IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 25, 2021, accepted August 3, 2021, date of publication August 9, 2021, date of current version August 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3103845

Towards Crossfire Distributed Denial of Service Attack Protection Using Intent-Based Moving Target Defense Over Software-Defined Networking

MUHAMMAD FARAZ HYDER [1] **AND TASBIHA FATIMA** [2]

[1] Department of Software Engineering and Technology, NED University of Engineering and Technology, Karachi 75270, Pakistan
[2] Department of Computer Science and Information Technology, NED University of Engineering and Technology, Karachi 75270, Pakistan

Corresponding author: Muhammad Faraz Hyder (farazh@neduet.edu.pk)

This work was supported by the Department of Computer Science and Software Engineering, NED University of Engineering and Technology, Karachi, Pakistan.

ABSTRACT Crossfire is an indirect target area link-flooding Distributed Denial of Service (DDoS) attack determined to affect the neighbors of the real target. Currently, Crossfire DDoS attacks are acquiring impetus because of their indistinguishability and undetectability. SDN (Software Defined Networking) is a progressing technique because of its adaptability and programmability. Moving Target Defense (MTD) is an arising security strategy to counter attacks by progressively changing the attacked plane. IBN (Intent-based Networking) is another promising methodology for providing dynamic network management. IBN-based MTD can provide efficient MTD solutions because of the concentrated control and observing capacities of the intents when translated into rules inside the SDN control plane. In this paper, a framework for the security of Crossfire DDoS attacks is proposed by making use of Intent-based Traffic modifications through the Open Networking Operating System (ONOS) Rest API and Domain Name System (DNS) port redirection. In this paper, we exploited Intent-based MTD to divert traffic from the principal host to virtual shadow hosts to counter this attack. Traffic redirection helps in masquerading the attacker headed for shadow host and consequently getting the erroneous path towards the network and, hence, the Crossfire attack couldn't be executed as expected. The proposed technique is simulated using Mininet and ONOS SDN controllers. The outcomes showed traffic is successfully redirected at a low computational expense. Therefore, Crossfire DDoS is efficiently mitigated as promising results are found.

INDEX TERMS Crossfire DDoS, network function virtualization, intent-based networking, moving target defense, software defined networking.

I. INTRODUCTION

Indirect Distributed Denial of Service (DDoS) attack carried out by utilizing bot-net driven computers has been known for some time. This novel class of DDoS attack is known as Crossfire attack or Link-Flooding attack [1]. This type of attack can remain undetected if any of the following conditions are fulfilled [2]:

1) Valid IP addresses might be used by bots therefore, all the detection or prevention mechanisms based on spoofed IPs turned out to be inappropriate.
2) Bots may achieve link-flooding attack on a target using legitimate traffic
3) Botnets can also slowly increase the intensity of attack so that it remains undetected by anomaly-based IDSs

Addressing such type of attacks inside networks pose numerous technological, monetary and legal challenges, especially, when there is need for new hardware installation which will eventually increase the capital expenses or modification of existing infrastructure may add up operational cost along with the increase in system complexity [3]. Therefore, one of the most efficient techniques to escalate cost for the attackers is MTD (Moving Target Defense) [4]–[7].

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

As in computer networks such as Cloud systems, Cloud Datacenters, IoT (Internet of Things), and their resources, the DDoS attack rate is extremely high nowadays. Among them, the Crossfire DDoS attack is a novel Indirect Target Link-Flooding DDoS attack which is increasing due to its various characteristics including indistinguishability and undetectability. Due to the distinct characteristics of Crossfire DDoS Attacks, we handled it with MTD. There are SDN-based existing solutions and a few NFV based one. Therefore, proposing protection for this type of attack is highly needed. We integrated SDN, NFV(Network Function Virtualization), and IBN(Intent-based Networking) as an innovative security mechanism. Our framework is a combination of IBN and MTD that yields two methodologies for safeguarding traffic flowing through nodes and the DNS port of nodes inside the network. Since IBN is an emerging paradigm and it provides networking professionals, capability to construct innovative and flexible security solutions. When incorporated with MTD, it enhances the potential to secure network infrastructure after the implementation of our proposed framework.

A. RESEARCH QUESTIONS

The focused areas in our research work are as per following:

1) How our proposed defense framework could be intended for safeguarding Crossfire DDoS attacks?
2) How to optimize performance while mitigating Crossfire DDoS attacks?
3) How can Intent-based MTD be fully utilized for fabricating Crossfire DDoS attack solution using IBN Modifications?

B. CONTRIBUTIONS

The main contributions of this framework are:

1) Implementation of a redirection method as a Crossfire DDoS protection technique using Intent-based Modifications using ONOS SDN controller in a distributed environment.
2) Execution of Intent-based Modifications through ONOS RestAPI utilizing ONOS cluster for Crossfire DDoS protection to reroute the packets by exploiting SDN, NFV, and Intent-based Networking.
3) Accomplishment of Intent-based Modifications through DNS port redirection utilizing ONOS application.
4) ONOS 3-node cluster is deployed as a virtual machine along with Mininet for creating a test environment.

This paper is organized as follows. Section II covers the Background of technologies utilized to achieve this mechanism. Related Work is defined in Section III. The proposed scheme is presented in Section IV. Intent-based Modifications through RestAPI and its Baseline Performance metrics are described in sections V and VI respectively. Similarly, Intent-based Modifications through DNS port redirection and its Baseline Performance metrics are described in sections VII and VIII respectively. However, the Experimental Results obtained are discussed in Section IX. While Section X concludes the paper and determines Future Recommendations.

II. BACKGROUND

Crossfire DDoS attacks have been receiving attention due to the congestion caused by them in the network by eventually intensifying the bot traffic and evenly distributing on the malicious systems to make it difficult for the defender to identify it. Therefore, we need a framework for such attacks to recognize them using a proactive approach. There are various methods to solve the above-defined problem other than SDN and NFV while the proposed framework exploits technologies such as SDN and NFV for managing traffic flows inside the network.

A. SDN (SOFTWARE DEFINED NETWORKING)

Three planes primarily constitute the computer networks: Control Plane, Data plane, and Management Plane. Networking devices i.e. switches and routers that have a responsibility to forward the traffic are organized inside the data plane, also known as the forwarding plane. Protocols that are residing at the control plane, are chiefly utilized for installing flow rules inside forwarding tables of devices at the data plane. Security and networking policies comprise the management plane for administering the network. These policies are enforced inside the data plane through the controller plane. The rigid and complexity caused by the traditional networks due to closely integrated data and control planes inside the networking device, has given rise to the development of Software Defined Networking (SDN) [8]. Since the implementation and deployment of novel networking applications have turned out to be a tiresome and complicated task for the networking engineers owing to the powerful integration of control and data planes within conventional networks [9]. Certain hardware up-gradation is required in each network device for carrying out modifications inside the control plane. Consequently, for deploying different network and security characteristics into the network by exploiting middleboxes for instance load balance, IDS (Intrusion Detection Systems), firewalls, etc. Usually, these middleboxes are employed at specifically fixed strategic sites, which poses difficulty for dynamic reconfiguration of the network.

B. NFV (NETWORK FUNCTION VIRTUALIZATION)

The transformation in computer networks brought by NFV aids users to hand over networking functions from vendor-specific and proprietary hardware applications to software based on Commercial-off-the-Shelf (COTS) platforms. References [10] The provision of network services in virtual machines (VMs) employed in Cloud setups where each virtual machine executes diverse networking functions such as IDS, load balancing, firewall, Deep Packet Inspection, etc. [11] Deploying network facilities as virtualized functions offer certain benefits as follows [10]:

- In the networking functions, distribution is flexible in general-purpose hardware
- Rapid execution and utilization of innovative network services
- Different versions of multi-tenancy and service are supported
- CAPEX prices reduced by efficiently handling energy used
- Operational practices are automated, hence, enhanced productivity and reduction in OPEX

C. IBN (INTENT-BASED NETWORKING)

Intent-based Networking is an exceptional strategy in the networking domain as it contained specialized software which facilitates enterprise models' design and schedule. Moreover, it allows consequently to declare modifications within the network throughout the processes with no interference in the running system. This approach eradicates the need for prerequisite skills for Virtual Network Functions (VNF) and Network Service (NS) orchestration through inputting intentions of users [12]. IBN or IDN (Intent-Driven Networks) biggest feature is their capability to transform conventional business requisites into network configuration plans automatically. Its chief employment that is currently acknowledged usage technique is the combination of AI (Artificial Intelligence) and SDN where AI along with Machine Learning are utilized for analyzing data collected, capturing intentions, and converting them into policies. Nevertheless, smart software like SDN controllers, resolves how intentions are translated into configurations for a specified structure, enabling the network to operate in the preferred way [13].

