Detection of Distributed Denial of Service Attacks in Software Defined Networks

Lohit Barki[1], Amrit Shidling[2], Nisharani Meti[3], Narayan D G[4] and Mohammed Moin Mulla[5]
B V Bhoomaraddi College of Engineering and Technology, Hubli, Karnataka, India[1,2,3,4]
KLE Technological University, Hubli, Karnataka, India[5]
lohit.barki1994@gmail.com, amritlshidling@gmail.com, narayan_dg@bvb.edu, nisharanimeti@gmail.com and moinbvb@gmail.com

***Abstract*—Software Defined Network (SDN) architecture is a new and novel way of network management. In SDN, switches do not process the incoming packets. They match for the incoming packets in the forwarding tables and if there is none it will be sent to the controller for processing which is the operating system of the SDN. A Distributed Denial of Service (DDoS) attack is a biggest threat to cyber security in SDN network .The attack will occur at the network layer or the application layer of the compromised systems that are connected to the network.In this paper we discuss the DDoS attacks from the traces of the traffic flow. We use different machine learning algorithms such as Naive Bayes, K-Nearest neighbour, K-means and K-medoids to classify the traffic as normal and abnormal. Then these algorithms are measured using parameters such as detection rate and efficiency.The algorithm having more accuracy is chosen to implement Signature IDS and results of it are then processed by Advanced IDS which detects anomalous behaviour based on open connections and provides accurate results of the hosts specifying which hosts is involved in the DDOS attack.**

***Keywords*—SDN, DDoS, Machine learning algorithms, IDS**

I. Introduction

Software Defined Network(SDN) [1] is a new emerging technology which provides network virtualization. SDN consists of switches and controller.When packets arrives to SDN, it will be sent to the switch,if switch don't have flow entry for packet then it will be sent controller for furthur processing and table entry.Hence controller is heart of SDN, in which major functionality of SDN depends on it. One can control whole SDN if he/she gains access to controller. Hence providing security to controller is very important. A Distributed Denial of Service (DDoS) attack in software defined network(SDN) which is a major threat in the internet. Attacker attack by compromising multiple numbers of computers to send a flood of traffic to servers which is created intentionally or fully utilizing their bandwidth .The DDOS attack is to disturb the software defined network(SDN) or make unavailable of the software defined network(SDN).

If large number of users are accessing the same software defined network (SDN) at the same time then it will be a flash crowd. The flash crowd and the access patterns by this crowd help us to determine the DDOS attack in the software defined network (SDN). Initially the attacker will compromise the computers connected to the software defined network (SDN) and make it malicious systems and then the attacker install the tools that are required to attack the server. The attack traffic will be transmitted from compromising computers (zombies).In order to attack the network attacker make use of spoofed IP addresses so that the zombies are unique and minimize the danger that the attacker will not be traced. The attacker may attack by TCP syn attack or UDP flood attack.

The compromised systems used in homes, educational institutes and government organization can be used for DDoS attack. These systems are called as bots. Traditionally DOS attacks are started from the network layer by flooding UDP, SYN or ICMP messages [11]. When the attack at the network layer fail then the attacker moves to the application layer and floods HTTP GET messages and these messages are called as application layer DDoS. DDoS attack can be through TCP SYN,UDP flood attack, DNS reflection attack, HTTP flood attack, ICMP flood.

In this paper we designed new IDS which is implemented in controller. Our IDS consists of two modules. First module is advanced Signature based IDS which processes request and finds set of host have normal or anomalous behavior and if they found having anomalous behavior then these set of hosts given to the second module which checks packets sent by each host. Second module checks for open connections and shows exact result whether they have normal or anomalous behavior. In this IDS we given more importance to processing time, since controller is heart of SDN it should process fast as possible. We designed Module 1 with different algorithms and we choose best among them. By running Module 1 first we can reduce time of processing because we reduce number of hosts given for processing to Module 2.

II. Related Work

A. Software Defined Network

Software-Defined Networks (SDNs) lies specifically in their ability to provide network virtualization, dynamic network policy enforcement, and greater control over network entities across the entire network fabric at reduced operational cost. Protocols like OpenFlow focus specially on the above aspects. However, by centralizing the control plane, SDNs place great burden on the administrator to manually ensure security and correct functioning of the entire network. Compromised network entities can be used to ex-filtrate sensitive information,

978-1-5090-2029-4/16/$31.00 @2016 IEEE

implement targeted attacks on other users, or simply bring down the entire network.

