A Framework for Identifying and Mitigating Malicious Flow in Software Defined Network Deployed over an IoT Ecosystem

Ravindra S
Research Scholar,
Visvesvaraya Technological University, Belagavi, Karnataka, India
Email: ravindraa.s@gmail.com

Dr. Shankaraiah
Professor and Head
Department of Electronics and Communication, SJCE, Mysuru, India

Abstract - With the rising demand for incorporating smartness over a security operation in networking technologies, Software Defined Network (SDN) has been witnessed to be extensively researched. SDN is one of the integral parts of operation in large scale networking operations e.g., Internet-of-Things (IoT), owing to its highly flexible communication protocols and centralized controlling features. Although there has been an extensive review of literature towards the security aspect of SDN, they do not offer full-fledged solutions especially if the adversary is unknown. Therefore, the proposed manuscript presents a novel framework capable of identifying the degree of severity of the attack from the rate of request message originated from a switch of SDN node and offers a decisive operation of resisting such malicious flows using an auxiliary agent. The auxiliary agent resides in the data plane and works alongside a switch to identify and confirm malicious flow. This information is further updated to the SDN controller, which can further take action that leads to isolating the adversary and allowing only flows with validated legitimacy. The study outcome shows, the proposed system excels better both in security and communication performance.

Keywords: Software Defined Network, Internet-of-Things, Security, Attack, Controller, Switch, Routing Request

1. Introduction

With the advancement of networking technologies, there is also a parallel rise of the adversaries and threats over large scale networks [1]. In order to make the network operate smarter and more efficiently, (SDN) has played a contributory role [2][3]. Being an emerging architecture, SDN offers adaptable and cost-effective features that separate the forwarding plane from the control plane. An SDN architecture is characterized by agility, centrally manageable, direct programmable, support towards open standard, etc. [4]. However, deployment of SDN architecture also invites various threats of various degrees of severity as very often; the attacks and threats are novel forms targeting the SDN controller system. The attacker usually targets the SDN controller node as it retains all sensitive information about the complete network. Therefore, compromising one SDN controller will mean compromising all the nodes that are connected to this SDN controller. There have been various studies being carried out towards securing SDN architecture [5][6], and it has been seen that there are mainly three different security issues that are required to be controlled effectively, e.g., i) offering availability of services hosted by the network, ii) safeguarding the system integrity, and iii) safeguarding the data confidentiality. It is evident that the SDN controller is an integral part of the security system, and it is required to precisely configured them as they can block the routing request as well as a specific route which are found to be vulnerable. It is also required for an SDN controller to carry out sufficient validation with a trusted platform.

There are various forms of attacks and vulnerabilities on the SDN system. From the viewpoint of attacks over data planes, the adversary can illegitimately access the physical medium from the network itself, leading to making the host a victim. Some examples of attacks over data plane in SDN are denial-of-service attack, man-in-middle attack, replay attack, etc. The attacks on the controller system lead to a rogue controller node, which led them to construct forged entries without being tracked by any engineers. There are different layers in SDN, and all of them have a very discrete demand for security. One such security demand is a configuration error. If this problem is not addressed, then it leads to the invitation of different other security threats. Denial-of-Service is one of the frequently reported attacks which use a flooding approach over the switch as well as over the controller, and hence complete layers of SDN will get affected. In the presence of any form of attacks over policy enforcement, the severity of the attack increases multifold over the entire upper three layers of SDN. In the case of authorization-

based intrusion, there is a higher possibility of restricted access to the controller system. Therefore, it is essential that to secure the complete environment of SDN, all the components within its architecture are also required to be protected. The essential target for any research idea is basically to protect the SDN controller from different forms of adversaries as the complete network, and this controller manages its associated operation. It is necessary to secure the complete operating system as any appearance of a state of compromise of the SDN controller will also lead to the vulnerability of the complete network connected to the controller. Existing studies also work towards encryption mechanisms in order to resist the intrusion of the adversarial flow of data while the emphases of the majority of the study are towards isolating the threats and protecting identity management [7] [8] [9]. However, existing approaches are never found at par with the increasing security threats. Therefore, the proposed system introduces a novel framework of a secured SDN architecture. The organization of this manuscript is as follows: Section 2 discusses the existing research techniques towards securing SDN while briefing of the identified research problem is carried out in Section 3. The proposed solution and its research methodology are discussed in Section 4, while the discussion of algorithm implementation is carried out in an elaborative way in Section 5. The obtained results of the simulation are briefed in Section 6, while a summary of the work is briefed in Section 7 as a conclusion.

