Proposal Defense for the degree of Master in Computer Engineering

Optimizing of Controller Placement based on Hybrid GENCLUST Algorithm for DDOS Mitigation on Software Defined Network

Sahaj Shakya
(2019-1-39-0020)

Nepal College of Information Technology
Faculty of Science and Technology
Pokhara University, Nepal

January, 2022

ABSTRACT

Today's widely used networks are complex and difficult to manage. To implement high-level network guidelines in traditional IP-based networks, network administrators must configure individual network devices with manufacturer-specific commands. As a result, it becomes very difficult to implement the desired policies and reconfigurations of network devices in today's IP-based networks. Software Defined Networking (SDN) has been adopted for the Flexible Internet and has the potential to be used for the next generation of the Internet by using a controller to separate the control plane from the data plane. A distributed denial of service (DDoS) attack can make on the SDN controller unable to process legitimate flow requests from the switch. The main approach is to protect a controller from DDoS attacks is based on attack detection which results in a high rate of false-negative and false-positive results. Existing mitigation techniques are basically based on external and additional resource or network traffic analysis and tend to be computationally intensive or have a high rate of false positives and / or false positives. The management of 150 switches in a system can only handle up to 20,000 new flow requests. Therefore, the controller must do a lot of work to handle network traffic efficiently. However, additional computational overhead can occur for multiple parameters and optimizing the controllers' positions. The purpose methods reduce the overhead in three steps, firstly selecting the controller for overhead traffic, secondly using genetic algorithm to elect the leader and third select the optimize position for controller from clustered controllers.

Keywords: SDN, DDOS, K-means Clustering, Genetic Algorithm, OpenFlow Controller

Table of content

List of Figures

List of abbreviation/acronyms

DDOS	Distributed Denial of Service
SDN	Software Defined Network
ANN	Artificial Neural Network
ANP	Analytic network process
SOM	Self-Organizing Map
DR	Detection Rate
TP	True Positive
FN	False Negative
FP	False Positive

CHAPTER 1

INTRODUCTION

A distributed denial of service (DDoS) attack occurs when one or more attackers attempt to block the delivery of a service. This can be achieved by forcing access to virtually everything, including servers, devices, services, networks, applications, and even specific transactions within an application. A DoS attack is a system that sends malicious data and requests. DDoS attacks come from multiple systems. Impacts range from service interruptions to entire websites, applications, and even business failures, from minor annoyances.

1.1 Statement of the problem

Distributed denial of service (DDoS) attacks are on the rise, and many organizations are either completely unprepared or poorly defended. The impact of a DDoS attack usually consists of loss of sales and customers, damage to the brand, and use as a smoke screen for further inbound attacks. Companies with business-critical online resources are not only exposed to greater losses, but are also more likely to be attacked by attackers. For this reason, enterprises need to be proactive in addressing DDoS threats [1].

In DDos detection and prevention based on Ann, there are mainly two issues

I. Maximum Controller detection overhead

The management of 150 switches in a system can handle up to 20,000 new flow requests Therefore, the controller must do a lot of work to handle network traffic efficiently. However, additional computational overhead can occur for multiple parameters which might impact controller performance. The number of parameters used for DDoS detection can be reduced in an ANN-based way. However, it can also be improved by using multiple machine learning algorithms to reduce the complexity of the controller. Similarly, multiple OpenFlow controllers can be also used to mitigate the problem [2].

II. False Negative and False Positive Value

False negative values are common when the attack rate parameter is set to a low value. However, an attack with a low packet flow rate will do less damage to the victim's

system [3]. However, this problem can be mitigated by using multiple machine learning algorithms [4].

1.2 Research objectives

The major objectives of this paper are

- To mitigate the Single Point Failure and overhead of Controller
- To reduce the false positive detection in case of Slow Attack
- To introduce Multiagent system in SDN
- To effectively Detect and Mitigate DDos attacks

1.3 Significance/Rationale of the study

With the rapid development in the field of IT infrastructure, the size of the network, its complexity has increased many times. This makes it increasingly difficult to guarantee key network characteristics such as integrity, confidentiality, authentication, information availability, and non-repudiation. In recent years, many researchers and industries have shifted their focus to developing more robust, scalable, and more secure networks. Today's widely used networks are complex and difficult to manage. To implement high-level network guidelines in traditional IP-based networks, network administrators must configure individual network devices with manufacturer-specific commands. As a result, it becomes very difficult to implement the desired policies and reconfigurations of network devices in today's IP-based networks [5].

