https://www.theregister.co.uk/2015/08/26/github_wobbles_under_ddos_attack

[4] B. Schneier, "Lessons from the dyn ddos attack." Accessed: June. 1, 2019. [Online]. Available: https://www.schneier.com/blog/archives/2016/11/lessons_from_th_5.html

[5] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 623–654, Firstquarter 2016.

[6] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, and Y. Liu, "A survey on large-scale software defined networking (sdn) testbeds: Approaches and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 891–917, Secondquarter 2017.

[7] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 325–346, Firstquarter 2017.

[8] E. Kaljic, A. Maric, P. Njemcevic, and M. Hadzialic, "A survey on data plane flexibility and programmability in software-defined networking," *IEEE Access*, vol. 7, pp. 47 804–47 840, 2019.

[9] D. Hu, P. Hong, and Y. Chen, "Fadm: Ddos flooding attack detection and mitigation system in software-defined networking," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[10] R. Mohammadi, R. Javidan, and M. Conti, "Slicots: An sdn-based lightweight countermeasure for tcp syn flooding attacks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 487–497, June 2017.

[11] V. Varadharajan, K. Karmakar, U. Tupakula, and M. Hitchens, "A policy-based security architecture for software-defined networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 897–912, April 2019.

[12] Z. Wang, H. Hu, and G. Cheng, "Design and implementation of an sdn-enabled dns security framework," *China Communications*, vol. 16, no. 2, pp. 233–245, Feb 2019.

[13] S. Yu, W. Zhou, R. Doss, and W. Jia, "Traceback of ddos attacks using entropy variations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 412–425, March 2011.

[14] B. Rashidi, C. Fung, and E. Bertino, "A collaborative ddos defence framework using network function virtualization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2483–2497, Oct 2017.

[15] B. Rashidi, C. Fung, and M. Rahman, "A scalable and flexible ddos mitigation system using network function virtualization," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–6.

[16] A. Jakaria, B. Rashidi, M. A. Rahman, C. Fung, and W. Yang, "Dynamic ddos defense resource allocation using network function virtualization," in *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, ser. SDN-NFVSec '17. New York, NY, USA: ACM, 2017, pp. 37–42. Accessed: June. 1, 2019. [Online]. Available: http://doi.acm.org/10.1145/3040992.3041000

[17] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, and M. Conti, "A survey on the security of stateful sdn data planes," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1701–1725, thirdquarter 2017.

[18] A. Abdou, P. C. van Oorschot, and T. Wan, "Comparative analysis of control plane security of sdn and conventional networks," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3542–3559, Fourthquarter 2018.

[19] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging sdn and nfv security mechanisms for iot systems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 812–837, Firstquarter 2019.

[20] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Y. Yau, and J. Wu, "Realtime ddos defense using cots sdn switches via adaptive correlation analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1838–1853, July 2018.

[21] L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, "Tennison: A distributed sdn framework for scalable network security," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2805–2818, Dec 2018.

[22] J. Steadman and S. Scott-Hayward, "Dnsxd: Detecting data exfiltration over dns," in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2018, pp. 1–6.

[23] S. Deng, X. Gao, Z. Lu, and X. Gao, "Packet injection attack and its defense in software-defined networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 695–705, March 2018.

[24] "Openvswicth." Accessed: June. 1, 2019. [Online]. Available: https://www.openvswitch.org/

[25] "Mininet." Accessed: June. 1, 2019. [Online]. Available: http://mininet.org

[26] D. Huistra, "Detecting reflection attacks in dns flows." Accessed: June. 1, 2019. [Online]. Available: https://pdfs.semanticscholar.org/4ad8/24537f212f70e25e4cbab55498f5a8e43942.pdf

[27] T. Rozekrans, M. Mekking, and J. de Koning, "Defending against dns reflection amplification attacks," Feb 2013.