2. Related Work

This section briefs about the existing studies towards security in SDN to continue our prior work [10]. The work carried out by Dai et al. [11] has addressed the problems associated with securing the hypervisor by using a tenant network for better access control. Considering different case studies of underwater vehicle networks, Eng et al. [12] have implemented a mechanism that can perform the identification of various dynamic conditions associated with system identification that contributes towards better network control. The adoption of blockchain towards securing SDN over the 5G network was carried out by Gao et al. [13]. The authors have used a trust model in order to offer secure communication in the network. Guo et al. [14] have used a deep reinforcement learning scheme for securing the communication in IoT when deployed alongside with SDN. The emphasis of the study is on data security with the aid of historical information from the traffic. Han et al. [15] have developed a framework that is capable of assessing the security aspect of the SDN over the cloud IoT system. The contribution of the security orchestration was studied by Hermosilla et al. [16], which is developed for intrusion detection systems in SDN along with Network function virtualization. A study towards mitigation a direct attack over SDN infrastructure was carried out by Li et al. [17], where the idea is to thwart the man-in-middle attack over an IoT architecture. The author has also used a bloom filter for performing encryption. Lin et al. [18] have developed a secure framework in SDN where the implementation is carried out using permutation of contents exclusively meant for transport security. Liu et al. [19] have developed a scheme which uses a mechanism for reduced delay where the integer linear formulation is carried out in order to offer enhanced data transfer security. Adoption of blockchain is also reported in the work of Medhane et al. [20], where the architecture of the SDN is rendered secured concerning its gateway operation. Consideration of the healthcare sector, along with data security, while sharing the informative contents over an SDN, is seen in the work of Meng et al. [21]. According to the study, the gateway system is incorporated with a novel firewall system which carries out authentication mechanism over different test environment. Ravi et al. [22] have developed a semi-supervised learning-based approach to prevent Distributed Denial-of-Service using cloud-based SDN architecture. Shafi et al. [23] have addressed the problem of anomaly detection where the controller system of SDN is run over fog computing, considering a case study of the IoT environment. The work carried out by Shahreza, and Ganjali [24] has addressed the issues associated with the identification of the scan for horizontal port over the home network. With the aid of the rate of dynamic sampling, the system renders low overhead over the controller. A unique work carried out by Sood et al. [25] has used mathematical modeling using the response time of the controller to carry out observation towards improving the security system of SDN when deployed over SDN architecture in IoT using proof of concept. A similar methodology has also been seen in the work of Yazdinejad et al. [26], where blockchain has been used for incorporating both security features as well as energy-efficient design over SDN architecture concerning its controller. The study outcome shows that it offers better security in comparison to the conventional blockchain-based approach. The work carried out by Yuan et al. [27] has used a secret sharing mechanism in order to forward secured data for effective control over the attacks of varied kinds. The work of Zarca et al. [28] [29] has developed a comprehensive architecture of the security, considering network function virtualization for dealing with existing cybersecurity problems over IoT architecture that uses SDN. The author has used a honeynet strategy in order to address this problem. Therefore, there exist different ranges of methodologies in current research work towards securing SDN architecture where each work has its own merits as well as constraint/limitation. The next section briefs about the identified research problem.

3. Research Problem

After reviewing the existing security approaches over an SDN architecture, following issues have been identified to be addressed in proposed solution viz. i) Majority of the approaches are based on pre-defined attack characteristic, which offers narrowed scope of security enhancement towards other attackers, ii) the major components of the study are mainly core architecture of SDN and very often essential components (e.g., switch, which his responsible for forwarding routing request) is ignored iii) the studies towards identification and resisting the dynamic threats over the flow is yet to be considered in the existing approach. The next section discusses the proposed solution towards resisting this flow-related threat over SDN architecture.

4. Research Methodology

The prime aim of the proposed system is to offer secure communication from the control plane to the data plane and application plane in existing SDN architecture when deployed over an IoT ecosystem. The idea is to ensure that switches that are present in the data plane should transmit the request of routing most securely towards the control plane while the routes for the newly arrived request are decided by the control plane, which further allocates rules of direction considering the presence of an application on the upper plane of application.