B. Architecture Design

SDN architecture consists of infrastructure layer and control layer.

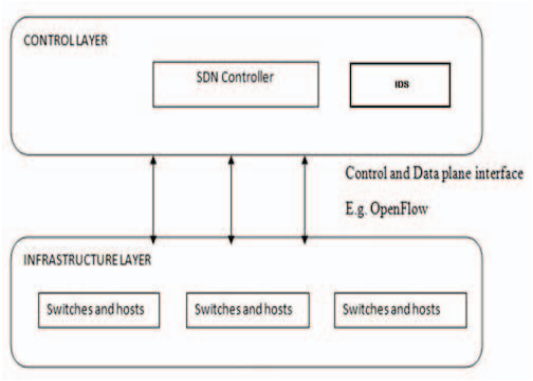

Figure 1: SDN Architecture

Figure 1 shows overview of the Architecture design. Segmentation of architecture design is further divided as follows:

- **Control plane**: The control plane is the part of a network that carries signaling traffic and is responsible for routing.
- **Infrastructure Layer**: The Infrastructure Layer(user plane, forwarding plane) is the part of a network that carries user traffic
- **IDS**: This modules contains code to determine intrusion.number of ICMP pings (echo request), which keeps the server bound in sending response. This paper looks at the specific problem of detecting security attacks on network topology and data plane forwarding originating within SDNs in real-time.The dataset is given for the training to the machine learning algorithm.

C. Machine Learning Algorithm

In this section few of the machine learning algorithms are discussed [3].

- **Naive Bayes**:Naive Bayes [4] is a conditional probability model, which assigns the class label to given data based on the probability.

$$P(C_k|x_1, ...x_n)$$

for each of K possible outcomes or classes. Since it uses probability the accuracy of classification is more but it takes more time for classification as training data increases. It works best when the attributes are independent of each other.

$$P(C_k|x) = \frac{P(C_k)P(x|C_k)}{\sum_{i=1}^{k} P(x|C_i)P(C_i)}$$

- **KNN**: K nearest neighbors [5] is a simple algorithm that classifies the data based on the similarity measure. We use distance as similarity measure. Data is classified based on the distance to its class label, with the data being assigned to the class most common amongst its K nearest neighbors measured by a distance function. Suppose K = 1, then the case is simply assigned to the class of its nearest neighbour.
- **K-means**: K-means clustering [6] aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. Given a set of data (x1, x2, , xn), where each data is a d-dimensional real vector, k-means clustering [7] aims to partition the n observations into (k_i=n)
sets C = {C1, C2, .., Ck} so as to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the K center). In other words, its objective is to find:

$$\sum_{i=1}^{k} \sum_{x \in C_i} ||x - Ci||^2$$

- **K-medoids**: K-medoids algorithm [8] shares the properties of the K-means algorithm. Instead of calculating the mean of items within the cluster, a representative item or medoid is selected from each cluster at each iteration. Medoids for each cluster are calculated by finding object i within the cluster that minimizes

$$\sum_{j \in C_i} d(i, j)$$

where Ci is the cluster containing object i and d(i,j) is distance between objects i and j.

D. Distributed Denial of service

DDoS attack [9] is the unusual traffic sent to the target. Under normal conditions, the bandwidth consumption rate is at an acceptable value and a certain pattern is present in the network activity. A sudden drop in performance of the network due increase in either traffic or delay or CPU utilization will often be considered abnormal. DDoS detection systems will be looking for such abnormalities in the network. The attack when directed to the network layer causes a bottleneck and when directed to the application layer causes exhaustion of CPU resources. Generally, the anomalies and the nature of data in the network are closely related. Hence understanding the type of data and its characteristics in the network can be termed as the first step in detecting anomalies. These characteristics can be packet header information, delay, packet size, protocol, etc. For instance, a server responding to TCP SYN requests is most likely to face the TCP SYN request flooding[10].

III. Proposed Work

The following section describes the proposed IDS using machine learning algorithms to detect DDoS attack in SDN.

