International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

Software Defined Network: DDoS Attack Detection

Mr. Ajinkya Patil[1], Mr. Pratik Jain[2], Mr. Ravi Ram[3], Mr. Venkatesh Vayachal[4], Prof. S. P. Bendale[5]

[1,2,3,4]B. E. Student, Dept. of Computer Engineering, NBN Sinhgad School of Engineering, Ambegaon, Pune – 411041, Maharashtra, India
[5]Professor, Dept. of Computer. Engineering, NBN Sinhgad School of Engineering, Ambegaon, Pune – 411041, Maharashtra, India

---

Abstract – *Software Defined Networking (SDN) is a cloud computing approach for programmatically configuring and managing the networks. It enables easy and efficient troubleshooting from a single point in the network. As SDN follows centralized architecture by separating the forwarding process of network packets from routing process. It also enables security risks such as Denial-of-Service Attacks on the network, the main goal of the attack is to flood the network resources in such a way that the target machine overloads the network and its resources making it unable to handle the requests from legitimate users. The attacker mostly floods the network but sometimes it may also load malicious codes in the network, further disrupting the network until it fails.*

This paper aims to address those risks and focusses on Distributed Denial-of-Service (DDoS) Attack specifically. Timely detection of DDoS attack is important to prevent the system failure and information leak. Various algorithms like TCM-KNN and DPTCM-KNN are used for detection of attack in the network traffic flow. The comparison between the two algorithms is done using parameters like length of packet and response time of the packet from the source. The use of two parameters in DPTCM-KNN algorithm makes the detection more accurate and efficient than TCM-KNN algorithm.

Key Words: Software Defined Network (SDN), Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), Attack Techniques, Detection Algorithms.

1. INTRODUCTION

As the continuous development of Internet technology, the network consisting of networking devices, has increased suddenly. Every office, home, institute, various companies are constantly connected to the Internet. As for the networks, they are still constructed using switches and routers and configuration of these types of networks are done manually. This introduces some risks such as vulnerabilities in old networking devices, traffic flow clogging, interfaces and hosts error, controller failures, also providing attackers to target servers.

As for internet security there are several threats, Distributed Denial of Service (DDoS) attacks is most dangerous from the other attacks. [1] A DDoS attack is attempt to disrupt the normal traffic flow in a network, by targeting the server (central network), or a service by flooding the target with fake packets to exhaust the resources of victims, such as CPU, memory and network bandwidth.

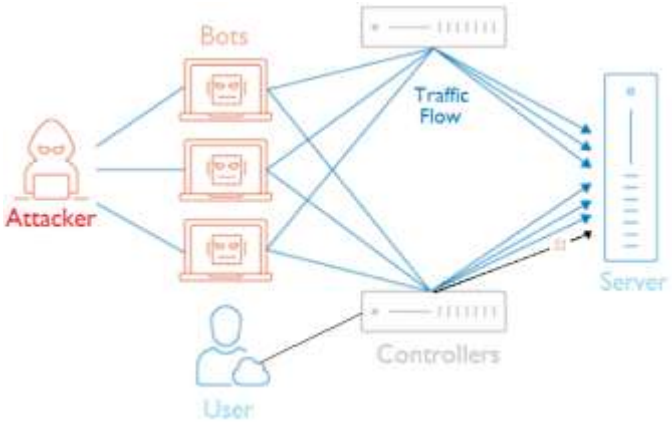

Fig – 1: Distributed Denial-of-Service Attack

A DDoS attack requires an attacker to gain control of the network devices to perform an attack. Devices in the network such as computers, networking devices, Internet of Things (IoT) devices are infected with malware, making them into bots. The collection of these bots is called botnet. When an attacker gains control over a botnet, he starts the attack by sending instructions remotely to the devices. The devices then flood the network with sending requests to the target server or network. Once a botnet is created the malware keeps running in the background either waiting for instructions form the botnet header (attacker) or automated program by sending continuous requests to the server. Once a machine is affected by malware, separating the attack traffic from normal traffic is difficult. There are many countermeasures proposed for these types of attacks but very few of them have been implemented because of their deployment complexities as well as prohibitive operational cost. One of the reasons is that such approaches require placing high end-equipment at routers and switches and sometimes even require human intervention, which increases extra storage and computational costs.

1.1 Software Defined Network

A Software Defined Network (SDN) is an architecture, [2] whose goal is to create an open and programmable network which can be managed easily as the network grows and/or its needs changes. The centralized control of SDN provides the organization a better and useful approach to control and manage the network (traffic engineering, security, and so on).