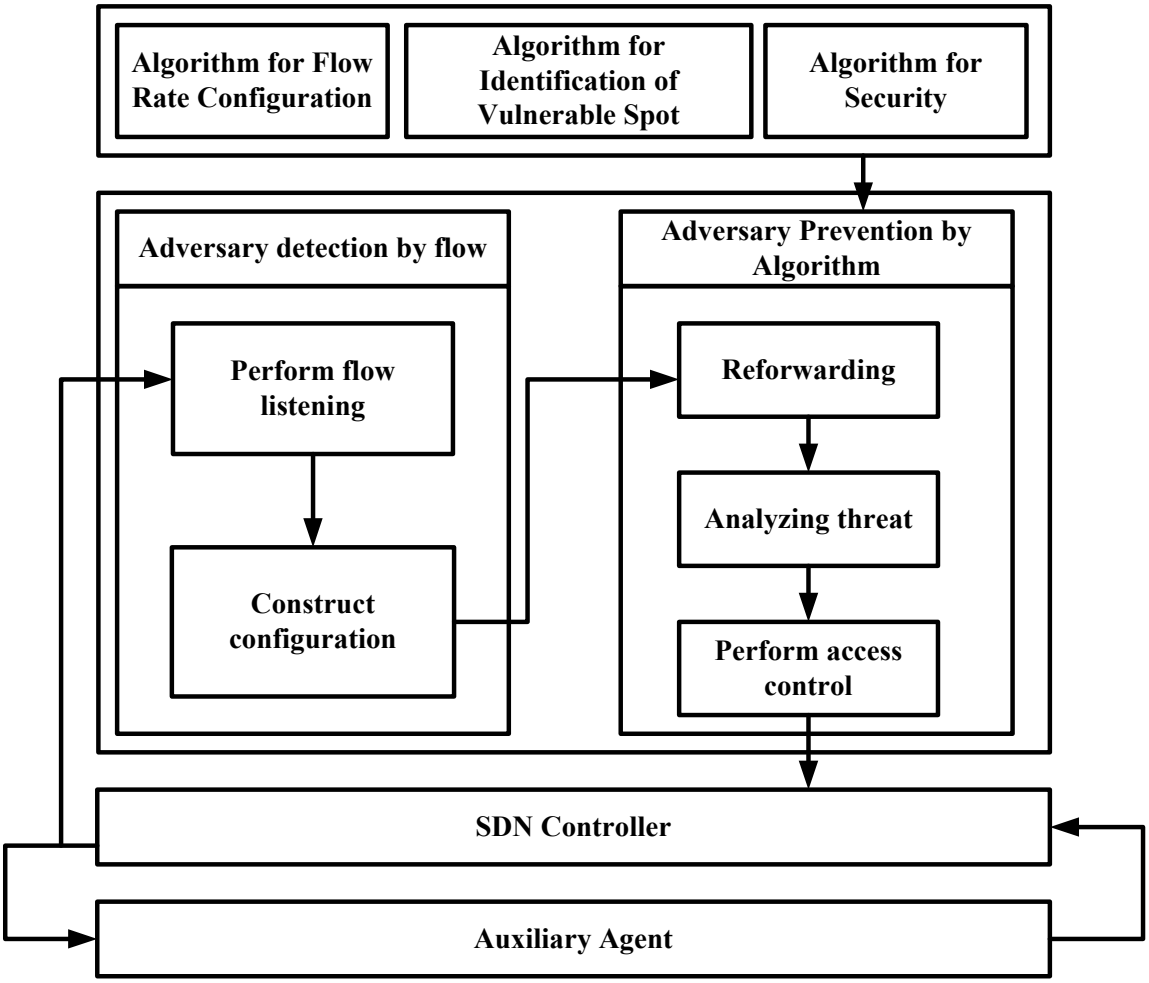

Figure 1 Proposed Architecture for Securing SDN

Figure 1 highlights the proposed architecture, which states uses an auxiliary agent in the data plane forwarding the secured routing request from switches towards the control plane. This request is initially listened to, followed by constructing an explicit configuration. The configuration information is further used in the application plane for performing the decision of reforwarding, followed by the analysis of the degree of severity of the presence of threats from incoming flow information. Finally, access control is designed, which isolates the vulnerable traffic flow in the IoT environment. The complete modelling is designed using two sets of operations where the first set performs detection of adversary flow, and the second set performs the prevention of an algorithm using three discrete algorithms. The discussion of algorithms follows in the next section.

5. Algorithm Implementation

This part of the paper discusses the algorithm that has been implemented in the proposed system in order to offer a secure connection towards the dynamic access induced by the software-defined network when deployed over an IoT. The idea is mainly to resist the dissemination of malicious users over the traffic flow. The complete idea of the proposed system is mechanized with the help of three sequential algorithms, as discussed below:

5.1 Algorithm for Flow Rate Configuration

The first algorithm is mainly responsible for the configuration of the flow rate, which takes the input of f (flow of message) and T (time slots) that, after processing, yields an outcome of α (rate of switch request) and β (similarity coefficient of the switch). The proposed system utilizes the rate of request for the switch to evaluate the degree of vulnerability present in the burst over data plane. This is the first operation where the configuration is carried out towards evaluating traffic flow features before detecting an attacker. The steps of the algorithm are as follows:

Algorithm for Flow Rate Configuration

Input: f (flow of message), T (time slots)

Output: α (rate of switch request), β (similarity coefficient of the switch)

Start

1. **For** i=1: f_{max}
2. **For** j=1: T
3. $\alpha(t) = \sum \frac{\phi(\tau)}{\sigma}$
4. $\beta(t) = \sum \frac{\sigma}{\phi(\rho)} . \pi$
5. **End**
6. $[\alpha, \beta]_t = \gamma[h, (1-h)]$
7. **End**

End

The operation taking place in the algorithm mentioned above is as follow:

