A Novel Trust-based Framework for Multi-Plane Security in Software Defined Networking (SDN)

Gaurav V. Sharmaa, Jagdish W. Bakalb
Pillai HOC College of Engineering and Technology, Navi Mumbai, India
Pillai HOC College of Engineering and Technology, Navi Mumbai, India

Abstract:
Software Defined Networking (SDN) is a new architectural concept in networking. It adds programmability, resulting in greater flexibility in network management. But it also raises a lot of security concerns. At different planes, SDN employs a variety of network elements, including applications, controllers, and switches, as well as a variety of communication techniques, including APIs and protocols like OpenFlow. Having a single security framework that applies to the entire SDN infrastructure is therefore difficult. Many security measures, including trust-based security frameworks, have been proposed; however, they are only applicable to either the North-Bound Interface (NBI) or the South-Bound Interface (SBI) and not to the entire SDN architecture. The proposed framework assigns a common trust score based on the behavior of the applications (NBI) and hosts (SBI). Each network entity's trust scores indicate how trustworthy it is. Additionally, the framework mitigates the attacks by blocking the network entities based on trust scores when they fall below a set threshold value. The framework gives the network administrator the option to select the attack detection method, as well as the option of setting the threshold and penalty levels to block entities in the environment. The framework's flexibility will make it easier for the security team to detect low-trust network entities, set priorities, and conduct further research on the threat. This framework can be applied to traditional networks as well as Hybrid-SDN network deployments, in addition to working with SDN.

Keywords: Software Defined Networking, SDN, Security, Trust, Trust Model Trust-based Framework, SDN Monitoring.
DOI: 10.24297/j.cims.2023-1-10

1. Introduction

In a Software Defined Network (SDN), the Control Plane and the Data Plane are separated, enhancing network management flexibility [1]. This separation enables a single control plane controller to manage many data-forwarding devices or switches. SDN Controllers at the control plane can be programmed since they are essentially servers to which applications may be deployed. Applications and the controller can communicate with one another through Application Programming Interfaces (APIs) such as RESTful, Java, Python, etc. APIs [2].

In SDN, the control plane handles a variety of network-related tasks. The controller supplies the forwarding rules (flows) to the Data Plane devices and provides data, statistics, and network management interfaces to the applications.

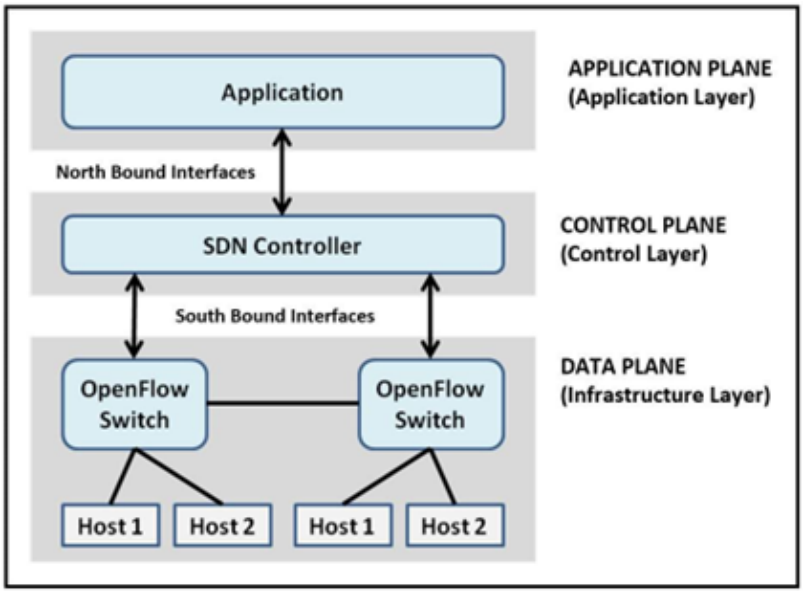

Fig. 1: SDN Architecture

The data plane performs the network's forwarding function. The controller provides the flow rules to the switch via the South-Bound Interface, such as the OpenFlow protocol, which is stored by the switch in its flow table and used for making forwarding decisions. If there are no rules for a particular flow, it asks the controller for that flow's rules.