The aim of the SDN is to create an environment where the network evolves as the software evolves.

The SDN model separates the network control logic and forwarding logic, mainly referred as Southbound APIs and Northbound APIs, the Southbound APIs have devices which receive packets, perform operations on packets, find out its destination, then sending those packets to forwarding devices by making changes to packet header. This process of forwarding of packets is instantaneous so when handling many such packets there can be cases where the packets may not be legitimate. To overcome this SDN Controllers are used, the controllers contains the programmable logic which control the flow of rules for any data flow or network traffic. The Northbound APIs are the business logic or networking application which actually perform operations on the transferred packets and responses are generated accordingly.

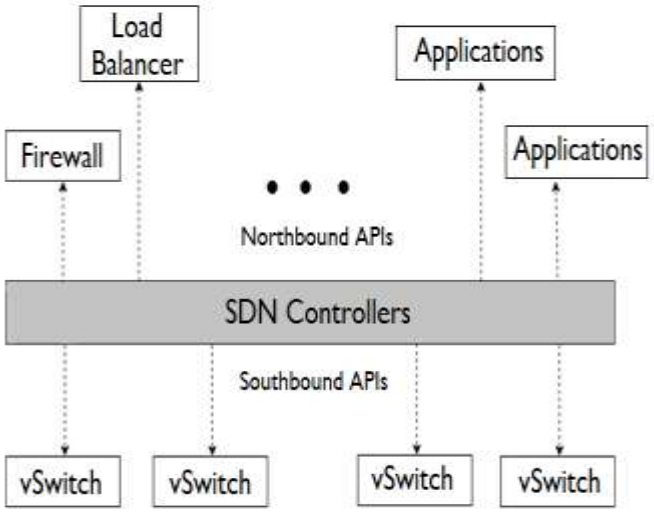

Fig – 2: Software Defined Architecture

The attackers aim is to overflow the network with fake packets to make the network unusable for the flow legitimate packets. [3] The SDN controllers cannot detect a DDoS attack by traditional methods because:

First, the switches cannot determine a malicious packet. The attacker hides himself so well in subnets and uses low-traffic flows to perform an attack, the switches cannot detect each packet flowing through each subnet and determine whether it is legitimate or not.

Second, it is not possible for the controller to determine based on the incoming packets, whether the network is under a DDoS attack.

Third, the attacker may not attack the controller directly but based on protocols such as UDP, TCP, etc.

In this paper, we propose an effective detection of attack detection in SDN flows, based on different algorithms. The algorithms are tested and compared with other algorithms.

2. DDOS ATTACKS AND EXISTING SOLUTIONS

2.1 DDoS Attack Techniques

Nowadays, various methods for performing an attack in the network to malfunction the working of a server. [4] They are:

A. SYN Flooding

In SYN Flooding, attacker sends SYN packets to the target containing incorrect source address which makes it impossible to receive SYN/ACK packets. As the ACK will not complete the port remains halted till the connection times out.

B. ICMP Flooding

In this attack the attacker sends multiple number of ICMP packets to the targets machine with fake return address, unable to transfer any data the server cannot handle other services. The bandwidth is jammed and the legitimate requests are declined. By creating an opportunity, the attacker loads Virus/Malicious Program on another machine spreading the attack.

C. UDP Flooding

The goal of the attacker is to fill random ports of server with UDP packets, as the host searches for corresponding application and finds no application is listening for that port. The host replies ICMP Destination Unreachable packet. As this cycle continues the requests from other clients are ignored.

2.2 Existing Solutions

DDoS Solutions consists of certain components, such as Attack Detection, Defense, and IP traceback. There are various ways for the attacker to mask his IP address and try to hog the traffic flow. The main aim of any security in Software Defined Network is detect the attack. Some of the methods to detect DDoS are-

Document [5] proposes to secure the SDN control plane using packet-in filtering mechanism. The mechanism at first keeps the packet header information before forwarding it, and then filters out IP addresses which aren't present in the list.

Document [6] introduces to entropy variation method, where it is used to trace back the source of DDoS attacks. It is defined by the randomness of the flows on routers. It tries to create an attack pattern based on Entropy of flow metric. The flow metric for non-DDoS attack is stable, and low for DDoS Attacks.