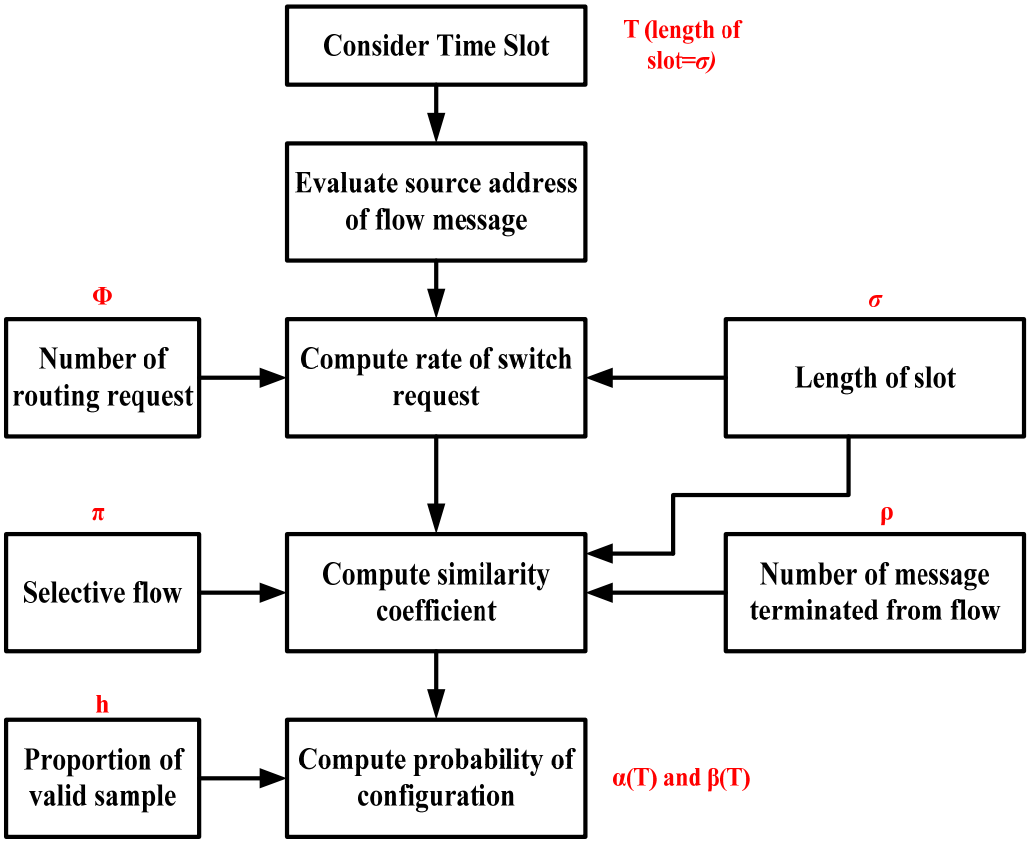

Figure 2 Process flow of Flow Rate Configuration

The proposed system uses this configuration while attempting to catch hold of an attacker's presence in the traffic flow. For this purpose, a time slot T is used with a uniform length of σ. It also represents the timeout value during the idle period. The algorithm considers that for the given maximum flow message f_{max} (Line-1), it considers all the time slots T (Line-2). The algorithm considers a variable τ, which retains all the information associated with the data packets forwarded from the i^{th} switch to order to compute the rate of switch request α for a specific instance of t timeslot as shown in Line-3. The variable Φ represents the *number of* requests of routing from the i^{th} switch. The next part of the algorithm implementation is to compute the similarity coefficient of the switch β (Line-4). In this phase of operation, the algorithm evaluates the number of data packets in the flow, the address of source node, the duration of the flow, and purpose fields associated with the variable ρ that represents the eliminated packet from the flow. The i^{th} switch receives the messages in order to compute this similarity coefficient, as shown in Line-4. In this computation, there is a variable π, which is computed by dividing the number of similar packets with the difference of duration of flow and number of messages. The proposed algorithm considers a proportion of the data distributed from the matrix α and β for the i^{th} switch, represented by a variable h (Line-6). The proposed system implements a function γ that uses a statistical measure of the real value of the data distribution in such a way that the probability of the value will always be within the limit of the value defined by h variable. Therefore, the probability statistical value of α is equivalent to γ(h), and that of β is equivalent to γ(1-h) as shown in Line-6. Therefore, the outcome of the algorithm is a configured value of α(T) and β(T), which is a direct representation of the highest permissible rate of request as well as the lowest permissible similarity coefficient associated with the flow of the ith switch.

5.2 Algorithm for Identification of Vulnerable Spot

This algorithm is responsible for performing the identification of the vulnerable spot in the nodes, where the term *spot* will represent the ingress port of the victim node. The idea of this algorithm is to identify the position of the ports of the vulnerable node in SDN deployed in the IoT ecosystem. The algorithm takes similar input as the prior algorithm and gives the output in the form of ψ (position of vulnerable spot). The steps of the algorithm are as follows:

Algorithm for Identification of Vulnerable Spot

Input: f (flow of message), T (time slots)

Output: ψ (position of vulnerable spot)

Start

1. **For** i=1: f_{max}
2. **For** j=1: T
3. Compute α(t)
4. **If** $\alpha_i(t) > \alpha_i(T)$
5. compute β(t)
6. **If** $\beta_i(t) > \beta_i(T)$
7. compute $\beta_{i,k}(t)$
8. Else
9. Vulnerable spot, ψ= ψ+1
10. **End**
11. **End**
12. **End**
13. **End**

End

For all the maximum flow (Line-1) and all timeslot (Line-2), the algorithm performs the computation of the rate of ith switch request using the prior algorithm (Line-3). Once this is done, the next step will be to find out of the obtained instantaneous rate $\alpha_i(t)$ is greater than the cumulatively configured rate $\alpha_i(T)$ (Line-4). In that case, the algorithm carries out the next step of computation of the similarity score β(t) (Line-5). This operation is carried out to check the vulnerable condition of the i^{th} switch due to the incoming flow of traffic. All the associated information of the flow can be extracted from the matrix, which retains statistical information about the ith switch directly from the controller. However, the computation of the similarity score is slightly different from the mechanism carried out in Line-4 in the prior algorithm. In this case (Line-5), the algorithm uses a similar length of timeslot σ, number of packets that entered in the i^{th} switch, and it also considers the cumulative value of the number of similar packets divided by duration it has come over specific observation flow. The proposed system compared the similarity coefficient $\beta_i(t)$ of the i^{th} switch for instantaneous time t with the cumulative value of similarity score $\beta_i(T)$. Suppose the conditional statement shown in Line-6 is found to be valid that it will represent

the possibility of vulnerable traffic flow with a higher possibility of the adversary in it. Hence, the similarity score is evaluated, as shown in Line-7, considering all the incoming flow information present in i^{th} switch for ingress port *k*. However, the confirmation of the attacked ingress port is determined otherwise (Line-8), and the position of port ψ is confirmed.