[28] C. Sun, B. Liu, and L. Shi, "Efficient and low-cost hardware defense against dns amplification attacks," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, Nov 2008, pp. 1–5.

[29] F. Guo, J. Chen, and T.-c. Chiueh, "Spoof detection for preventing dos attacks against dns servers," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, ser. ICDCS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 37–. Accessed: June. 1, 2019. [Online]. Available: https://doi.org/10.1109/ICDCS.2006.78

[30] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, "A fair solution to dns amplification attacks," in *Proceedings of the Second International Workshop on Digital Forensics and Incident Analysis*, ser. WDFIA '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 38–47. Accessed: June. 1, 2019. [Online]. Available: https://doi.org/10.1109/WDFIA.2007.2

[31] Z. A. El Houda, L. Khoukhi, and A. Hafid, "Chainsecure - a scalable and proactive solution for protecting blockchain applications using SDN," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 1–6.

[32] Z. A. E. Houda, A. Hafid, and L. Khoukhi, "Brainchain - a machine learning approach for protecting blockchain applications using sdn," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[33] R. Braga, E. Mota, and A. Passito, "Lightweight ddos flooding attack detection using nox/openflow," in *IEEE Local Computer Network Conference*, Oct 2010, pp. 408–415.

[34] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Recent Advances in Intrusion Detection*, R. Sommer, D. Balzarotti, and G. Maier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 161–180.

[35] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed ddos detection mechanism in software-defined networking," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, Aug 2015, pp. 310–317.

[36] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A sdn-oriented ddos blocking scheme for botnet-based attacks," in *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2014, pp. 63–68.
[37] A. Zaalouk, R. Khondoker, R. Marx, and K. Bayarou, "Orchsec: An orchestrator-based architecture for enhancing network-security using network monitoring and sdn control functions," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–9.
[38] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," *Comput. Netw.*, vol. 62, pp. 122–136, Apr. 2014. Accessed: June. 1, 2019. [Online]. Available: http://dx.doi.org/10.1016/j.bjp.2013.10.014
[39] J. B. Hong and D. S. Kim, "Assessing the effectiveness of moving target defenses using security models," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 163–177, March 2016.
[40] "sflow-rt." Accessed: June. 1, 2019. [Online]. Available: http://www.sflow-rt.com
[41] "Rest api." Accessed: June. 1, 2019. [Online]. Available: http://www.sflowrt.com/reference.php
[42] "Opendaylight." Accessed: June. 1, 2019. [Online]. Available: https://www.opendaylight.org/
[43] "Floodlight." Accessed: June. 1, 2019. [Online]. Available: http://www.projectfloodlight.org/
[44] S. C. E, "Prediction and entropy of printed english." Bell system technical journal, pp. 50–64.
[45] P. D. Hoff, "A first course in bayesian statistical methods," *Springer*, 2009.
[46] "Openflow switch specification." Accessed: June. 1, 2019. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf
[47] "Scapy." Accessed: June. 1, 2019. [Online]. Available: http://www.secdev.org/projects/scapy
[48] "Nodejs." Accessed: June. 1, 2019. [Online]. Available: https://nodejs.org/en/
[49] "Tcpdump." Accessed: June. 1, 2019. [Online]. Available: https://www.tcpdump.org/
[50] "Wireshark." Accessed: June. 1, 2019. [Online]. Available: https://www.wireshark.org/
[51] "Iperf." Accessed: June. 1, 2019. [Online]. Available: https://iperf.fr/
[52] Z. Abou El Houda, A. S. Hafid, and L. Khoukhi, "Cochain-sc: An intra- and inter-domain ddos mitigation scheme based on blockchain using sdn and smart contract," *IEEE Access*, vol. 7, pp. 98 893–98 907, 2019.
[53] Z. A. El Houda, A. Hafid, and L. Khoukhi, "Co-iot: A collaborative ddos mitigation scheme in iot environment based on blockchain using sdn," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.