Document [7] proposes the controllers to contain the detection method, specifically the Northbound Controller, as it receives all information of the traffic flow from the switches and routers directly it contains collection of behavioral data of legitimate users over period of time. It is done by

distinguishing between the flow's statistics. Doing so, generates rules which lowers the false alarm rate for attacks.

Document [8] uses feature selection and feature weight mechanisms which helps optimizing the algorithm also reducing the cost and improving performance. It finds out the deviations by calculating p values, between the normal sets. It is also called as Transductive Confidence Machine for K-Nearest Neighbors (TCM-KNN).

Document [9] merges Ant Colony Optimization Algorithm with K-Nearest Neighbor algorithm to improve the results, it is applied on various classifiers. As the misclassified features can be combined and used for improving the time. Similarly, the classification of features is done by ACO which selects classifiers which take less time to compute. It also does feature reduction using ID3 (decision tree) algorithm.

Document [10] aims on anomaly flow detection method by collecting information from the switches and controllers such as flow details, type, features and classifiers between multiple flows. The Double P-value of Transductive Confidence Machines for K-Nearest Neighbors (DPTCM-KNN) is used for classifying the features and network flow. It has robust real-timeliness and it learns easily. It improves its detection precision by taking two p values from the TCM-KNN algorithm. The two p values are concept of independence of estimation of absolute deviation, and strangeness of relative deviations.

2.3 Detection Mechanism based on SDN

The SDN Controllers are used to collect the information of flow feature vectors in large numbers. Using the collected data, the vectors are classified using two parts: preprocessing module and detection module. The preprocessing module uses methods such as standardizing and normalizing the feature flow vectors. Assume that each vector has t features (X_{pq} *(1 ≤ p ≤ n, 1 ≤ q ≤ t)).*

- Standardization

$$X'_{pq} = \frac{X_{pq} - Mean_q}{AvgDev_q}$$

where,

$$Mean_q = \frac{X_{1q} + X_{2q} + \cdots + X_{nq}}{n}$$

$$AvgDev_q = \frac{|X_{1q} - Mean_q| + \cdots + |X_{nq} - Mean_q|}{n}$$

In Eq. (1), $Mean_q$ is the mean value and $AvgDev_q$ is mean absolute deviation value. At the time of standardization, following points are also considered:

1. If $Mean_q = 0$, $X_{pq} = 0$
2. If $AvgDev_q = 0$, $X_{pq} = 0$

- Normalization

It normalizes the standardized data into [0, 1].

$$X''_{pq} = \frac{X'_{pq} - X_{min}}{X_{max} - X_{min}}$$

where,

$$X_{min} = \min\{X'_{pq}\}$$
$$X_{max} = \max\{X'_{pq}\}$$

After the preprocessing module finishes its operations next part is anomaly detection. The detection module uses the DPTCM-KNN algorithm to detect the traffic.

In the anomaly detection system, the elements are divided into two parts Normal and Abnormal Elements. Classification is done with the help of following definitions.

Euclidean Distance:

It is a fundamental step in KNN algorithm as it finds out the proximity or absolute spatial distance between multiple points. The formula is denoted by:

$$D^y_{pq} = \sqrt{\sum_{a=1}^{t} (X_{pa} - X_{qa})^2}$$

where, $X_{p\ and}\ Xq$ are used for extracting the features of p and q respectively, X_{pa} is ath value of set X_p and t is length of X_p

Strangeness:

The strangeness assumes that y and $-y$ is used to denote the normal and abnormal elements. When the results for calculating the Euclidean distance between p and other points is sorted in accordance with KNN algorithm. The ratio of the two is defined as strangeness in DPTCM-KNN.

$$\alpha_{py} = \frac{\sum_{q=1}^{r} D^y_{pq}}{\sum_{k=1}^{r} D^{-y}_{pq}}$$

where, k is number of nearest neighbor and py is strangeness. If the strangeness is greater that point doesn't belongs to specified category.

Independence:

The independence calculates the absolute distance between a point and category. It is sum of Euclidian distance between the point and its neighbors.

$$\theta_{py} = \sum_{q=1}^{r} D^y_{pq}$$

where, θ_{pq} is independence, and D_{pq} is distance between the point and nearest neighbor. If independence is smaller that point starts belonging to the category.