The latest advances, such as SDN (Software Defined Networking), are a step towards the dynamic and centralized nature of networks compared to traditional network static distributed environments. Recently, Software Defined Networking (SDN) has been adopted for the Flexible Internet and has the potential to be used for the next generation of the Internet. The concept is to control the network through software. This approach facilitates network management and enables new applications in virtual environments. SDN was developed with extended network support from the central controller. SDN is a new network architecture that gives hope to an efficient network infrastructure. First, vertical integration is eliminated by separating the control plane (network control logic) from the data plane (routers and switches that route network traffic according to

the network control logic). This isolation control then provides a simple packet forwarding device that simplifies flexibility, implementation speed, programmability, and network management with a logically centralized controller and network logic installed on the network switch. The SDN architecture can improve the security of the network with the help of a centralized controller, but it creates global transparency for the network and, if necessary, traffic routing rules. However, security issues still remain in the SDN. [4].

However, SDN still has its own concerns and challenges regarding network security, scalability, and support capabilities. Security is paramount because of all these issues. Since the central controller is responsible for managing the network, a failure of this controller will affect the entire network. Central control and communication between the controller and the switch can be the target of advanced DDoS attacks.

Detecting flood-based distributed denial of service (DDoS) attacks is one of the biggest challenges for Internet security today. The DDoS attack mechanism is based on exploiting the huge resource asymmetry between the Internet and the victim's server limitations in handling a large number of fake requests. As a result, system resources are exhausted and the victim is taken from the Internet, so legitimate user requests are not processed [6].

With the advent and development of software-defined networks over the last decade, the ability to combat DDoS attacks in cloud environments has expanded. SDN has been determined to be an integral part of cloud and service providers that make networks programmable [7]. One of the problems is changing the packet header fields, similar to regular fields. As a result, it is very difficult to distinguish between legitimate regular traffic packets and useless packets sent from the compromised host to the victim. The second problem is the large number of packets that need to be analyzed. These are challenges that make detection difficult and slow response times.

The SDN controller can be a single point of failure. Security challenges are expected to increase as SDN technology becomes increasingly available. These SDN vulnerabilities focus on different layers of the SDN architecture. Analysis different vulnerabilities, security challenges can be summarized at different levels [4].

CHAPTER 2

LITERATURE REVIEW

Machine learning algorithms can use training data to automatically create classification models and use flow functions to classify network flows. In a real-time network environment, machine learning algorithms can detect known and unknown DDoS attacks. In the area of network security, the benefits of this SDN allow developers to easily and flexibly update their classification mechanisms to detect anomalous attacks on the network control plane. The SDN controller quickly collects and analyzes information from the switch and sends operational decisions back to the switch. Due to this flexible and effective process, the detection of SDN-based DDoS attacks has recently attracted the attention of the research community.

Initial was of mitigation mechanism uses HTTP-based and XML-based DDoS attacks preventions against cloud computing environments using a filter tree. This tree-based filtering scheme to prevent DDoS attacks uses five levels of filtering to mitigate the impact of HTTP-based and XML-based DDoS attacks. This tree-based DDoS attack prevention approach with filtering analyzes suspicious packets in the puzzle resolver to address issues caused by malicious data packets generated based on SOAP headers. This filter tree scheme identifies the IP address that was initially initiated by the malicious message to send the puzzle, and the puzzle sent is resolved to determine the actual client [8].

In case of "Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow", the researcher has used multiple parameters for calculation of the current state of the network. Researchers have presented a lightweight method for detecting DDoS attacks based on traffic flow characteristics that extract such information with very little effort compared to traditional approaches [6]. This is made possible by the use of the NOX platform, which provides a programmatic interface that facilitates the processing of switching information. Other important contributions by researchers include high detection rates and very low false alarm rates achieved by flow analysis using self-organizing maps.