III. RELATED WORK

Owing to the significance of network reconnaissance concerning effectively performing DDoS attack, various MTD techniques that endeavor to solidify the challenge has been recommended lately. The primary goal is to mystify attempts of the aggressor for collecting phases of an aggressor's conceivable targeted links by robustly altering properties of the network, for instance, network paths, IP addresses, and TCP/UDP port numbers [6] as a Crossfire DDoS shielding technique.

A. SDN-BASED CROSSFIRE DDoS PROTECTION TECHNIQUES

Rafique *et al.* [14] offered a plausible solution for Crossfire DDoS attack security utilizing SDN in IoT-Edge by executing a scenario. Floodlight SDN controller and Mininet have been deploying to create the CFADefense prototyping solution. The model is constructed utilizing the SDN controller's application layer as a result, to generate the least overhead execution cost for the network.

An RDS by Achleitner *et al.* [15] suggested recognizing filtered malignant traffic sources by analyzing the SDN flow rules statistics by simulating virtual topologies. The distinctive element of this structure is the utilization of physical networking characteristics that causes the aggressor to derive target links. Honeypots had been utilized as decoy servers and trap in virtual networks for perceiving illegitimate scanning traffic and distinguish bot hosts.

Lim *et al.* [16] proposed a plan for hindering DDoS attacks that utilized botnets operating with ostensible support in SDN. It had implemented by exploiting an SDN application running on a POX SDN controller. Mininet is a testing emulator which shows that the botnet has proficiently obstructed DDoS attacks. It has a prerequisite to have continuous correspondence between the application (DDoS Blocking) and the server that required protection.

Ma *et al.* [17] presented a security method for Blind DDoS based on MTD utilizing numerous controllers. Their fundamental concept is to employ a system based on the MTD mechanism for the packets' response time, which would also be changing progressively even during examining as various employed controllers expand dynamically. The testing arrangement had carried out by implementing OpenVSwitch for switching and Floodlight as SDN controller. Apache Tomcat as web service installed on Windows Server including IXIA equipment for attack data flow and background flow generation with MTD management operating over controllers.

Belabed *et al.* [18] established a heuristic methodology for identifying botnets that performed the attack. They executed the mitigation strategy at the detecting stage. Using SDN because of adaptability and incorporated design, they tend to address Link-flooding attacks (i.e., Crossfire & Coremelt) security through intelligent forwarding to accomplish soundness inside the network. They have exploited Traffic Engineering (TE) in ISPs for the shielding arrangement.

B. SDN & NFV BASED CROSSFIRE DDoS DEFENSE MECHANISMS

Ayedgar *et al.* [19] proposed an emulated ground for performing reconnaissance attacks such as Crossfire DDoS and perceive persistent links. For agility in ISP networks, MTD and Network forensics strategies are used in the proposed framework exploiting SDN and NFV. This technique solidifies the target regions of attackers by increasing hop counts of traceroute and storage complexity. Additionally, the virtual collection point (VCP) advances authorized traffic to the Floodlight SDN controller and acts as a filter for traffic other than the legitimate one before landing at the SDN controller joining with the provenance of network resources when needed.

Zheng *et al.* [20] determined RADAR (Reinforcing Anti-DDoS Actions in Realtime) as an architecture for DDoS attacks (e.g., Crossfire) protection utilizing an approach known as adaptive correlation analysis upon COTS SDN switches. It can catch and choke 13 diverse assimilated DDoS attacks happening concurrently with neither any changes in SDN protocols and switches nor an additional application for attack detection recognition while coordinating both the innovations, i.e., SDN and NFV. Investigation to assess the environment of testbeds through the successful exhibition of RADAR for distinguishing various types of attacks having deferrals of short intervals.

C. SDN AND IBN NETWORK SECURITY SOLUTIONS

Beshley *et al.* [21] discovered a mechanism for Cloud-based technologies inside network infrastructures using SDN, NFV, and IBN by focusing and evaluating two schemes for the construction and organization of reserved networking resources. The balancing formula of QoS is examined for networks such as conventional and hybrid both. The benefits obtained for providing desired QoS level are as follows:

- Paying for only additional equipment's operational time
- Added resources are instantly deployed and collapsed
- Human interference is minimized in equipment setup
- Occurrence of automatic installation and breakdown in networks

Comer and Rastegatnia *et al.* [22] presented a framework known as Open Software-Defined Framework (OSDF) which stipulates an array of high-level operating network facilities which may be evoked using managerial applications for configuration and scrutinizing network, allowing the administrator to input policies in the system without worrying about low-level definitions. Their main contribution is the provision of abstraction through high-level rather than former programming languages of the network. It enables the manager to state requirements for applications exclusive of layer 2 and 3 identifiers. It also contains a module for the management of policy conflict which analyzes a set of policies and identifies conflicts to resolve them through the algorithm of conflict resolution by suggesting high-level abstractions. It might be used for SDN applications implementation such as firewall, intra-domain rate limiting.

Szyrkowiec *et al.* [23] defined configuration of security, as the selection of encryption is dependent on the intentional requisites. They showed an extended version of their service named automatic intent-based multilayer secure service creation. Multilayer secure services are created by the ACINO orchestrator. They are validated depending on abstract prerequisites signified through the application, it can choose parameters of service in various layers and carry out automated delivery of multiple layers secured services. The metrics for performance specify that the surplus processing time is insignificant while, the configuration and removal times differ for various technologies and therefore, must be measured during the fulfillment of application intents. As it is a genuine approach, it might be applicable for forthcoming advancements, for instance, quantum-secure encryption.

TABLE 1 represents the comparison of our proposed scheme with the existing solutions. Principal attributes are mainly focused to provide a straightforward understanding of the contrast among different technology-enabled solutions.

D. INTENT BY DEFINITION

Two terms are strongly linked to the term Intents: **Policy** and **Configuration**. Through intents, maximum abstraction can be achieved due to their capability to express easily, regardless of being achievable or not. A rule that illustrates an achievable intent in a definite manner is known as Policy while Configuration is the extraction of vital information from policy about the environment in the existing network and generates a specified layout of information about update at the device of physical or virtual network [24]. There can be two types of intents i.e. imperative or declarative. Thus, languages of the northbound interface are also divided into declarative and imperative languages (incorporating languages such as interactive, functional, logical, etc.) [25].

E. INTENT-BASED NETWORKING

Even though the abstraction through match-action increased flexibility but the set of flow rules configuration inside each device in a network to get the required global network policy signifies an error-prone task [26]. For this, a corresponding epitome evolved which is termed Intent-based Networking. By using it, a high-level policy can be specified by application developers without bothering about the way desired functionality would be achieved in the network. Intents are submitted to the SDN controllers through NBI (North-Bound Interface) and then, translate, by using a method for compiling into low-level flow rules to attain the intended targets. Programming via intent-based networking delivers benefits such as [27]:

- Abstraction of network complexity.
- Handling modifications inside the network by the controller to achieve high-level policy. For instance, automatic recompilation of intents and network reconfiguration for restoring the required connectivity with no interference from user or application provided the occurrence of network failures.

Characteristics of IBN: The main aspects of IBNS (Intent-based Networking System) are [28]:

- Translate and validate -Commands or business intents can be translated into actions to be done by the software along with authentication of successful intent execution.
- Automatic Implementation – After the definition of intent, this system will do resource allocation and policy enforcement to achieve the objective.
- Awareness of Network State – Data is continuously gathered and monitored to exhibit the current state.
- Remediation and Assurance – ML(Machine Learning) is employed by the system for implementing and maintaining the network's desirable state while corrective action is performed automatically if required.

F. ONOS INTENT FRAMEWORK

ONOS Intent Framework is provided by ONOS through a mechanism called IBN (Intent-based Networking). It allows forwarding as a result of standardized and extendable flow rules and utilizes intents for proactive connection between source and destination nodes. The path having minimal hop count is calculated between nodes and forwarding rules are deployed proactively [29].