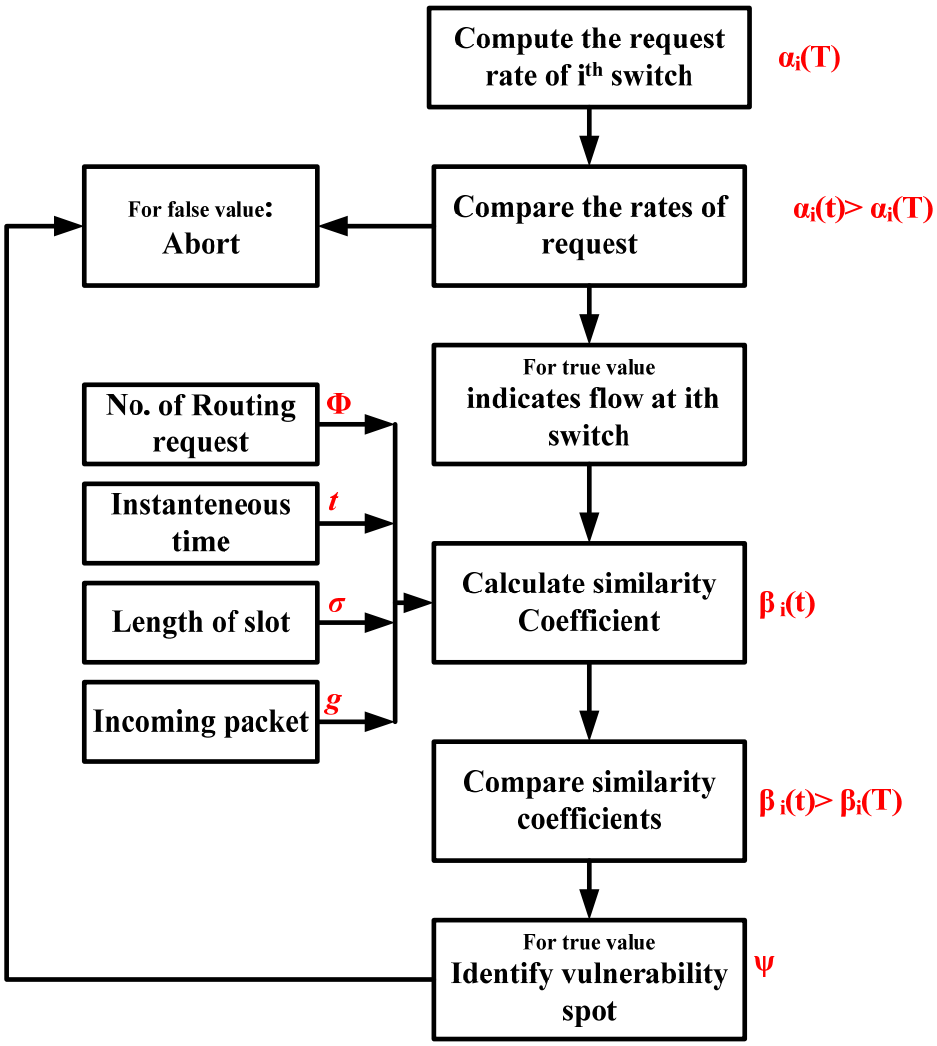

Figure 3 Process Flow of Identification of Vulnerable Spot

5.3 Algorithm for Security

This is the continuation of the prior mitigation algorithm, where the idea is to perform isolation of attack over SDN. For this purpose, the proposed system considers the presence of an auxiliary node that receives the information from the prior algorithm. This algorithm finally helps identify the degree of threat and create a matrix of legitimate links and illegitimate links information. This link information is used for allowing or resisting the malicious traffic over the SDN and thereby protect the complete network from the intrusion.

Algorithm for Security

Input: f (flow of message)

Output: η_1 (legitimate link matrix), η_2 (illegitimate link matrix)

Start

1. **For** i=1: f_{max}
2. **If** $g_1(k_1)=g_2(k_2)$
3. check $c[g_1(k_1), g_2(k_2)]$
4. **If** c<Th threshold
5. $g_1(k_1)$ is a part of η_1
6. **Else**
7. $g_1(k_1)$ is a part of η_2

8. **End**

9. **End**

End

The algorithm needs to have enough capability to identify the malicious flow dynamically. It is also not feasible to allocate the flow information of incoming traffic associated with each adversary flow to address the vulnerable spot. Therefore, the proposed system initially forwards the vulnerable flow from the vulnerable spot towards the auxiliary node, a software agent meant for processing the information for formulating the decision to be given to the controller. Based on the evaluation, the auxiliary node allocates the information of the flow over the switch, which is detected to be vulnerable to resist flows of the adversary. After obtaining the reforwarding command, the packets with a very low similarity coefficient are forwarded to the auxiliary node. This operation results in the halt of malicious packets on vulnerable spots associated with control and data planes in SDN architecture. It is to be noted that the inclusion of the auxiliary node offers an extensive advantage as they are more potential towards addressing problems of the illegitimate data packet. The auxiliary node is not programmed to update its information about attacks to the SDN controller, and therefore, this algorithm contributes towards updating the controller.

The algorithm considers all the flow of traffic (Line-1) as an input while after processing, it generates an outcome of η_1 (legitimate link matrix), η_2 (illegitimate link matrix). The algorithm initially checks if the number of incoming information in the flow $g_1(k_1)$ has connected outgoing information about the $g_2(k_2)$ (Line-2). In such a case, the algorithm checks if the number of data counters *c* and performs the comparison. If the counter value c is found to be less than a certain threshold Th than the study confirms that incoming information of the flow $g_1(k_1)$ is part of legitimate link matrix η_1 (Line-5), or else it is considered as illegitimate link matrix η_2 (Line-7). Fig.4 shows the flow of the proposed algorithm.