We used machine learning algorithms to implement signature IDS and TCP open connections in Advanced IDS. We used Ryu controller to implement IDS.

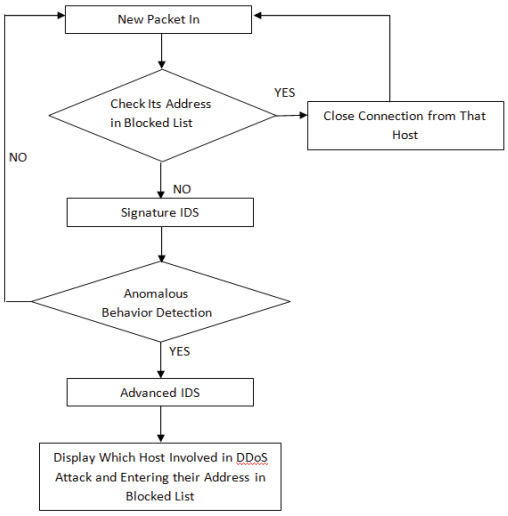

Figure 2: Flow diagram

Figure 2 Shows the flow diagram. Since we designed first module of IDS using machine learning algorithms we need training dataset for training these algorithms. We have taken real-time dataset which is traced file obtained from TCP traffic between Lawrence Berkley Laboratory and the rest of the world. The dataset is as shown below:

A. Data Set

In this subsection we will discuss dataset used for training machine learning algorithms.

Figure 5 shows that the dataset consists of records such

req_time	src_host	dst_host	fg
759103800.008185	1	2	0
0.010445	2	1	0
0.023775	1	2	1
0.026558	2	1	0
0.029002	3	4	1
0.032439	4	3	0
0.049618	1	2	0
0.052431	5	2	1
0.056457	2	5	1
0.057815	6	7	1
0.072126	8	9	0

Figure 3: Training dataset

as req_time, src_host, dst_host, fg which are request time for connection, source host ,destination host and flag bit respectively. But since we designed first module based on number of requests per ten seconds we processed training dataset as shown in below:

Figure 6 shows that the dataset consists of no_host_time, which is number of hosts connected in 10 seconds in peek or off-peek time and lb, which is class label. We considered peek time and off-time for request classification because traffic is going to vary based on time.

Figure 4: Processed dataset

B. Implemented algorithms

1) **Algorithm 1** Access control

When a host makes a request, its address is checked for presence in the blocked list and then the decision is taken whether to allow the request or not.

1: The address of each host in attacked list is given
2: When host sends request to a server, its address is checked with attacked list
if *it is in the blocked list* **then**
allow host to connect the server
else
Deny host from accessing server
end if

2) **Algorithm 2** Signature IDS using naive Bayes We use

Naive Bayes classifier to classify the incoming requests as anomaly or normal. Based on the obtained values the users sending anomaly requests are sent warning.

1: Training a naive Bayes classifier when controller starts
2: Counting the number hosts connected per 10 seconds
3: Then obtained value is classified by naive Bayes based on the probability and this value along with class label is appended in training dataset
if *packet is classified as anomaly* **then**
warning is given to the user
else
entry is made in the table of a switch
and sent through the port to a destination
end if

3) **Algorithm 3** Signature IDS using KNN

We use KNN classifier to classify the incoming requests as anomaly or normal. Based on the obtained values the users sending anomaly requests are sent warning.

11 **1**: Training a KNN classifier when controller starts
2: Counting the number hosts connected per 10 seconds
3: Then obtained value is classified by KNN based on the distance and this value along with class label is appended in training dataset
if *packet is classified as anomaly* **then**
warning is given to the user
else
entry is made in the table of a switch
and sent through the port to a destination
end if

4) **Algorithm 4** Signature IDS using K-means algorithm

The traffic is recorded and clusters are formed and their mean values are calculated. The nodes are then reclassified into clusters with the nearest mean value. The entire clusters are classified as normal or anomalous.

1: Recording the traffic
2: Making cluster out of recorded values as normal and anomalous
3: Calculating mean values of both clusters
4: Reassigning each node to cluster with nearest mean
5: Repeating step 3 until assigning of nodes in previous iteration is same as present iteration
6: Repeating from step 1
7: Based on traffic rate cluster will be declared normal or anomalous
8: Obtained value along with class label is appended in training dataset

5) **Algorithm 5** Signature IDS using K-medioids algorithm

Record the traffic, make clusters based on the traffic and calculate their mean. The new traffic data is recorded and the distance between new data point and the mean of the clusters is calculated. We then update the mean of the clusters. Finally obtained value along with class label is added in training dataset.