TABLE 1. Comparison of proposed scheme with existing solutions.

Systems	SDN Enabled	IBN Enabled	NFV Enabled	MTD	Controller	Control Plane (Single / Distributed)	Crossfire Protection
1. M. F. Hyder et al. (Proposed Scheme)	✓	✓	✓	✓	ONOS	Distributed	✓
2. Rafique et al. [14]	✓	×	×	×	Floodlight	Single	✓
3. S. Achleitner et al. [15]	✓	×	×	×	POX	Single	×
4. S. Lim et al. [16]	✓	×	×	×	POX	Single	×
5. Ma D. et al. [17]	✓	×	×	✓	Floodlight	Single	×
6. Belabed et al. [18]	✓	×	×	×	SDN Controller	Single	✓
7. Ayedgar et al. [19]	✓	×	✓	✓	Floodlight	Single	✓
8. Zheng et al. [20]	✓	×	✓	×	Floodlight	Single	✓
9. Beshley et al. [21]	✓	✓	✓	×	SDN Controllers and Switches	Single	×
10. Douglas et al. [22]	✓	✓	×	×	NOS (Network Operating System)	Single	×
11. Szyrkowiec et al. [23]	✓	✓	×	×	ONOS	Single	×

G. DNS

The most vital element of the Web, the Domain Name System (DNS), yields not only maps human-understandable domain names matched with machine-readable IP addresses, yet in addition essential trust anchors for reaching out to Internet providers. Numerous internet applications, such as HTTP, email, and FTP, indirectly or directly rely upon the security framework implemented by DNS to determine a specific domain name to its IP address correspondent to building up interconnections. Unfortunately, security was not one of the represented design aspects for DNS, and it has consistently been an appealing objective to attackers [30]–[32].

Wang *et al.* [33] proposed an MDNS, a unique heterogeneous excess engineering dependent on programming characterized organizing. To upgrade the power of DNS, Multi-DNS is utilized to choose the legitimate answers and recognize the cache poisoning assaults as per their responses. Furthermore, to expand the measure of unconventionality, the possibility of dynamicity is embraced in MTD and acknowledged by progressively controlling traffic of the network with the assistance of SDN. Moreover, to additional increment, the trouble for attackers conceding all the Multi-DNS, heterogeneous DNS is utilized to stay away from similar weaknesses. Finally, they carried out MDNS and test its presentation, and the trial results demonstrate the interruption/adaptation to non-critical failure of MDNS. The average delay can be controlled by 0.3s when they used MDNS-5-s.

IV. PROPOSED SCHEME

A. THREAT MODEL

As the adversary investigative procedure to decide a long-lasting connection, we ponder that the aggressor primarily separates the most utilized link connections in the network as per the obtained results of traceroute and picks principal ‘n’ connections from the acquired assorted listing. An aggressor utilizes the fundamental rules to choose the long-lasting connection inside the networking infrastructure, which depends upon the Crossfire DDoS characteristics, accordingly:

1) The connections ought not to be contiguous with each other since they are assumed to assault the encompassing region and can’t be directly associated with one another.
2) They are not, additionally, highlighting a similar explicit target server straightforwardly.

For the defending setup, we used boundary esteems for the maximum allowable attack interval that lies in the span of 30 to 120 seconds and configured the default path on every single SDN switch for forwarding packets towards the SDN controller. Our research work has used IBN as intents are used for providing greater flexibility inside the networks and DNS port modification through intents. Both the approaches have not been used for Crossfire protection up till now.

B. EXPERIMENTAL SETUP

The implementation of the proposed MTD (Moving Target Defense framework) is made by using Mininet [34] emulator while, ONOS [35] is employed as an SDN controller. A 3-node cluster of ONOS SDN controllers is used. While the OpenFlow switch is connected to the hosts. In this paper, we will discuss two different methodologies for handling Crossfire. One is Intent-based modification through RestAPI while, the other is through DNS port redirection. To the best of our knowledge, IBN is not utilized for creating a Crossfire DDoS protection framework.

C. MININET

It is a network emulator that can simulate devices such as switches, controllers, hosts in a virtual environment. Mininet hosts operate using standard Linux networking software. Moreover, its switches support OpenFlow aimed to provide flexibility through SDN and custom routing.

D. OPEN NETWORK OPERATING SYSTEM (ONOS)

ONOS is an open-source distributed SDN controller. It is a networking tool that was developed and supported by way of a project named ONOS. It might work as a distributed system through multiple servers permitting it to utilize CPU and memory resources of servers along with offering fault tolerance in case of failure of a server and hypothetically sustaining hardware and software rolling upgrades with no network traffic interruption. Its goal is to generate Network OS (Operating System) that is too open-source to enable service providers to model real SDNs (Software Defined Networks). It is capable of providing a control plane to SDN, management of networking components, for example, routers, switches, links, etc., and executing software or modules for offering communication services to end-hosts and neighboring networks.

Figure 1 shows the experimental topology employed to construct the Crossfire Protection mechanism.

V. INTENT-BASED MODIFICATION THROUGH RestAPI

A. ATTACKER SETUP

The attacker might install a script for running the traceroute command repeatedly on each bot and saving the output in a text document. The output obtained through the total number of bot machines is evaluated by the attacker to distinguish the potential persistent linkages for bot-server pairs across the network. Although, the attacker can manually examine those files that would eventually become tiresome for him. Hence, the attacker would build another script for executing analysis that will be based on some criteria.

B. DEFENDER SETUP

We have deployed intents in ONOS RestAPI which will redirect destination addresses for setting parameters designed for route mutation of packets received on SDN switch. The phenomenon would use Intent Framework to provide NFV and IBN functionality. **Figure 2** represents the process followed for the modification of traffic. Moreover, **Algorithm 1** describes its pseudocode for a better understanding of the proposed scheme.

VI. BASELINE AND PERFORMANCE METRICS

A baseline is considered in which the route mutation strategy would be forced by the SDN controller whenever it finds any packet to be illegitimate. For performance measurement, costs for Attackers and Defenders have been utilized as metrics for the proposed system. The main point to be observed is that installed intents would affect metrics for defender cost rather than attacker cost metrics.

A. ATTACKER COST

During reconnaissance, the attacker's cost would be increased due to the injected ONOS intents over the SDN controller. Therefore, for the effectiveness of this technique, the performance metric would be:

Success Rate of Attacker: It is a proportion of total common links count appropriately distinguished by an attacker to the definite common links count within the network.

B. DEFENDER COST

Additional costs incurred for the defender while exploiting the framework would be:

1) FLOW INJECTION TIME

It is the total time the flow rule would take to be injected in the SDN switch after getting translated from intent.

2) HOP COUNT

The average rise in the number of hops in paths (in percentage) produced due to the route redirection that could be taking the extended route towards the destination, hence, utilizing the system's additional resources.

3) INCREASE IN END TO END TRANSMISSION DELAY

Legitimate packets might suffer this delay because of the proposed framework.

VII. INTENT-BASED MODIFICATION THROUGH ONOS APPLICATION USING DNS PORT REDIRECTION

A. ATTACKER SETUP

The attacker might install a program for running computer worms repeatedly on each bot and saving the output in a text document. The output obtained through the total number of bot machines is evaluated by the attacker to distinguish the potential vulnerabilities inside bot machines across the network. He would try to analyze those files for identifying the persistent links to reach the target area to carry out the attack.