OpenFlow-based SDN protection against DDoS attacks was provided to prevent problems arising from smooth packet replay attacks. This OpenFlow-based SDN framework leverages the mutual benefits of the data plane and control plane used in the investigation to control the

strength of DDoS attacks. This OpenFlow-based SDN framework is also important for managing and monitoring traffic flows that utilize SDN in the cloud. A fast and accurate SDN-based DDoS attack prevention scheme has been proposed to solve the problems caused by flood-based DDoS attacks using the process of entropy variation. It has been found that this entropy coefficient of variation used in this accurate SDN-based DDoS attack defense scheme can enable an accurate and clear classification of normal and malicious traffic. The researcher found that the false positive rate of this entropy variable factor-based DDoS attack mitigation scheme was reduced by 12% compared to the mitigation framework focused on the OpenFlow-based monitoring process [9].

As per “Evaluating the Controller Capacity in Software “, an important issue with the OpenFlow architecture is the capacity of the controller. This can be defined as the number of switches that the controller can manage. This white paper models the switch-to-controller flow setup requirements as a batch arrival process Mk / M / 1. In addition, queuing theory is used to analyze controller performance to derive an equation for average flow uptime. In the situation of limited flow rate setting time, the number of switches is determined, which provides a way to evaluate the capacity of the regulator. The centralized controller is only responsible for creating policies. The researcher state that on management of 150 switches, only 20,000 new flow requests can be handled. However, this additional computational overhead for multiple parameters can impact controller performance. [1].

As per the paper “QoS improvement with an optimum controller selection for software-defined networks”, the controller has multiple functions that guide the network from the center point and respond to updates related to topology changes. However, the ability to support these features is strong on one controller and weak on another. Choosing the best SDN controller can be considered a multiple-criteria decision-making problem (MCDM), as multiple controllers and each controller have many features. Here the researchers have proposed a two-step approach for choosing an SDN controller. First, the researcher classifies the controllers using the Analytical Network Process (ANP) according to the qualitative characteristics that affect the performance of these controllers, and then perform a performance comparison to see the improvement in QoS. Controllers with high weights from feature-based comparisons are analyzed quantitatively by experimental analysis. The researcher's main contribution is to

confirm the applicability of ANP to controller selection in SDN, taking into account feature and performance analysis in real-world Internet and Brite topologies [3]. Choosing the best controller with ANP results in faster topology discovery times and delays in normal and traffic load scenarios. Researchers have also seen an increase in throughput with the controller, which makes good use of the central processing unit.

KNN classifies flows by measuring the distance between flow feature vectors. It's simple and effective [10]. Peng et al. We proposed a KNN-based method using a transudative confidence machine (TCM) for SDN anomaly detection [11]. Nam et al. Combined ANN and SOM to address the DDoS flood. Linear ANNs are very time consuming, so the Kth Dimensional tree (KD tree) stores training points in a tree structure for quick queries on ANNs [4]. However, the KD tree must create an index chain for all training units. Changes in training sessions affect detection accuracy [3].

Shin et al. Have proposed a system called Avant-Guard that can identify DDoS of TCP SYN floods including alarm services [12]. Line Switch then extended it to include a statistics collector. Also, Kotani et al. We proposed a packet filter mechanism to prevent attacks by Open flow for TCAM. The implementation here is focused only on software switches and open switch [13].

PATGEN, a Protocol to reduce the effects of DDoS Attacks using an advanced Genetic algorithm with optimized and new operators, thereby significantly reducing the effects of attacks and increasing the efficiency of the multi-controller SDN [14].

CHAPTER 3

METHODOLOGY

KMEANS provides a highly efficient steepest descent method for all quadratic representation errors. However, not only do we need the parameter k, but we also assume the density similarity between the clusters. Therefore, it is greatly affected by noise. Despite being absorbed in a multi-start scheme, perhaps more seriously, it can often be locally optimally attracted. The purpose method combines the ability of genetic operators to integrate various solutions in the exploration space with the use of hill climbers. [15].