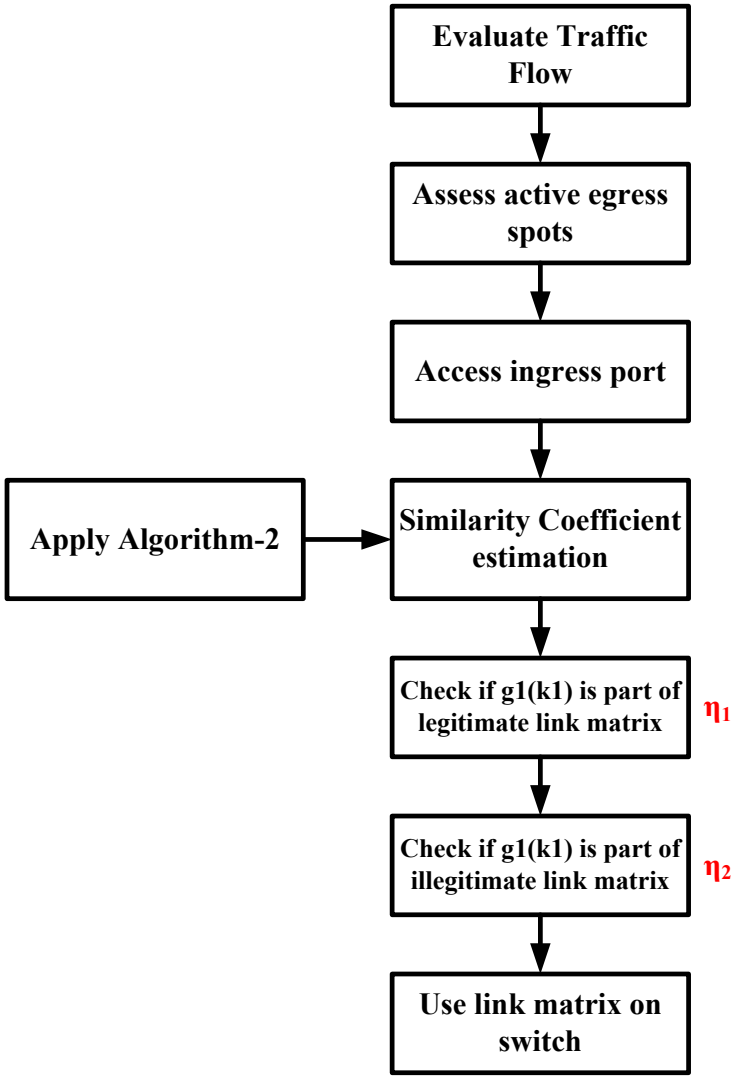

Figure 4 Process Flow of Proposed Security

6. Result Analysis

This section discusses the simulated outcome of the proposed study. Scripted in MATLAB, the proposed system assesses multiple networking environments in order to obtain this outcome. The analysis is carried out by comparing the conventional IoT based environment and the proposed SDN based IoT environment. However, for an effective benchmarking, the study outcome is compared with standard OpenFlow based security protocol [30] and recent work carried out by Yin et al. [31] towards resisting denial of service attack termed as DaD in this analysis. The work carried out by Nobakht et al. [30] has developed a policy-based flow monitoring system where OpenFlow is used for security. The approach presented by Yin et al. [31] has used a concept of where different components of SDN and IoT are linked together using gateway while a similar-based approach (Cosine) is applied for assessing the inbound packet flow for assessing vulnerability. Both the work has the best match with the research goal of the proposed system and hence chosen for comparative analysis.

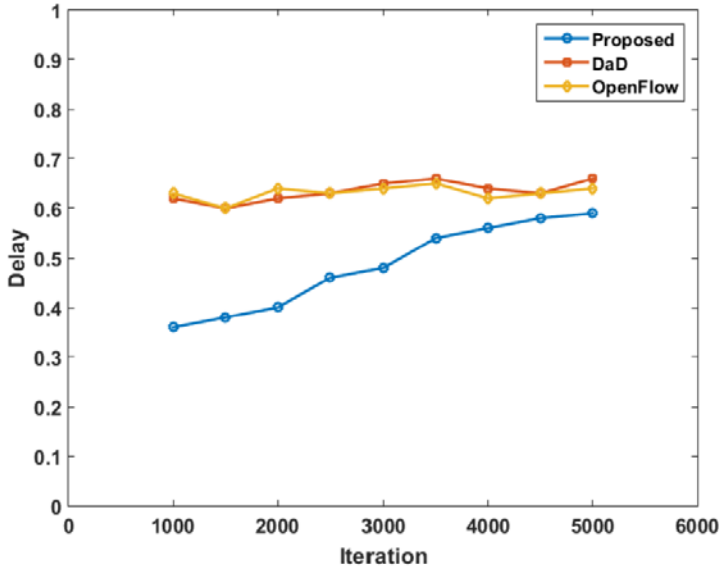

Figure 5 Comparative Analysis of Delay

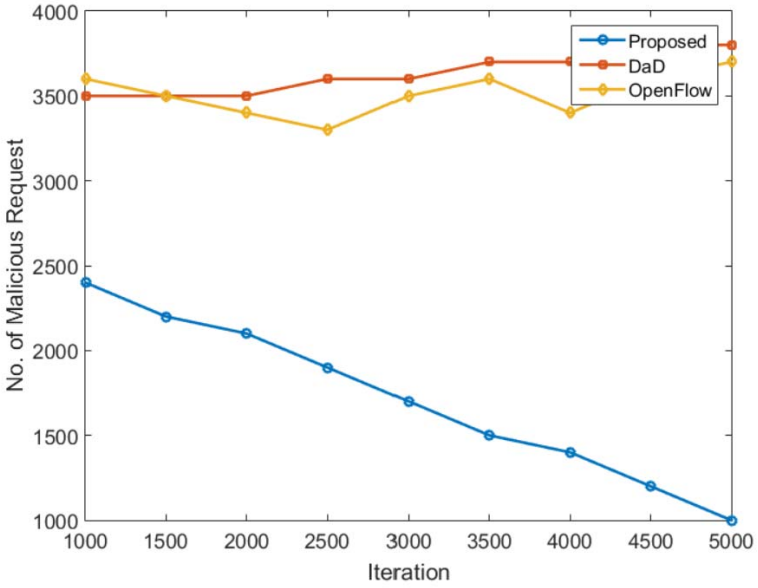

Figure 6 Comparative Analysis of No. of Malicious Node

The outcome is shown in Figure 5, and Figure 6 highlights that the proposed system offers better delay performance as well as effective in reducing the number of malicious requests, respectively, for increasing iteration as compared to the existing approach. This is because the OpenFlow-based security approach [30] has been using policy rules for access, which is highly time and resource-consuming for the increasing number of ports. Apart from this, the rules are not updated at an equal interval, which results in missing the identification of malicious nodes. Hence, it offers poor delay and lesser capability to reduce the request of the malicious node. The approach developed by Yin et al. [31] has overcome this problem by creating a controller pool which connects multiple IoT domains with the cloud concerning its switches. Hence, they demand lesser time to perform this communication; however, the information obtained from switches is assessed only on the local level, and these updates are only passed on to different domains when anyone requests. Hence, they have a reduced rate of detecting malicious nodes if the adversary migrates to a different domain. However, the proposed system has an integrated gateway system directly connected to switches causing a significant reduction in delay. Further, the information from the switch request is assessed rigorously concerning its flow rate followed by computation of multiple parameters, e.g., the proportion of valid samples, number of messages terminated from flow, similarity coefficient, and all this information are consistently updated among switches leading to higher detection rate. This contributes to lowering the number of malicious requests.