Double p Value:

The detection of the point is calculated by taking the probability of relative point to the neighboring points, the

degree of a point to a category. The greater the p-value the more it belongs to subsequent category. It is calculated using formula:

$$p_1(\alpha_p) = \frac{\#\{q = (1 \ldots n) : \alpha_q \geq \alpha_p\}}{n+1}$$

$$p_2(\theta_p) = \frac{\#\{q = (1 \ldots n) : \theta_q \geq \theta_p\}}{n+1}$$

where, α_p is strangeness and θ_p is independence relative y set to p^{th} point; # is element number.

p_1 (α_p) and p_2 (θ_p) are both used to calculate and considered for detection standard in DPTCM-KNN algorithm.

3. ALGORITHMIC APPROACH

In this section we discuss about the available attack detection techniques the TCM-KNN combines the Transductive Confidence Machine with K-Nearest Neighbor classification algorithm, it calculates the distance from an *unknown point* to class *u.* Taking the ratio of both, classification is done and *p-value* is found out from this information the class in which the *unknown point* belongs to is found out. In TCM-KNN algorithm a single precision technique is used to detect the attack on a network. That parameter is *Strangeness*. Only relative deviation between the *unknown point* and *class* is considered.

To further enhance the performance of TCM-KNN algorithm, an additional parameter is introduced, and the algorithm is modified accordingly. Thus, forming Double P-value of Transductive Confidence Machine for K-Nearest Neighbor (DPTMM-KNN) algorithm, the additional parameter is *Independence*, the accuracy of the algorithm is greatly improved. By combining both *Strangeness* and *Independence* a result is obtained.

Step 1: Standardization and Normalization

In the first step, we standardize the data and find the average deviation which is calculated using mean values and absolute deviation values from the Training Set. In the normalization step the minimum and maximum values are calculated those values are stored and further used for other operations.

Step 2: Calculate the Euclidean Distance of the Training Set

To calculate the Euclidean distance between the normal points and abnormal points in the training set. This is an important step which is used in the TCM-KNN algorithm and it's used as the basis for all the operations. This step classifies the detection points into normal and abnormal points.

Step 3: Calculate the Strangeness and Independence of the Training Set

The strangeness is calculated using the various points which are found using K-Nearest Neighbor algorithm and with Euclidean distance found out in *Step 2*. Using the formulas mentioned above for standardization and normalization.

Step 4: Calculate the Strangeness and Independence of detection point with reference to training Set

This is done by calculating the Euclidean distance between a *point a* and other points available in the training set. A list is created which sorts all the results based on distance values.

Step 5: Get the Double-p values of the detection points

The *p1* value it is calculated with the α_p which obtained from sorting the normal points in descending order in Strangeness Set. Similarly, the *p2* value is calculated with θ_p by Independence of normal set. Further using both the *p1* and *p2* values detection for an Attack is done.

Step 6: Classification of the data points

In this step, the classification if the detection points are done by using the parameters such as *Resp_Time*. By clustering the detection points by τ_1 and τ_2 with the confidence values like (0.01 or 0.05 or 0.1), the detection points are set as normal or abnormal.

4. ALGORITHM DESIGN

Algorithm 1: Attribute based DPTCM-KNN Algorithm

Parameters:

Number of nearest neighbors – (*k*), Number of normal subset *y* – (m_1), abnormal subset *-y* - (m_2), and confidence τ_1 and τ_2

Input:

Points for detection

Output:

Normal or Abnormal

1. **for** $p = 1$ to m_1 **do**
2. Calculate the Euclidean distances D_{pq}^{y} of points in normal subset
3. Save the results
4. **end for**
5. **for** $i=1$ to m_2 **do**
6. Calculate the Euclidian distance D_{pq}^{-y} of points in abnormal subset
7. Save the results
8. **end for**
9. Calculate α for each point in normal subset
10. Calculate θ for each point in normal subset
11. Calculate α_p for detection point p
12. Calculate θ_p for detection point p
13. Sort the strangeness and independence of point in normal subset
14. Calculate $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ of detection point p for every

attribute

If $p_1 \geq \tau_1$ && $p_2 \geq \tau_2$ **then**

    return normal

else

    return abnormal

end if

5. ANALYSIS AND EXPERIMENTS

Based on the experimental results performed on KDD'99 Dataset which is most used data set for anomaly detection methods and research. [11] KDD99 is feature extracted version of DARPA which is the base raw dataset. In the 1998 DARPA Intrusion Detection System Evaluation Program was prepared, consisting of an attack scenario to Air-Force base. It consists of host and network dataset files; Host dataset file is small dataset containing system calls. Network Dataset is mostly used because it consists of seven weeks of network traffic (TCP/IP dump)