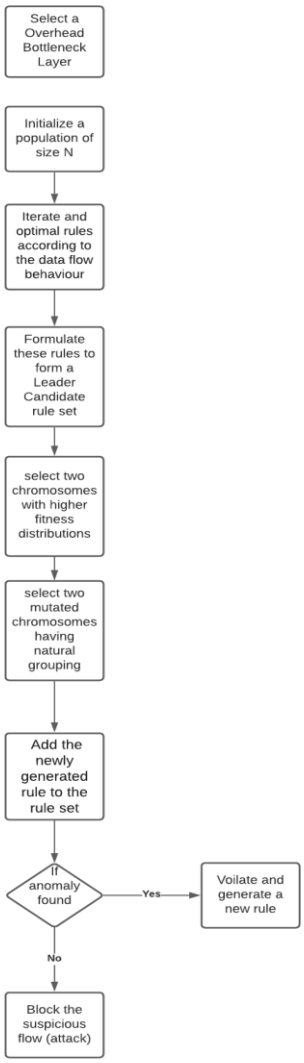

Fig 3.1 Proposed Methodology

The algorithmic procedure of interest is based on Patmos and Patgen Protocol incorporated with GENCLUST (Combination of Genetic Algorithm and K mean Clustering) to mitigate the effects of DDoS attacks on SDN controllers. These steps are divided into three phases: bottleneck discovery, selection, and creation. The bottleneck detection phase identifies the controller that is overloaded by the attack. In the election phase, a leader is elected to coordinate the clustering process. Finally, during the configuration phase, the controllers are clustered to mitigate the effects of the attack. [14]

During the bottleneck detection phase, the controller is used to identify overloaded controllers. During the election phase, the controller is used to assume the roles of candidate, leader, Vice leader (VL), and elite. As a candidate controller, leader with a high level of performance is selected. The Leader is a controller with a high level of performance. The VL is the second-best performance level controller to become a leader if the current leader is the target of a DDoS attack. The Elite Controller is a high-performance controller that acts as a member of the Coordinator Advisory Group (CAG) and also responsible to elects a new VL if a leader is targeted by a DDoS attack. During the configuration phase, the leader seeks the optimal cluster configuration using K-mean Clustering to mitigate the effects of the attack. The controller that the leader chooses to assemble the cluster acts as a worker to handle the flow requests received from the overloaded controller. To mitigate the congestion generated by DDoS attacks and increase system resource usage, Clustering Algorithm is used to select a cluster leader using a priority-based crash fault tolerance algorithm along with genetic algorithm to control the controller per cluster. The proposed method assumes load balancer before SDN to avoid the excessive number of flow tables and lack of resources in the network.

CHAPTER 4

EXPECTED OUTPUT AND VALIDATION CRITERIA

4.1 Statement of the problem

Where the causal links with the outcome must be thoroughly understood, prediction models should of the outcome, but should also use to represent the casual relationship within the data. When paired with powerful visualization tools and analytics, continuous data collecting, fine adjustment, and validation of the results, modeling and simulation can reveal insights.

The expected output of this proposed model is

- To detect how larger network data is spread
- To select the efficient controller positions
- To reduce overhead
- To provide a powerful visualization model for control and monitor of Network Flow

4.2 Validation Criteria (proposed)

However, no model is without flaws. Even today's most complex models, with hundreds of different parameters, are simplifications of reality. The variations in prediction are due to the complexity of reality where there are far too many variables for a perfect model to exist.

The validation of a predictive model entails

- dividing an initial sample set into training and validation datasets
- inferring a model with the training dataset
- evaluating the model's quality with the validation dataset using the aforementioned metrics.