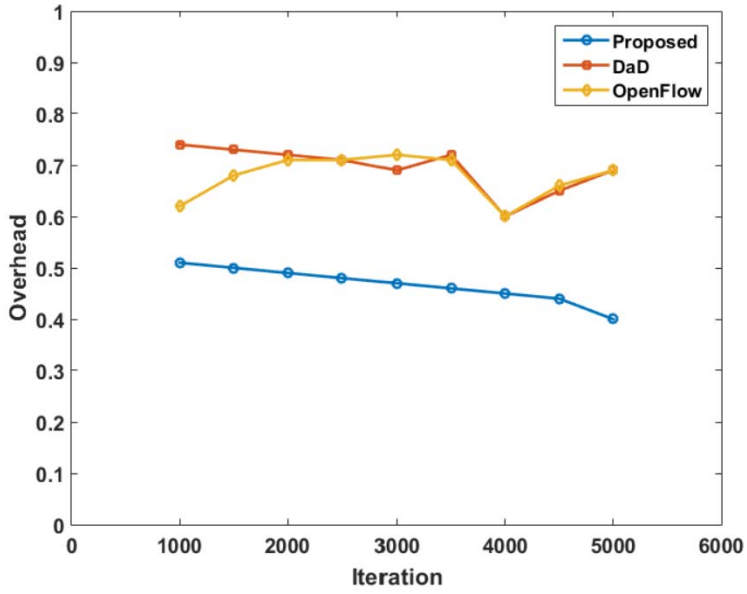

Figure 7 Comparative Analysis of Overhead

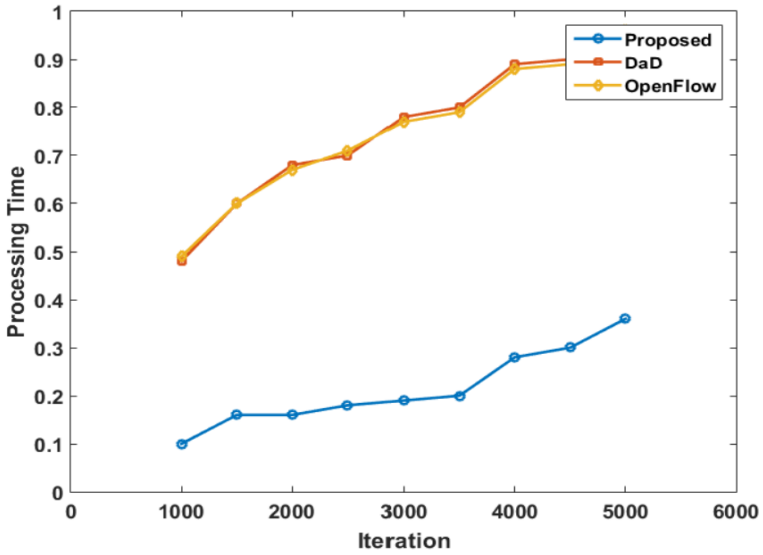

Figure 8 Comparative Analysis of Processing Time

Figure 7 and Figure 8 highlights the comparative analysis of overhead and processing time, where a similar trend of outcomes are observed for both proposed and existing system (OpenFlow, DaD). The lesser overhead of the proposed system is because all the flow packets are accountable in the proposed switching system, and there is no scope of any surplus message to bypass the switch. This phenomenon is not present in the existing system. The lower processing time of the proposed system is because the complete process of detection and mitigation is highly progressive, and there is no inclusion of any form of dependencies of iterative operation. Apart from this, the complete process of attack mitigation is carried out in more straightforward steps using simplified conditional logic. This contributes towards reducing algorithm processing time with increased iteration also, which is not found in the existing system. Hence, the proposed system exhibits cost-efficient security behavior using simplified and novel SDN architecture deployed on the IoT ecosystem.

7. Conclusion

Security is always a concern for the SDN based architecture, and despite various research attempts, it has been found that conventional SDN architecture is not ready to be deployed over the large-scale distributed network like IoT. Therefore, the proposed system introduces a novel framework for offering resistance to attack, which is dynamic and represents in the form of flow. Developed using three sequential algorithms, the proposed system has a detection module and a mitigation module. The detection module, after computing the degree of severity of attack from the request message generated from a switch, is forwarded to the auxiliary node, which lies in the data plane, and it is responsible for updating the controller about all vulnerable events. Hence without any loss of resources and exhaustion of operation in any planes of SDN, the proposed system offers a good balance between communication performance and secure communication in SDN when deployed in an IoT environment.