B. DEFENDER SETUP

The intent will be deployed through the ONOS application which will redirect UDP destination port addresses to achieve route mutation strategy after setting parameters of UDP packets received on SDN switch. The application would use Intent

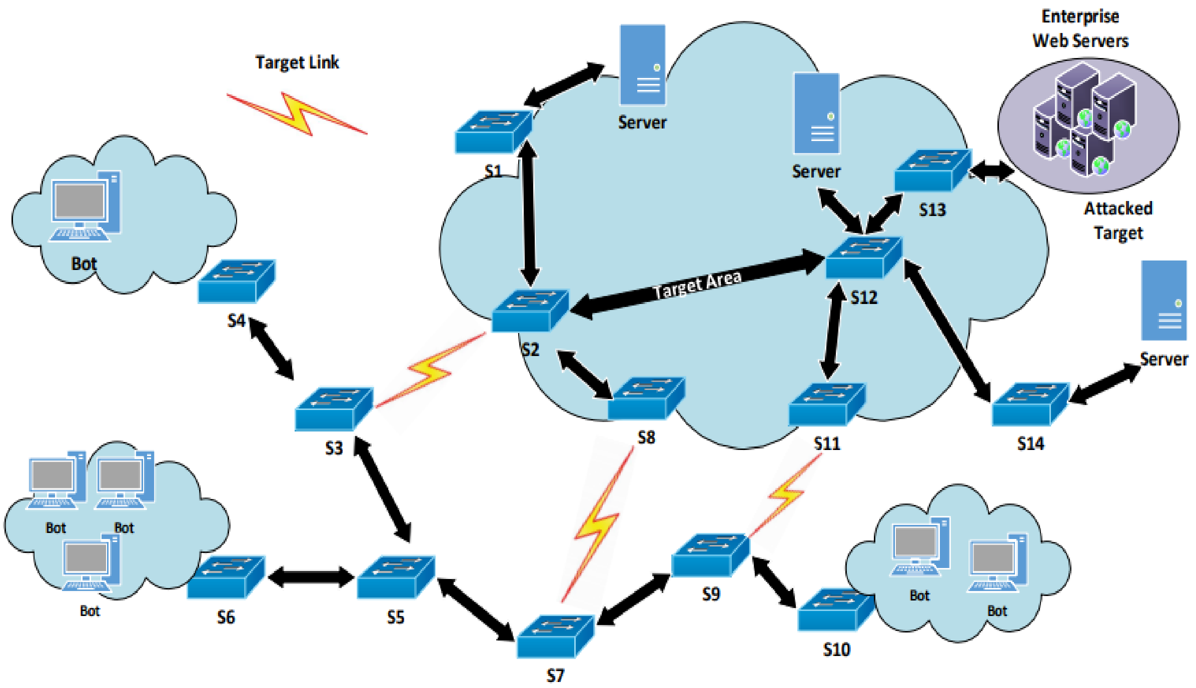


FIGURE 1. Experimental network topology.

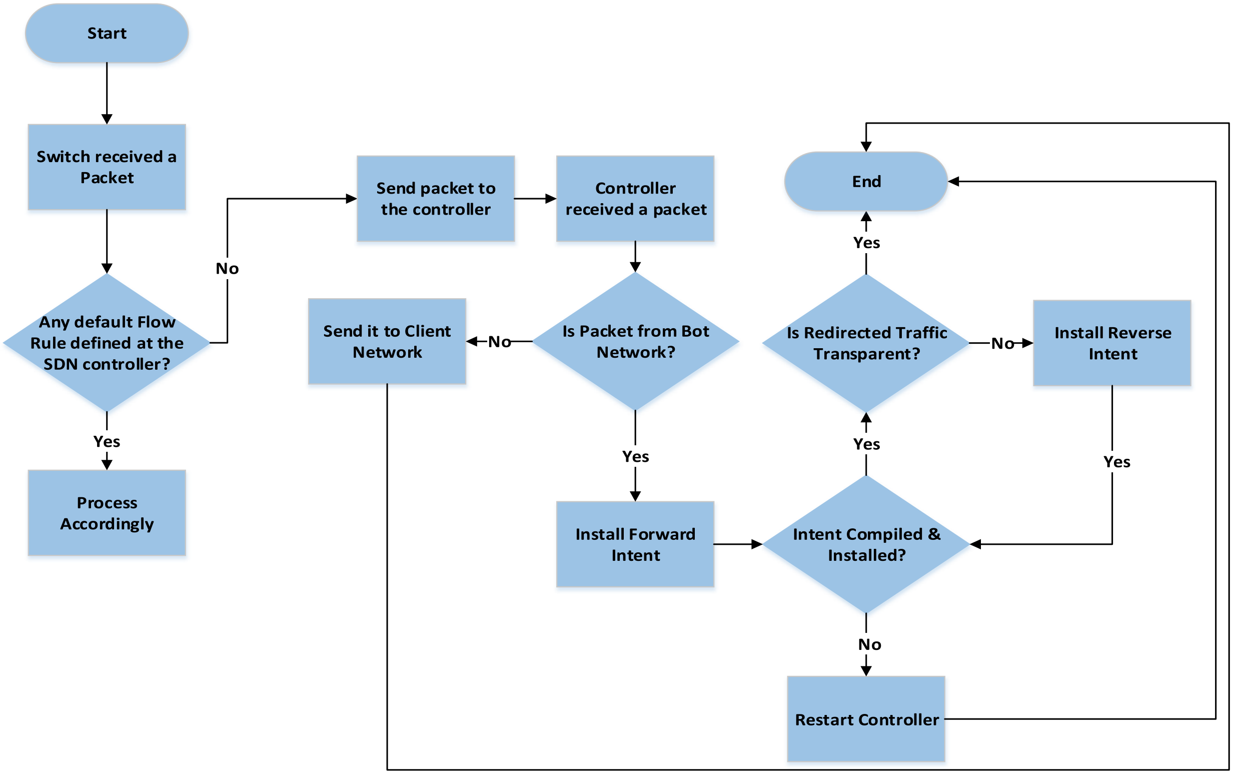

Start
Switch received a Packet
Any default Flow Rule defined at the SDN controller?
Yes
Process Accordingly
No
Send packet to the controller
Controller received a packet
Is Packet from Bot Network?
No
Send it to Client Network
Yes
Install Forward Intent
Intent Compiled & Installed?
No
Restart Controller
Yes
Is Redirected Traffic Transparent?
Yes
End
No
Install Reverse Intent
Yes

FIGURE 2. Process for intent-based modification.

Framework to provide NFV and IBN functionality. **Figure 3** represents the methodology for the port modification of DNS traffic. Moreover, **Algorithm 2** describes its pseudocode for better interpretation of the defense technique.

Algorithm 1 Intent-Based Modification for Redirection of Traffic

```
$Pkt :=\leftarrow Packet$;
$DefFlow :=\leftarrow DefaultFlowRule$;
$SdnCtrl :=\leftarrow SDNController$;
$BNet :=\leftarrow BotNetwork$;
$ForInt :=\leftarrow ForwardIntent$;
$RevInt :=\leftarrow ReverseIntent$;
$CNet :=\leftarrow ClientNetwork$;
$Switch \leftarrow ReceivesAPkt$
if Any $DefFlow$ defined at $SdnCtrl$ then
    Process according to $DefFlow$
else
    $SendPkt \rightarrow SdnCtrl$
end if
$SdnCtrl \leftarrow ReceivesAPkt$
if $Pkt \leftarrow BNet$ then
    Install $ForInt \rightarrow SdnCtrl$
else
    Send $Pkt \leftarrow CNet$
end if
$RedTrf :=\leftarrow RedirectedTraffic$
if Intent $\rightarrow$ Compiled & Installed successfully then
    if isTransparent($RedTrf$) then
        Traffic is Redirected Successfully
    else
        Install $RevInt \rightarrow SdnCtrl$
    end if
else
    $SdnCtrl \rightarrow Restart$
end if
$End$
```

VIII. BASELINE AND PERFORMANCE METRICS

A baseline is considered in which the route mutation strategy would be forced by the SDN controller whenever it finds any packet to be illegitimate. For performance measurement, costs for Attackers and Defenders have been utilized as metrics for the proposed system. The main point to be observed is that installed intent would affect metrics for defender cost rather than attacker cost metrics.

A. ATTACKER COST

During reconnaissance, the attacker's cost would be increased due to the proposed ONOS application. Therefore, for the effectiveness of an application, the performance metric would be:

Success Rate of Attacker: It is a proportion of total common links count appropriately distinguished by an attacker to the definite common links count within the network.