The first steps in the process should be to start with a limited set of data for training. For later data points, and evaluate the accuracy of the predicted data points for optimal position. Following that, the same predicted data points are incorporated in the next training dataset, and additional data points are use to optimize the location. Similarly,

the throughput, Performance, the number of packets processed by the controllers, CPU speed, delay, Network fluctuations.

$$DR = \frac{TP}{TP + FN}$$

Fig4.1 Validation Criteria for Detection Rate

The efficiency of the detection mechanism was assessed by detection rate measurement (DR) and false alarm rate (FA).

$$FA = \frac{FP}{TN + FP}$$

Fig 4.2 Validation Criteria for False Alarm

TP (True Positive) attack traffic log classified as an attack, FN (False Negative) is a legitimate attack traffic log, FP (False Positive) is a legitimate traffic log classified as an attack, and TN (True Negative) is legitimate. Traffic Logs classified as legitimate.

Bibliography / References

[1] L. Yao, P. Hong and W. Zhou, "Evaluating the controller capacity in software defined networking," 2014 23rd International Conference on Computer Communication and Networks (ICCCN), 2014, pp. 1-6, doi: 10.1109/ICCCN.2014.6911857.

[2] Ali J, Roh Bh, Lee S, "QoS improvement with an optimum controller selection for software-defined networks,", 2019, PLOS ONE 14(5): e0217631, doi: 10.1371/journal.pone.0217631

[3] Nam, Tran Manh et al. "Self-organizing map-based approaches in DDoS flooding detection using SDN." 2018 International Conference on Information Networking (ICOIN) (2018): 249-254.

[4] Jagdeep Singh, Sunny Behal, Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions, Computer Science Review, Volume 37, 2020, 100279, ISSN 1574-0137, doi: 10.1016/j.cosrev.2020.100279.

[5] P. Preamthaisong, A. Auyporntrakool, P. Aimtongkham, T. Sriwuttisap and C. So-In, "Enhanced DDoS Detection using Hybrid Genetic Algorithm and Decision Tree for SDN," 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2019, pp. 152-157, doi: 10.1109/JCSSE.2019.8864216.

[6] R. Braga, E. Mota and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," IEEE Local Computer Network Conference, 2010, pp. 408-415, doi: 10.1109/LCN.2010.5735752.

[7] Harikrishna, P., Amuthan, A. SDN-based DDoS Attack Mitigation Scheme using Convolution Recursively Enhanced Self Organizing Maps. Sādhanā 45, 104 (2020). Doi: 10.1007/s12046-020-01353-x

[8] Karnwal, Tarun & Sivakumar, T. & Gnanasekaran, Aghila. (2012). A filter tree approach to protect cloud computing against XML DDoS and HTTP DDoS attack. Advances in Intelligent Systems and Computing. 182. 10.1109/SCEECS.2012.6184829.

[9] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," 2015 International Conference on Computing, Networking and Communications (ICNC), 2015, pp. 77-81, doi: 10.1109/ICCNC.2015.7069319.

[10] S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks," in IEEE Access, vol. 8, pp. 5039-5048, 2020, doi: 10.1109/ACCESS.2019.2963077.

[11] H. Peng, Z. Sun, X. Zhao, S. Tan and Z. Sun, "A Detection Method for Anomaly Flow in Software Defined Network," in IEEE Access, vol. 6, pp. 27809-27817, 2018, doi: 10.1109/ACCESS.2018.2839684.

[12] Seungwon, Shin & Yegneswaran, Vinod & Porras, Phillip & Gu, Guofei. (2013). AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks. 10.1145/2508859.2516684.

[13] Ambrosin, Moreno & Conti, Mauro & De Gaspari, Fabio & Poovendran, Radha. (2015). LineSwitch: Efficiently Managing Switch Flow in Software-Defined Networking while Effectively Tackling DoS Attacks. ASIACCS 2015 - Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security. 10.1145/2714576.2714612.

[14] Iranmanesh, Amir & Naji, Hamid. (2021). A protocol for cluster confirmations of SDN controllers against DDoS attacks. Computers & Electrical Engineering. Volume 93. 10.1016/j.compeleceng.2021.107265.

[15] Islam, Md & Estivill-Castro, Vladimir & Rahman, Md Anisur & Bossomaier, Terry. (2017). Combining K-Means and a Genetic Algorithm through a Novel Arrangement of Genetic Operators for High Quality Clustering. Expert Systems with Applications. 91. 402-417. 10.1016/j.eswa.2017.09.005.