6) **Algorithm 6** Advanced IDS based on TCP three way handshaking

TCP traffic is recorded and header information is extracted. It is processed sending request to the response of SYNACK from server. If the hand shaking is not complete then the host is said to be an attacker.

1: Recording the traffic
2: Making cluster out of recorded values as normal and anomalous
3: Calculating mean values of both clusters
4: Read new traffic data
5: Calculate distance from new data point to mean of both clusters
6: Assign data point to the nearest cluster
7: Update mean value of cluster
8: Repeat from step 4
9: Based on traffic rate cluster will be declared normal or anomalous
10: Obtained value along with class label is appended in training dataset

1: Recording the TCP traffic
2: Extracting the header information from recorded data.
3: Processing the each record starting from sending request to the server with SYN message and response from the server with SYN-ACK message.
4: If the handshake is complete with one more SYN-ACK message then hand shaking is said to be completed otherwise the requesting host might be an attacker (spoofed host).
5: Based on TCP handshaking the host is said to be attacker or the normal user.

IV. Implementation

We used Ryu controller to implement access control, Signature IDS and Advanced IDS and tested DDoS attack using mininet environment with topology as in the Figure 5.

A. Topology

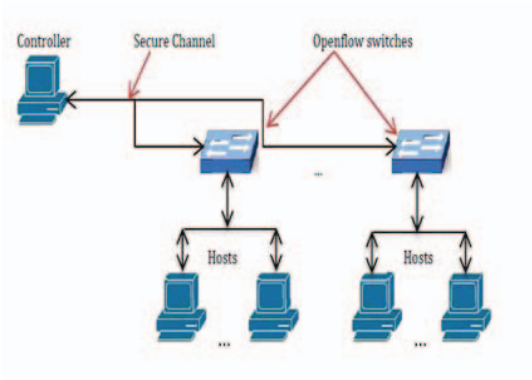

Figure 5: Simulation topology

Referring to the Figure 5 tables are present in each switch to show the ingress and egress paths of a packet for that switch. This property is used by Openflow and the controller is given access to these tables. The addition and deletion of

flow table entries from controller to switch occurs through a secure channel. On the arrival of a new packet, the Openflow switch looks into the flow table to find a match. If there is no match in the table, the packet is sent to the controller for further processing. The controller performs an action to detect DDoS like:

- Controller keeps access control list
- Each packet MAC address is checked with MAC address in Access Control Lists (ACL).
- If it is present, then it is blocked Else it is allowed to access the server.

Although controller allows hosts to access server it keeps track of each hosts because there may be chance of DDoS attack from new hosts which will be handled by IDS. If implemented IDS detects hosts showing anomalous behavior then these hosts are blocked and these hosts addresses are entered in ACL to prevent them from attacking again. We used ryu-manager controller to design IDS modules which is explained in methodology section.

V. Results and Discussion

The following section describes results obtained by simulating proposed IDS in Mininet environment. Mininet is a emulator which creates virtual hosts, switches, controllers, and links. Mininet hosts run Linux network software, and its switches support OpenFlow for custom routing and Software-Defined Networking.

Number of hosts	Actual Class Label	Naive Bayes		KNN		K-Means		K-Mediod	
2	Normal	Normal	TP	Normal	TP	Normal	TP	Normal	TP
3	Normal	Normal	TP	Normal	TP	Normal	TP	Normal	TP
4	Normal	Normal	TP	Normal	TP	Normal	TP	Normal	TP
6	Normal	Normal	TP	Normal	TP	Normal	TP	Normal	TP
8	Normal	Normal	TP	Anamoly	FP	Normal	TP	Normal	TP
9	Anamoly	Anamoly	TP	Anamoly	TP	Normal	FP	Normal	FP
10	Anamoly	Anamoly	TP	Anamoly	TP	Anamoly	TP	Anamoly	TP
11	Anamoly	Anamoly	TP	Anamoly	TP	Anamoly	TP	Anamoly	TP
15	Anamoly	Anamoly	TP	Anamoly	TP	Anamoly	TP	Anamoly	TP
20	Anamoly	Anamoly	TP	Anamoly	TP	Anamoly	TP	Anamoly	TP