B. DEFENDER COST

Additional costs incurred for the defender while exploiting the framework would be:

Algorithm 2 Intent-Based Modification for Redirection of Traffic Through DNS Port

```
1:  $DNSInt :=\leftarrow DNSIntent$ Installed Through Application;
2:  $Pkt :=\leftarrow Packet$;
3:  $DefFlow :=\leftarrow DefaultFlowRule$;
4:  $SdnCtrl :=\leftarrow SDNController$;
5:  $BNet :=\leftarrow BotNetwork$;
6:  $CNet :=\leftarrow ClientNetwork$;
7:  if $Intent$ $\rightarrow$ Compiled & Installed successfully then
8:      $Switch \leftarrow ReceivesAPkt$
9:  else
10:     $SdnCtrl \rightarrow Restart$
11: end if
12: if Any $DefFlow$ defined at $SdnCtrl$ then
13:     Process according to $DefFlow$
14: else
15:     $SendPkt \rightarrow SdnCtrl$
16: end if
17: $SdnCtrl \leftarrow ReceivesAPkt$
18: if $Pkt \rightarrow UDP$ then
19:     if $Pkt \leftarrow BNet$ then
20:         $RedTrf :=\leftarrow RedirectedTraffic$
21:         $RedTrf \rightarrow$ Port defined in Intent
22:     else
23:         Send $Pkt \leftarrow CNet$
24:     end if
25: else
26:     Go to $Step12$
27: end if
28: $End$
```

1) APPLICATION INSTALLATION TIME

It is the total time the ONOS application would take to be installed on the SDN controller after successful compilation without any error.

2) FLOW INJECTION TIME

It is the total time the flow rule would take to be injected in the SDN switch after getting translated from intent.

3) HOP COUNT

The average rise in the number of hops in paths (in percentage) produced due to the port redirection, that could be taking the extended route towards destination port address, hence, utilizing the system's additional resources.

4) INCREASE IN END TO END TRANSMISSION DELAY

Legitimate packets might suffer this delay because of application installation on the ONOS SDN controller.

IX. EXPERIMENTAL RESULTS & EVALUATION

A. EVALUATION

We analyzed the traffic flow between the nodes to evaluate our experimental setup. Modified traffic is observed

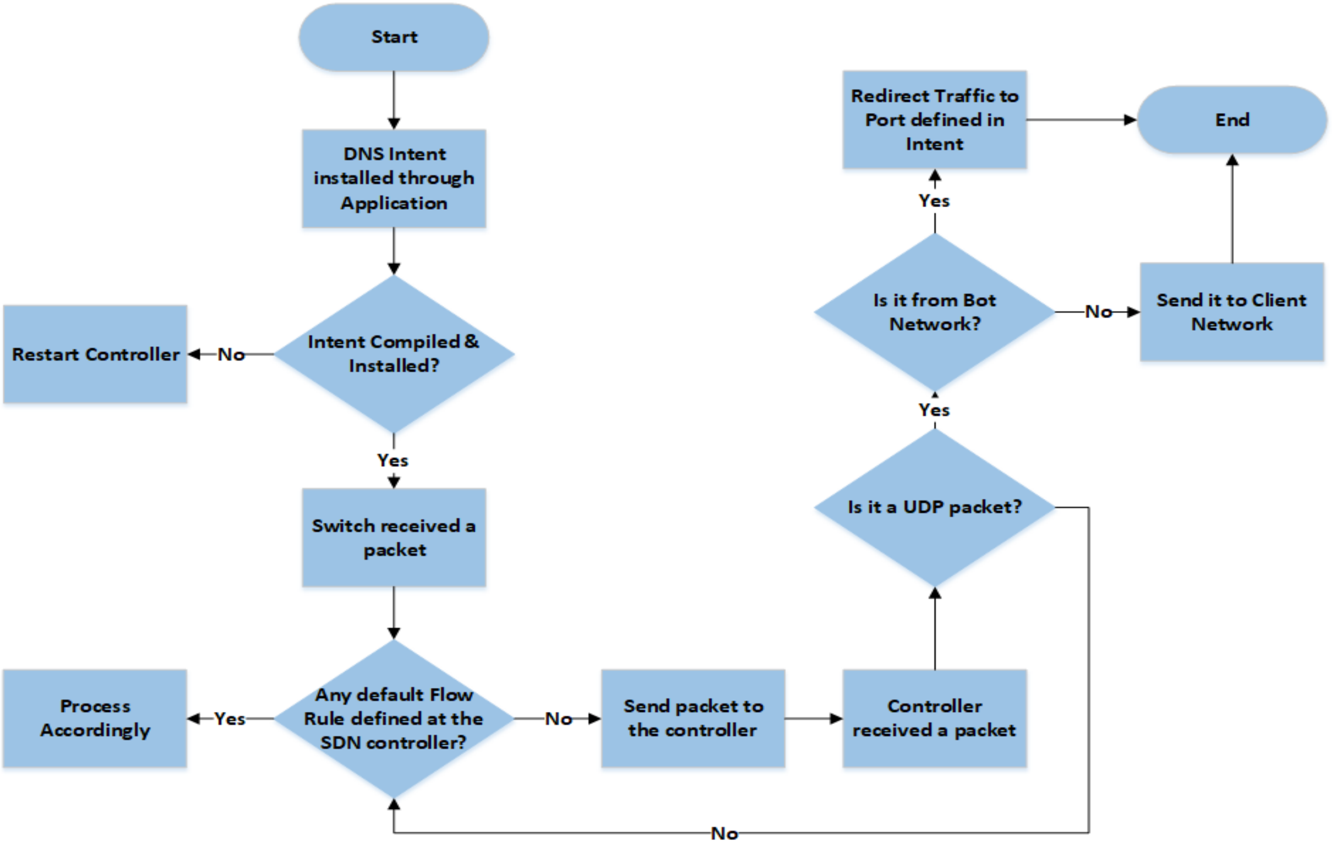
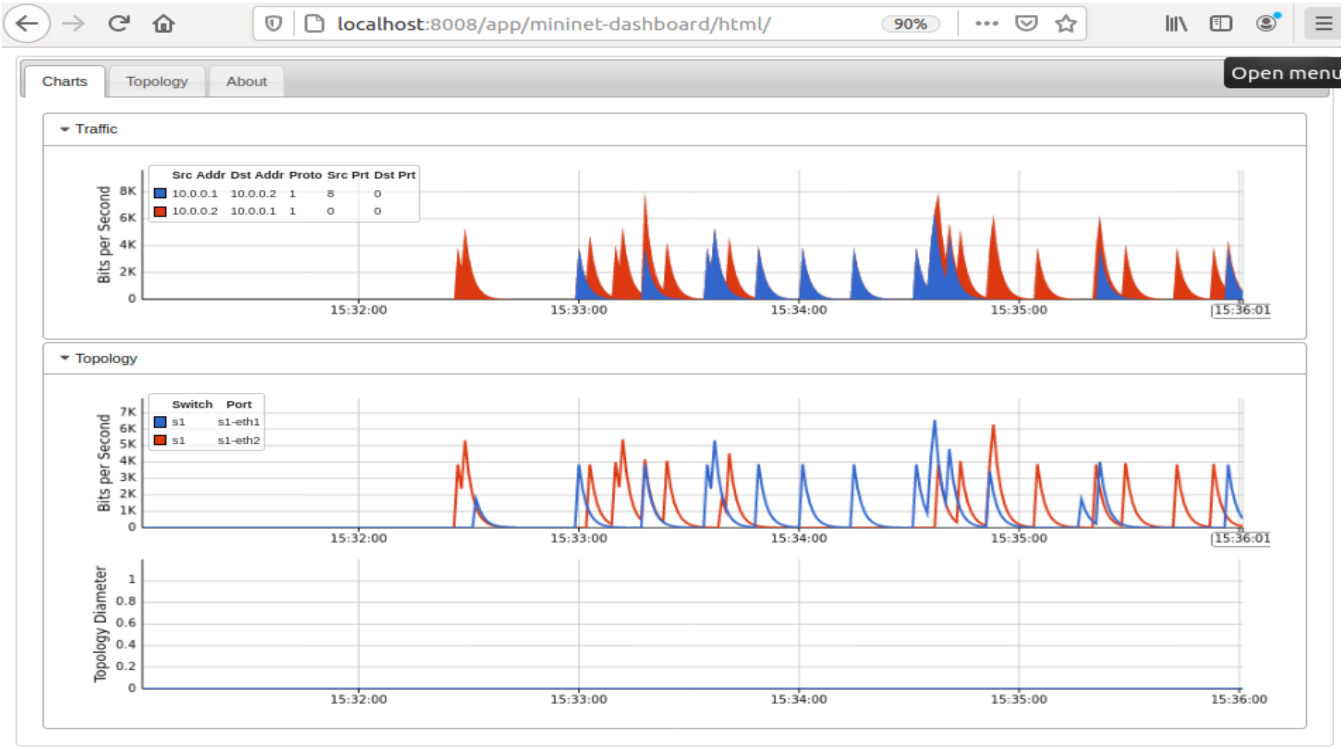

Start
DNS Intent installed through Application
Intent Compiled & Installed?
No
Restart Controller
Yes
Switch received a packet
Any default Flow Rule defined at the SDN controller?
Yes
Process Accordingly
No
Send packet to the controller
Controller received a packet
Is it a UDP packet?
No
Yes
Is it from Bot Network?
No
Send it to Client Network
Yes
Redirect Traffic to Port defined in Intent
End

FIGURE 3. Process for DNS port redirection.