Table -1: Confusion Matrix

Predicted Values	Actual Values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Confusion Matrix is used to measure the performance of the classification algorithm or classifier. We can derive many ratios from confusion matrix:

- **True Positive (TP):** These are cases in which we predict 'Yes' and the actual result shows 'Yes'.
- **True Negatives (TN):** These are cases in which we predict 'Yes' but the actual result shows 'No'
- **False Positive (FP):** Here we predict 'Yes', and actual result is 'No'. This is also called Type I error.
- **False Negative (FN):** Here we predict 'No' and actual result is 'Yes'. This is also called Type II error.

The major change between the DPTCM-KNN and the attribute based DPTCM-KNN is that double-precision values have been calculated for selected attributes instead of cumulating the attributes deviations. This gives the network more control over the traffic flow in the network as per the requirements, of the network administrator.

Since, Software Defined Network (SDN) majorly used in large organizations where the traffic flow is varying dynamically. The attribute wised double-precision value data packets help to detect the unwanted packets more precisely.

Considering the KDD'99 Dataset following are the results for randomly selected packets.

Table -2: Results

Packet Count	Actual		Predicted		Accuracy %
	Normal	DDoS	Normal	DDoS	
1996	1001	995	988	994	99.298597
1996	1001	995	1000	980	99.198397
2200	1162	1038	1112	1074	99.363636
2200	1162	1038	1138	987	96.590909
2332	1478	1022	1293	896	93.867925
2332	1478	1022	1392	893	97.984563
2856	1255	1601	1156	1563	95.203081
2856	1255	1601	1332	1493	98.914566
2652	1270	1382	1303	1198	94.306184
2652	1270	1382	1167	1176	88.348416

The attribute wise Double-Precision calculations for

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

individual packets has shown more accuracy and control over the data.

6. CONCLUSIONS

In this paper, we have studied various methodologies which are used for detection of Distributed Denial-of-Service (DDoS) Attacks on Software Defined Network (SDN), based on the findings and results we have concluded that the Attribute based Double P-value of Transductive Confidence Machines for K-Nearest Neighbors method gives more efficient way to find out anomalous flow in Software Defined Network.

REFERENCES

[1] A. S. Patil, P. S. Jain, R. G. Ram, V. N. Vayachal, S. P. Bendale (2018) Detection of Distributed Denial-of-Service (DDoS) Attack on Software Defined Network (SDN), IRJET

[2] Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z. (2018). A detection method for anomaly flow in software defined network. IEEE Access.

[3] Nadeau, T. D., & Gray, K. (2013). SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies. "O'Reilly Media, Inc.".

[4] Dong, P., Du, X., Zhang, H., & Xu, T. (2016, May). A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. In

Communications (ICC), 2016 IEEE International Conference on (pp. 1-6). IEEE.

[5] Kamboj, Priyanka & Chandra Trivedi, Munesh & Kumar Yadav, Virendra & Kumar Singh, Vikash. (2017). Detection techniques of DDoS attacks: A survey. 675-679. 10.1109/UPCON.2017.8251130.

[6] Kotani, D., & Okabe, Y. (2014, October). A packet-in message filtering mechanism for protection of control plane in openflow networks. In Proceedings of the tenth ACM/IEEE symposium on Architectures for networking and communications systems (pp. 29-40). ACM.

[7] Mousavi, S. M. (2014). Early detection of DDoS attacks in software defined networks controller (Doctoral dissertation, Carleton University).

[8] Dong, P., Du, X., Zhang, H., & Xu, T. (2016, May). A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. In Communications (ICC), 2016 IEEE International Conference on (pp. 1-6). IEEE.

[9] Li, Y., & Guo, L. (2008, March). TCM-KNN scheme for network anomaly detection using feature-based optimizations. In Proceedings of the 2008 ACM symposium on Applied computing (pp. 2103-2109). ACM.

[10] Jaiswal, S., Saxena, K., Mishra, A., & Sahu, S. K. (2016, March). A KNN-ACO approach for intrusion detection using KDDCUP'99 dataset. In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on (pp. 628-633). IEEE.

[11] Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z. (2018). A detection method for anomaly flow in software defined network. IEEE Access.

[12] S. P. Bendale, J. R. Prasad, "Security threats and challenges in Future Mobile Wireless Networks", IEEE International Conference proceeding GCWCN, 2018-19.