Figure 6: Test dataset

Referring to the Figure 6 shows simulation results of the four machine learning algorithms by testing them with test dataset consisting of 50 tuples. When two hosts are connected in 10 seconds all algorithms show results as normal which is actual result. Similarly these algorithms show same results when 4,6,8 hosts are connected in 10 seconds. But when 9 hosts are connected in 10 sec NB shows results as normal which is True positive, but algorithm shows result as ”anomaly which is false positive.

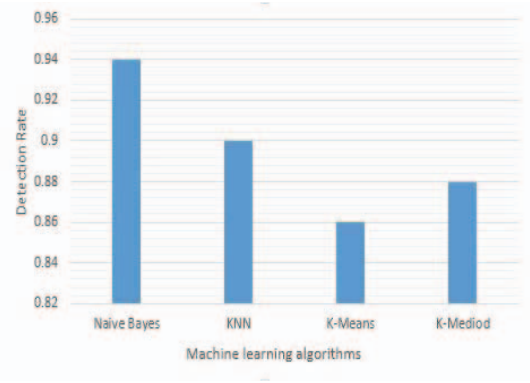

Figure 7: Detection rate VS M/C learning algorithm

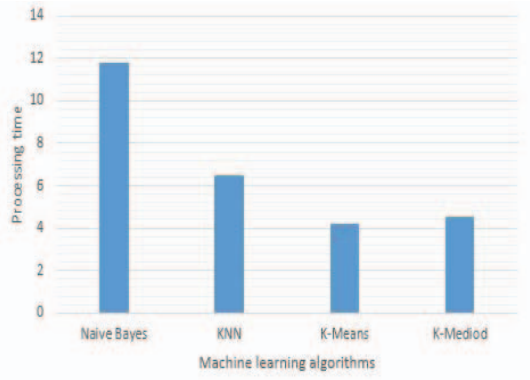

Figure 8: Processing time VS M/C learning algorithm

Figure 7 and Figure 8 show detection rate and processing time of each machine learning algorithms. Baysian algorithm has detection rate of which measure the accuracy of up to 94 percent. This accuracy is due to large dataset given as input to bayes classifier. But only disadvantage of this algorithm is it takes more time training and processing which takes 11.8 seconds. KNN has detection rate of 90 percent algorithm. The K-means(cluster algorithm) has less accuracy but since there is no need training it takes lesser time compared to both classifiers. So it faster than both classifiers. K-mediods approximately same accuracy as K-means but lesser than both classifiers .Since there is no training data is required for training, it will be faster than bayes and KNN.

From these results we have seen that all algorithms which we have implemented have both advantages and disadvantages. But in IDS since we give more attention to reduce false positive rate its better to implement IDS in SDN using nave Bayes which has higher accuracy compared to other machine learning algorithms. Only disadvantage of this algorithm is it takes more time to get trained. As training dataset gets increased, it takes more time for training but it is handled by training NB classier only once when controller gets started(result is takes more time to start controller) and secondly we need to handle is size of dataset which is handle by removing tuples from

training dataset which are old(by setting time threshold) since technology in networking is advancing rapidly resulting old data becoming opaque. Running signature based IDS before anomaly based IDS helps filter connections and then we can proceed for processing to detect open connections. Thus help in faster and accurate IDS.

The Module 1 implemented by nave bayes classifier classifies requests which help in detecting DDoS. If it finds set of hosts showing anomalous behavior than these hosts are given for processing to module 2.

Module 2 will make use of the TCP flag bits in order to detect whether the host is anomalous or the authorized user. When the host request the server for the service three way handshake should happen,in the sense initially host will send the request to server with SYN flag bit ,and server will acknowledge the requesting host will SYN-ACK bit then again the host will acknowledge the server with SYN-ACK bit, then the three way handshaking is said to be completed.

If the requesting host is not a genuine user or it is a spoofed host prepared to do DDoS attack then those host will not going to complete the three way handshake by not accepting the SYN-ACK message from server.