	Src Addr	Dst Addr	Proto	Src Prt	Dst Prt
🟦	10.0.0.1	10.0.0.2	1	8	0
🟥	10.0.0.2	10.0.0.1	1	0	0

	Switch	Port
🟦	s1	s1-eth1
🟥	s1	s1-eth2

FIGURE 4. Mininet dashboard without application of intents.

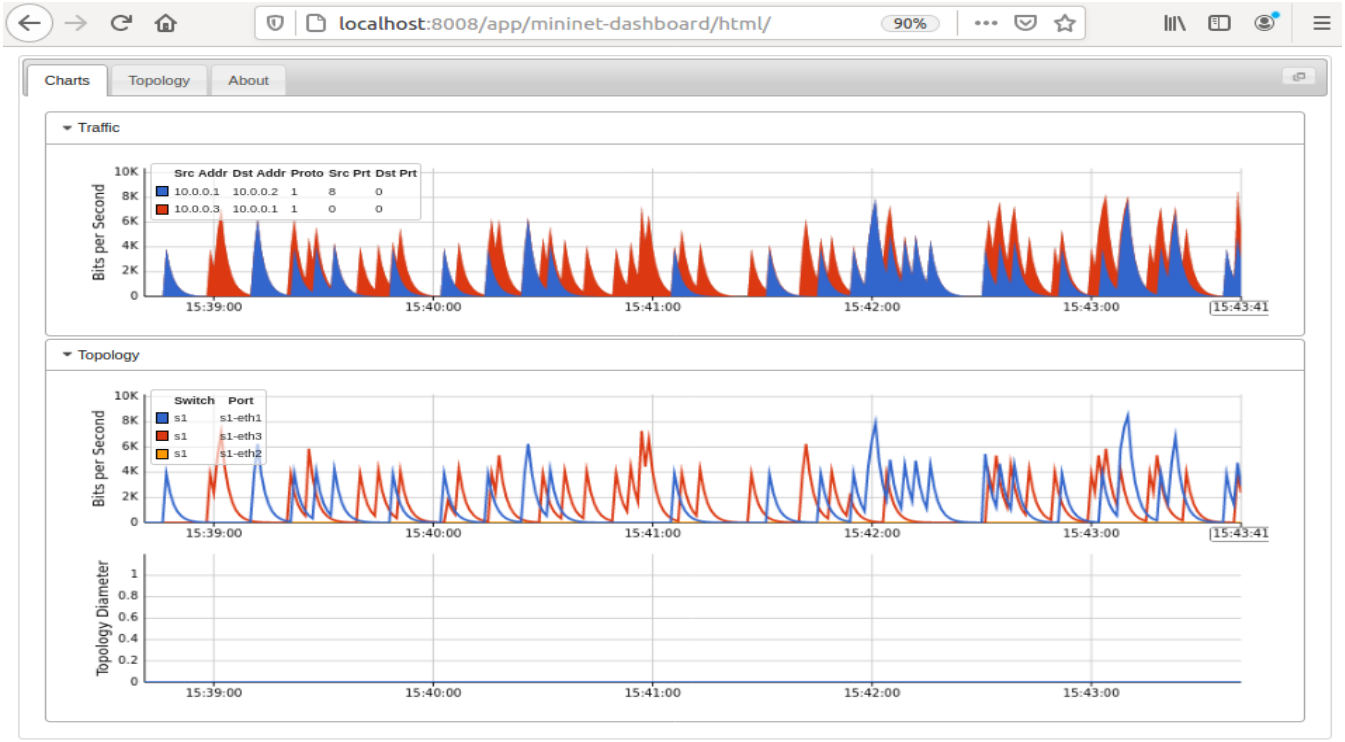


FIGURE 5. Mininet dashboard after application of forward intent.

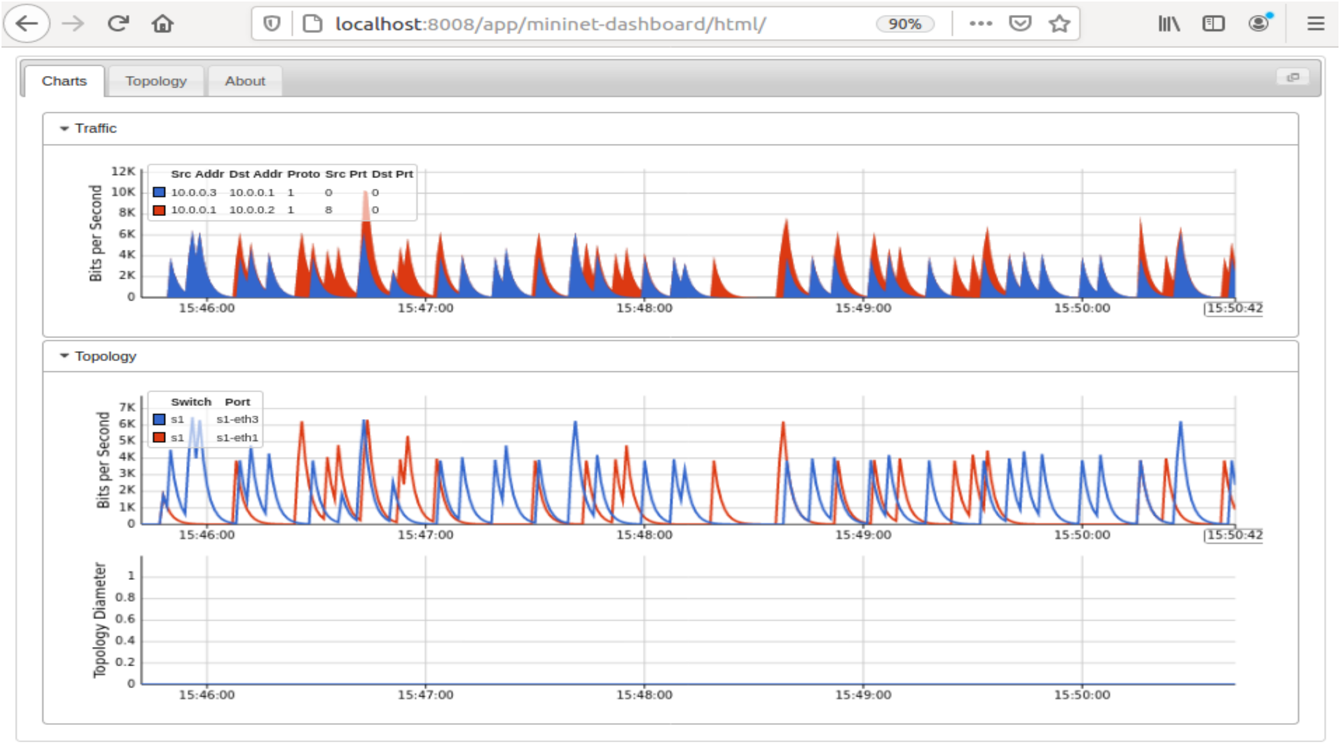
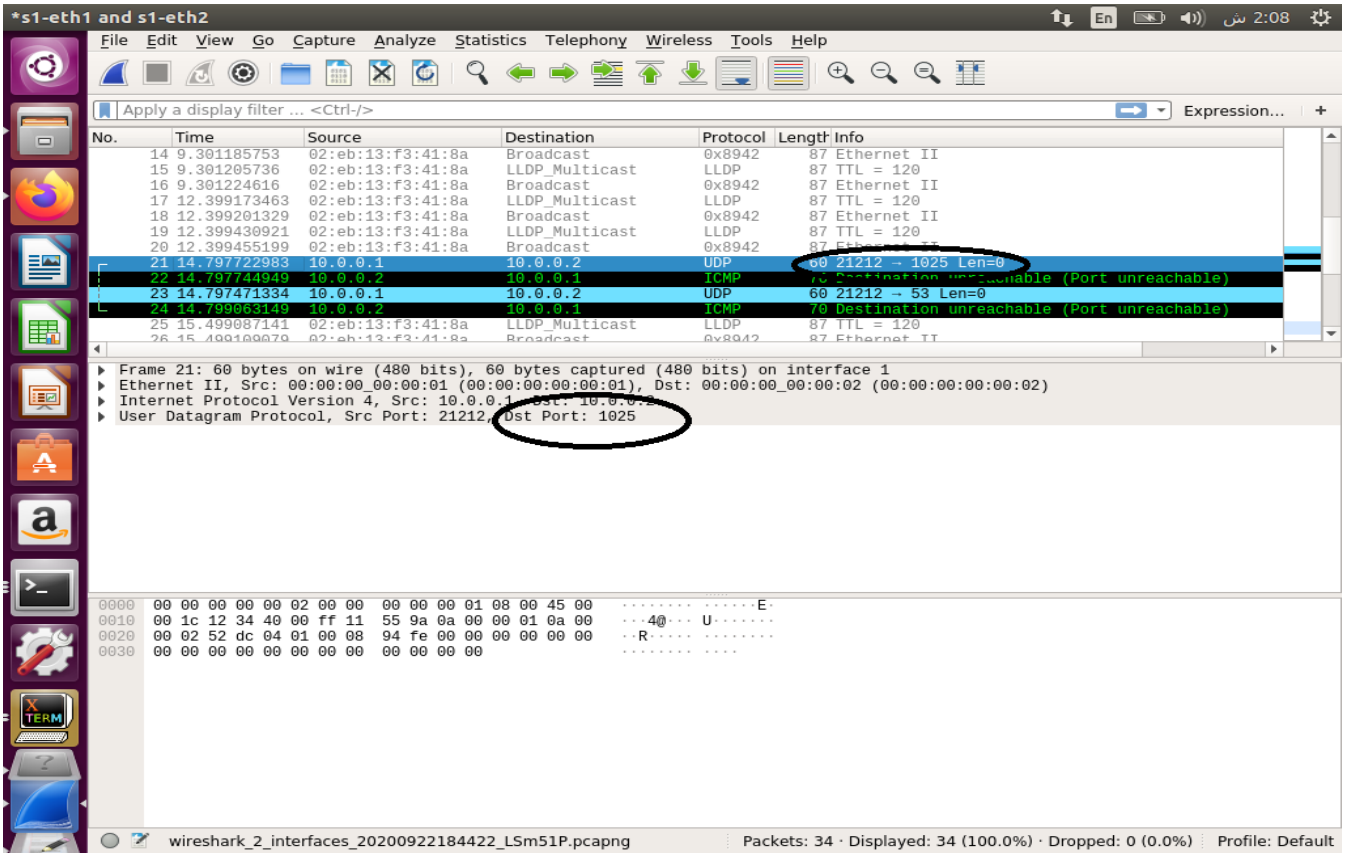