References

[1] Mahmood, Kashif, Ameen Chilwan, Olav Østerbø, and Michael Jarschel. "Modelling of OpenFlow-based software-defined networks: the multiple node case." IET Networks 4, no. 5 (2015): 278-284.
[2] Ghosh, Uttam, Pushpita Chatterjee, and Sachin Shetty. "A security framework for SDN-enabled smart power grids." In 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 113-118. IEEE, 2017.
[3] Al-Rubaye, Saba, Ekhlas Kadhum, Qiang Ni, and Alagan Anpalagan. "Industrial internet of things driven by SDN platform for smart grid resiliency." IEEE Internet of Things Journal 6, no. 1 (2017): 267-277.
[4] K. Raghunath and P. Krishnan, "Towards A Secure SDN Architecture," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018, pp. 1-7, DOI: 10.1109/ICCCNT.2018.8494043.
[5] Gonzalez, Carlos, Olivier Flauzac, Florent Nolot, and Antonio Jara. "A novel distributed SDN-secured architecture for the IoT." In 2016 International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 244-249. IEEE, 2016.
[6] Flauzac, Olivier, Carlos Gonzalez, and Florent Nolot. "Original secure architecture for IoT based on SDN." In 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS), pp. 1-6. IEEE, 2015.
[7] Ding, Ke, Xiulei Wang, Guomin Zhang, Zhen Wang, and Ming Chen. "A flow-based authentication handover mechanism for multi-domain sdn mobility environment." China Communications 14, no. 9 (2017): 127-143.
[8] Giotis, Kostas, Christos Argyropoulos, Georgios Androulidakis, Dimitrios Kalogeras, and Vasilis Maglaris. "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments." Computer Networks 62 (2014): 122-136.
[9] Jantila, Saksit, and Kornchawal Chaipah. "A security analysis of a hybrid mechanism to defend DDoS attacks in SDN." Procedia Computer Science 86 (2016): 437-440.
[10] Ravindra, S. "MAPSDN-EESC: A Modeling of Authentication Process for the Software Defined Network using Encrypted Entity Scheme Cryptography", Indian Journal of Computer Science and Engineering (IJCSE), Vol. 11, No. 4, pp. 394-404, 2020
[11] Dai, Weiqi, Pengfei Wan, Weizhong Qiang, Laurence T. Yang, Deqing Zou, Hai Jin, Shouhuai Xu, and Zirong Huang. "Tnguard: Securing iot oriented tenant networks based on sdn." IEEE Internet of Things Journal 5, no. 3 (2018): 1411-1423.
[12] Eng, You Hong, Kwong Meng Teo, Mandar Chitre, and Kien Ming Ng. "Online system identification of an autonomous underwater vehicle via in-field experiments." IEEE Journal of Oceanic Engineering 41, no. 1 (2015): 5-17.
[13] Gao, Jianbin, Kwame Opuni-Boachie Obour Agyekum, Emmanuel Boateng Sifah, Kingsley Nketia Acheampong, Qi Xia, Xiaojiang Du, Mohsen Guizani, and Hu Xia. "A blockchain-SDN-enabled Internet of vehicles environment for fog computing and 5G networks." IEEE Internet of Things Journal 7, no. 5 (2019): 4278-4291.
[14] Guo, Xuancheng, Hui Lin, Zhiyang Li, and Min Peng. "Deep Reinforcement Learning based QoS-aware Secure Routing for SDN-IoT." IEEE Internet of Things Journal (2019).
[15] Han, Zhuobing, Xiaohong Li, Keman Huang, and Zhiyong Feng. "A software defined network-based security assessment framework for cloudIoT." IEEE Internet of Things Journal 5, no. 3 (2018): 1424-1434.
[16] Hermosilla, Ana, Alejandro Molina Zarca, Jorge Bernal Bernabe, Jordi Ortiz, and Antonio Skarmeta. "Security orchestration and enforcement in NFV/SDN-aware UAV deployments." IEEE Access 8 (2020): 131779-131795.
[17] Li, Cheng, Zhengrui Qin, Ed Novak, and Qun Li. "Securing SDN infrastructure of IoT–fog networks from MitM attacks." IEEE Internet of Things Journal 4, no. 5 (2017): 1156-1164.
[18] Lin, Yi-Bing, Tse-Jui Huang, and Shi-Chun Tsai. "Enhancing 5G/IoT Transport Security Through Content Permutation." IEEE Access 7 (2019): 94293-94299.
[19] Liu, Yanbing, Yao Kuang, Yunpeng Xiao, and Guangxia Xu. "SDN-based data transfer security for Internet of Things." IEEE Internet of Things Journal 5, no. 1 (2017): 257-268.
[20] Medhane, Darshan Vishwasrao, Arun Kumar Sangaiah, M. Shamim Hossain, Ghulam Muhammad, and Jin Wang. "Blockchain-enabled Distributed Security Framework for Next Generation IoT: An Edge-Cloud and Software Defined Network Integrated Approach." IEEE Internet of Things Journal (2020).
[21] Meng, Yunfei, Zhiqiu Huang, Guohua Shen, and Changbo Ke. "SDN-Based Security Enforcement Framework for Data Sharing Systems of Smart Healthcare." IEEE Transactions on Network and Service Management 17, no. 1 (2019): 308-318.
[22] Ravi, Nagarathna, and S. Mercy Shalinie. "Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture." IEEE Internet of Things Journal 7, no. 4 (2020): 3559-3570.

[23] Shafi, Qaisar, Abdul Basit, Saad Qaisar, Abigail Koay, and Ian Welch. "Fog-assisted SDN controlled framework for enduring anomaly detection in an IoT network." IEEE Access 6 (2018): 73713-73723.
[24] Shirali-Shahreza, Sajad, and Yashar Ganjali. "Protecting home user devices with an SDN-based firewall." IEEE Transactions on Consumer Electronics 64, no. 1 (2018): 92-100.
[25] Sood, Keshav, Kallol Krishna Karmakar, Shui Yu, Vijay Varadharajan, Shiva Raj Pokhrel, and Yong Xiang. "Alleviating Heterogeneity in SDN-IoT Networks to Maintain QoS and Enhance Security." IEEE Internet of Things Journal (2019).
[26] Yazdinejad, Abbas, Reza M. Parizi, Ali Dehghantanha, Qi Zhang, and Kim-Kwang Raymond Choo. "An energy-efficient SDN controller architecture for IoT networks with blockchain-based security." IEEE Transactions on Services Computing (2020).
[27] Yuan, Bin, Chen Lin, Huan Zhao, Deqing Zou, Laurence Tianruo Yang, Hai Jin, and Chunming Rong. "Secure Data Transportation with Software-defined Networking and kn Secret Sharing for High-confidence IoT Services." IEEE Internet of Things Journal (2020).
[28] Zarca, Alejandro Molina, Jorge Bernal Bernabe, Ruben Trapero, Diego Rivera, Jesus Villalobos, Antonio Skarmeta, Stefano Bianchi, Anastasios Zafeiropoulos, and Panagiotis Gouvas. "Security management architecture for NFV/SDN-aware IoT systems." IEEE Internet of Things Journal 6, no. 5 (2019): 8005-8020.
[29] Zarca, Alejandro Molina, Jorge Bernal Bernabe, Antonio Skarmeta, and Jose M. Alcaraz Calero. "Virtual iot honeynets to mitigate cyberattacks in sdn/nfv-enabled iot networks." IEEE Journal on Selected Areas in Communications 38, no. 6 (2020): 1262-1277.
[30] Nobakht, Mahdi, Craig Russell, Wen Hu, and Aruna Seneviratne. "IoT-NetSec: policy-based IoT network security using OpenFlow." In 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 955-960. IEEE, 2019.
[31] Yin, Da, Lianming Zhang, and Kun Yang. "A DDoS attack detection and mitigation with software-defined Internet of Things framework." IEEE Access 6 (2018): 24694-24705.