To detect this kind of DDoS attack the TCP handshake file is traced for complete three way handshake, if any of the host is not willing to complete the hand shaking then that host will be declared as the zombie or attacker machine. Then IP address of such host will be entered into the access control list.

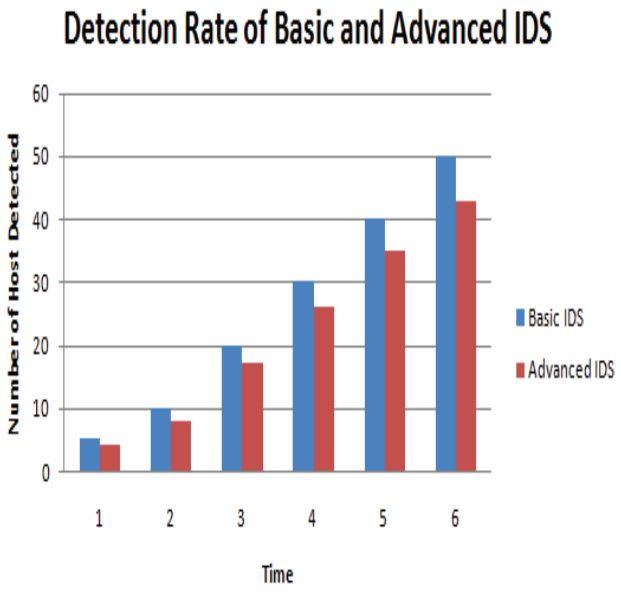

Figure 9: Final result of IDS

Figure 9 shows the result obtained from module 2, which shows the total number of TCP requests vs time.The red region in graph shows the number of host that complete the TCP handshaking process and are not attacker or spoofed host on the other hand the red area in graph shows the hosts which are not willing to complete TCP handshaking and which are may be the attacker or spoofed hosts.

VI. Conclusion and Future work

Software Defined Network (SDN) is an emerging architecture that is dynamic, manageable, cost effective and adaptable, making it ideal for high bandwidth, dynamic nature of todays applications. The goal of SDN is to allow network engineers and administrators to respond quickly to changing business requirements. This paper provides security to SDN controller. Further paper shows how DDoS attacks can be detected by classifying the incoming requests using various machine learning algorithms. The implementation of IDS in SDN using machine learning algorithms has shown better results. We implement the proposed mechanism using Mininet and POX controller. Results show the effectiveness of the solution by trying it on different topologies.

As a future work we plan to use other machine learning algorithms like hidden Markov model to detect the DDOS attacks.

References

[1] Kreutz, Diego, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103, no. 1 (2015): 14-76.

[2] Jarraya, Yosr, Taous Madi, and Mourad Debbabi. "A survey and a layered taxonomy of software-defined networking." IEEE Communications Surveys Tutorials 16.4 (2014): 1955-1980.

[3] S. Umarani, D. Sharmila, Predicting Application Layer DDoS Attacks Using Machine Learning Algorithms World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:8, No:10, 2014.

[4] Murphy, Kevin P. "Naive bayes classifiers." University of British Columbia (2006).

[5] Leung, K. Ming. "k-Nearest Neighbor algorithm for classification." Polytechnic University Department of Computer Science/Finance and Risk Engineering (2007).

[6] Ng, Andrew. "CS229 Lecture notes." CS229 Lecture notes 1.1 (2000): 1-3.

[7] Mark Hebster, "K-means algorithm - GI07/M012 - UCL Computer Science." University college London, 2014.

[8] Reynolds, Alan P., Graeme Richards, and Vic J. Rayward-Smith. "The application of k-medoids and pam to the clustering of rules." International Conference on Intelligent Data Engineering and Automated Learning. Springer Berlin Heidelberg, 2004.

[9] Mousavi, Seyed Mohammad. "Early Detection of DDoS Attacks in Software Defined Networks Controller." PhD diss., Carleton University Ottawa, 2014.

[10] Haopei Wang,Lei Xu, Guofei Gu, FloodGuard: A DoS Attack Prevention Extension in Software-Dened Networks.

[11] Dimitrios Gkounis, Cross-domain DoS link-ooding attack detection and mitigation using SDN principles Master Thesis MA-2013-18 October 14, 2013 to April 13, 2014.