FIGURE 6. Mininet dashboard after application of reverse intent.

to determine whether the required results are obtained as expected or not. Moreover, redirected traffic is noticed through graphical results to achieve refined output. Further, we investigated systematically as shown in **TABLE 1** to



```
No.   Time           Source              Destination      Protocol  Length Info
   14 9.301185753    02:eb:13:f3:41:8a   Broadcast        0x8942    87 Ethernet II
   15 9.301205736    02:eb:13:f3:41:8a   LLDP_Multicast   LLDP      87 TTL = 120
   16 9.301224616    02:eb:13:f3:41:8a   Broadcast        0x8942    87 Ethernet II
   17 12.399173463   02:eb:13:f3:41:8a   LLDP_Multicast   LLDP      87 TTL = 120
   18 12.399201329   02:eb:13:f3:41:8a   Broadcast        0x8942    87 Ethernet II
   19 12.399430921   02:eb:13:f3:41:8a   LLDP_Multicast   LLDP      87 TTL = 120
   20 12.399455199   02:eb:13:f3:41:8a   Broadcast        0x8942    87 Ethernet II
   21 14.797722983   10.0.0.1            10.0.0.2         UDP       60 21212 → 1025 Len=0
   22 14.797744949   10.0.0.2            10.0.0.1         ICMP      70 Destination unreachable (Port unreachable)
   23 14.797471334   10.0.0.1            10.0.0.2         UDP       60 21212 → 53 Len=0
   24 14.799063149   10.0.0.2            10.0.0.1         ICMP      70 Destination unreachable (Port unreachable)
   25 15.499087141   02:eb:13:f3:41:8a   LLDP_Multicast   LLDP      87 TTL = 120
   26 15.499109079   02:eb:13:f3:41:8a   Broadcast        0x8942    87 Ethernet II

▸ Frame 21: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 1
▸ Ethernet II, Src: 00:00:00_00:00:01 (00:00:00:00:00:01), Dst: 00:00:00_00:00:02 (00:00:00:00:00:02)
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
▸ User Datagram Protocol, Src Port: 21212, Dst Port: 1025

0000  00 00 00 00 00 02 00 00  00 00 00 01 08 00 45 00   ········ ······E·
0010  00 1c 12 34 40 00 ff 11  55 9a 0a 00 00 01 0a 00   ···4@··· U·······
0020  00 02 52 dc 04 01 00 08  94 fe 00 00 00 00 00 00   ··R····· ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····

wireshark_2_interfaces_20200922184422_LSm51P.pcapng    Packets: 34 · Displayed: 34 (100.0%) · Dropped: 0 (0.0%)    Profile: Default
```

FIGURE 7. Output of DNS application monitored on wireshark.

compare with the existing Crossfire protection schemes so that it is proved that our framework provides a thorough safeguard for large geographical networks.

Moreover, we noticed the CPU utilization of the system which runs ONOS. The obtained outcomes demonstrated that it ranges between 20% to 30% of CPU however, it could increase up to 80% in case a single SDN controller gets the entire received traffic which poses the threat of a single point of failure. Our proposed framework eradicated this issue by using Distributed Control Plane, therefore, CPU utilization was reduced back to the above-defined range. The major differences between the two below defined mechanisms include the modification through RestAPI is done using JSON script on the controller while modification for DNS port is executed through ONOS application utilizing Maven and Java code. Moreover, the first one redirects all the traffic for the victim node towards the shadow node, while, second is only redirecting DNS traffic from the default port to the defined redirected port.

B. INTENT-BASED MODIFICATION THROUGH RestAPI

We need two intents for the redirection of hosts. Intents are injected through ONOS RestAPI in JSON formats. After that, we may check the states of our intents through ONOS CLI to verify whether they have been installed properly or not.

After successful installation of both the intents, redirected traffic can be seen on terminals of nodes h2 and h3. Here, it is noticed that intents select ports intelligently after compilation of JSON scripts and then, translate them into flow rules. That's why node h2 is receiving traffic the same as node h3 but the response is generated from h3 rather than h2 and h1 nodes perceive it to be the authentic response from h2 instead of h3.

In **Figure 4**, ordinary traffic is seen through various utilizations of sflow-rt, for example, Mininet-dashboard and Flow-graph have been used for noticing the nodes' conduct before applying intents after exploiting ping facility between h1 and h2.

From **Figure 5** and **Figure 6**, we detected that even though h1 needs traffic flowing through h2 however it is diverted towards h3, although h2 is also receiving the same response as h3. Likewise, h1 is getting feedback from h3 however it is straightforwardly made imperceptible from h1.

C. INTENT-BASED MODIFICATION THROUGH ONOS APPLICATION USING DNS PORT REDIRECTION

Since the DNS intent is now successfully installed through the ONOS application. We evaluated results by using a Linux GUI packet generator and Wireshark tools. After setting the required input fields, we observed the host interfaces. It was found out that the packet is successfully redirected to the port defined in intent as shown in **Figure 7**.

X. CONCLUSION & FUTURE RECOMMENDATIONS

A. CONCLUSION

This framework has been implemented using the Mininet emulator and ONOS SDN controller as a 3-node cluster for a distributed environment. Crossfire DDoS protection through Intent-based Modifications through RestAPI and ONOS application, are developed to redirect the traffic by using the integration of SDN, NFV, and IBN. Redirection is chosen because if we drop suspicious packets, there is a chance that illegitimate traffic could have been masqueraded as legitimate in another form to enter into the network. While after redirection, packets can be analyzed for creating more advanced security techniques to make it challenging for the attacker to reach the actual node and decreasing the overhead cost for the defender. The implemented proposed solution attained efficient results. Therefore, it provides defending mechanism at a low computation cost by using programmable intents instead of installing new hardware that could be expensive for the defender.

B. FUTURE RECOMMENDATIONS

Our proposed framework could be applied in networks specifically Cloud data centers and IoT systems as it is a combination of SDN and NFV that can be exploited for efficiency and optimal performance in networks. Network administrators have to deal with a variety of DDoS attacks with the increase in technology dependency. Hence, the proposed system with the incorporation of IBN can help networking engineers secure their organizations by implementing in real-time network environments where the traffic is extremely high. Moreover, the formal convergence proof of the proposed solution could credibly be done as an extension of this framework in the future.

REFERENCES

[1] M. Rezazad, M. R. Brust, M. Akbari, P. Bouvry, and N.-M. Cheung, "Detecting target-area link-flooding DDoS attacks using traffic analysis and supervised learning," in *Proc. Future Inf. Commun. Conf.* Cham, Switzerland: Springer, 2018, pp. 180–202.

[2] M. Suk Kang, S. Bum Lee, and V. D. Gligor, "The crossfire attack," in *Proc. IEEE Symp. Secur. Privacy*, May 2013, pp. 127–141.

[3] T. Benson, A. Akella, and A. Shaikh, "Demystifying configuration challenges and trade-offs in network-based ISP services," in *Proc. ACM SIGCOMM Conf. SIGCOMM (SIGCOMM)*, 2011, pp. 302–313.

[4] Q. Duan, E. Al-Shaer, and H. Jafarian, "Efficient random route mutation considering flow and network constraints," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2013, pp. 260–268.

[5] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Formal approach for route agility against persistent attackers," in *Proc. Eur. Symp. Res. Comput. Secur.* Berlin, Germany: Springer, 2013, pp. 237–254.

[6] A. R. Chavez, W. M. S. Stout, and S. Peisert, "Techniques for the dynamic randomization of network attributes," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Sep. 2015, pp. 1–6.

[7] U. Rauf, F. Gillani, E. Al-Shaer, M. Halappanavar, S. Chatterjee, and C. Oehmen, "Formal approach for resilient reachability based on end-system route agility," in *Proc. ACM Workshop Moving Target Defense*, Oct. 2016, pp. 117–127.

[8] R. Sahay, W. Meng, and C. D. Jensen, "The application of software defined networking on securing computer networks: A survey," *J. Netw. Comput. Appl.*, vol. 131, pp. 89–108, Apr. 2019.

[9] T. Benson, A. Akella, and D. A. Maltz, "Unraveling the complexity of network management," in *Proc. NSDI*, 2009, pp. 335–348.

[10] N. F. Virtualisation, "An introduction, benefits, enablers, challenges & call for action," SDN OpenFlow World Congr., Darmstadt, Germany, White Paper #1, Oct. 2012, p. 73.

[11] N. ISG, "Network functions virtualisation (NFV)-network operator perspectives on industry progress," ETSI, SDN OpenFlow World Congr., Frankfurt, Germany, White Paper #2, Oct. 2013. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper2.pdf

[12] A. Rafiq, A. Mehmood, T. A. Khan, K. Abbas, M. Afaq, and W.-C. Song, "Intent-based end-to-end network service orchestration system for multi-platforms," *Sustainability*, vol. 12, no. 7, p. 2782, Apr. 2020.

[13] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A survey of networking applications applying the software defined networking concept based on machine learning," *IEEE Access*, vol. 7, pp. 95397–95417, 2019.

[14] W. Rafique, X. He, Z. Liu, Y. Sun, and W. Dou, "CFADefense: A security solution to detect and mitigate crossfire attacks in software-defined IoT-edge infrastructure," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun.; IEEE 17th Int. Conf. Smart City; IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Aug. 2019, pp. 500–509.

[15] S. Achleitner, T. F. L. Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha, "Deceiving network reconnaissance using SDN-based virtual topologies," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 1098–1112, Dec. 2017.

[16] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A SDN-oriented DDoS blocking scheme for botnet-based attacks," in *Proc. 6th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2014, pp. 63–68.

[17] D. Ma, Z. Xu, and D. Lin, "Defending blind DDoS attack on SDN based on moving target defense," in *Int. Conf. Secur. Privacy Commun. Netw., 2014: Springer*, pp. 463–480.

[18] D. Belabed, M. Bouet, and V. Conan, "Centralized defense using smart routing against link-flooding attacks," in *Proc. 2nd Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2018, pp. 1–8.

[19] A. Aydeger, N. Saputro, and K. Akkaya, "A moving target defense and network forensics framework for ISP networks using SDN and NFV," *Future Gener. Comput. Syst.*, vol. 94, pp. 496–509, May 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 Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1838–1853, Jul. 2018.

[21] M. Beshley, A. Pryslupskyi, O. Panchenko, and H. Beshley, "SDN/Cloud solutions for intent-based networking," in *Proc. 3rd Int. Conf. Adv. Inf. Commun. Technol. (AICT)*, Jul. 2019, pp. 22–25.

[22] D. Comer and A. Rastegatnia, "OSDF: An intent-based software defined network programming framework," in *Proc. IEEE 43rd Conf. Local Comput. Netw. (LCN)*, Oct. 2018, pp. 527–535.

[23] T. Szyrkowiec, "Automatic intent-based secure service creation through a multilayer SDN network orchestration," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 4, pp. 289–297, Apr. 2018.

[24] Y. Tsuzaki and Y. Okabe, "Reactive configuration updating for intent-based networking," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2017, pp. 97–102.

[25] L. Pang, C. Yang, D. Chen, Y. Song, and M. Guizani, "A survey on intent-driven networks," *IEEE Access*, vol. 8, pp. 22862–22873, 2020.

[26] C. Trois, M. D. Del Fabro, L. C. E. de Bona, and M. Martinello, "A survey on SDN programming languages: Toward a taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2687–2712, 4th Quart., 2016.

[27] D. Sanvito, D. Moro, M. Gulli, I. Filippini, A. Capone, and A. Campanella, "ONOS intent monitor and reroute service: Enabling plug&play routing logic," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 272–276.

[28] WhatIs.com. (2020). *What is Intent-Based Networking (IBN)? Definition From WhatIs.com*. [Online]. Available: https://whatis.techtarget.com/definition/intent-based-networking-IBN

[29] A. Rafiq, M. Afaq, and W.-C. Song, "Intent-based networking with proactive load distribution in data center using IBN manager and smart path manager," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 4855–4872, Feb. 2020.

[30] S. M. Bellovin, "Using the domain name system for system break-ins," in *Proc. USENIX Secur. Symp.*, 1995, pp. 1–11.

[31] J. Jiang, J. Liang, K. Li, J. Li, H. Duan, and J. Wu, ‘‘Ghost domain names: Revoked yet still resolvable,’’ in *Proc. 19th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2012, pp. 1–13.

[32] D. Kaminskey, ‘‘It’s the end of the cache as we know it,’’ in *Proc. Black Hat Briefings*, USA, 2008.

[33] Z. Wang, H. Hu, and G. Cheng, ‘‘Design and implementation of an SDN-enabled DNS security framework,’’ *China Commun.*, vol. 16, no. 2, pp. 233–245, 2019.

[34] *Mininet Overview*. Accessed: Jun. 10, 2021. [Online]. Available: http://mininet.org/overview/

[35] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O’Connor, P. Radoslavov, W. Snow, and G. Parulkar, ‘‘ONOS: Towards an open, distributed SDN OS,’’ in *Proc. 3rd workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.

MUHAMMAD FARAZ HYDER received the M.Eng. degree in telecommunications engineering and the M.Eng. degree in computer systems engineering from the NED University of Engineering and Technology, in 2010 and 2014, respectively, and the Ph.D. degree in computer systems engineering (with specialization in cybersecurity) from NED University, in 2020. He has 16 years of experience in the industry, academics, and research centers. His areas of research include cybersecurity, moving target defense (MTD), cloud security, software defined networking, network function virtualization, network and information security, privacy, and digital forensics.

TASBIHA FATIMA received the B.S. degree in computer science and information technology from the NED University of Engineering and Technology, in 2018, and the M.S. degree in information security from NED University, in 2020. She received two gold medals for her B.S. degree.