North-Bound Interface in the application plane of the controller is used to integrate applications with the network. Python, Java, and RESTful APIs are used to facilitate the application and controller to communicate with one another.

2. Security Challenges And Threats To Software Defined Networks (Sdn)

SDN security threat vectors have been identified by *Kreutz et al.* [3]. The numerous abuse or attack routes in SDN have been classified by *Yoon et al.* in [4]. The research at various SDN Layers/Interfaces is also thoroughly compared by *Scott-Hayward et al.* in [5]. A contemporary layered or interface taxonomy of the documented security vulnerabilities, threats, and challenges of the SDN is developed by *Akhunzada et al.* [6] to highlight the key categories of vulnerabilities of each SDN layer or interface. *Li et al.* [7] have focussed on the OpenFlow-based

SDN and have categorized its security issues and solutions. *Yurekten et al.* [8] have proposed the threat category for SDN based on threat intelligence. The study claims that SDN is vulnerable to the same threats as traditional networks. Due to differences in architectural design, SDN may need to employ different or modified strategies to counter such threats discussed in [9].

In conclusion, SDN faces security challenges that the traditional networks does alog with many new ones. SDN controller may act as a single point of failure and a prime target for attacks. Switches at the Data Plane can also be the target of attacks. The lack of application authentication and authorization, conflicts between applications regarding flow rules, and chances for malicious applications to insert fraudulent traffic are just a few of the main issues with the SDN controllers [22].

3. Related Work

In recent years, a significant number of researchers have presented security solutions for SDN using a variety of techniques.

Chowdhary et al. [10] proposed a dynamic game-based security framework for SDN. *Banse et al.* [11] proposed a taxonomy-based policy framework. *Niemiec et al.* [12] have proposed a Risk Assessment and Management approach to SEcure SDN (RAMSES) in which the SDN controller uses a risk-aware methodology to review the reputations of external applications.

Many machine learning-based approaches have been presented to detect and deal with intrusions and attacks in SDN [13],[14],[15].

Giotis et al. [16] and *Zaalouk et al.* [17] combined the SDN protocol, OpenFlow, with existing protocols like NetFlow/IPFIX and sFlow. *Yu et al.* [18] proposed FlowSense, a push-based method to obtain flow data from the switches. SDN-Mon is a framework for SDN monitoring proposed by *Phan et al.* [19] that separates the monitoring logic and the forwarding logic. A multi-level distributed monitoring and remediation framework for SDN named Tennison is introduced by *Fawcett et al.* [20]. *Tsai et al.* [21] have presented an overview of SDN monitoring techniques and have compared it with traditional network monitoring techniques.

Yao et al. [22], proposed the Trust Management Framework (TMF) wherein an application's trust value is evaluated and dynamically updated using Trusted Platform Module (TPM). "Trust Trident," a framework for trust-based authentication, has been presented by *Duy et al.* [23]

which intercepts requests from applications for authentication and authorization. A distributed trust management framework named "TRUFL" has been proposed by *Chowdhary et al.* [24]. A security policy analysis approach has been proposed by *Pisharody et al.* [25] to assess the vulnerability of flow rules. For the secured information flow between the SDN Controller and the network applications, Aliyu et al. [26] have proposed a trust management system. The Trust-Oriented Controller Proxy (ToCP) was proposed by *Betgé-Brezetz et al.* in [27] which uses several redundant controllers and polling mechanism for trust. *Banse et al.* [28] provided the controller with a web-based, secure, independent northbound interface.

Isong et al. [29] proposed a trust model that restricts access to network resources and communication with the controller to applications with a minimum level of trust. *Burikova et al.* [30], *Cui et al.* [31] and *Scott-Hayward et al.* [32] have proposed a framework for authorizing and authenticating the applications. FRESCO is a framework proposed by *Shin et al.* [33] which imposes flow restrictions and defend the network from threats using security enforcement kernel [33], [34].

The limitations of SDN trust are discussed in [35]. The Policy-based Security Architecture (PbSA) that dynamically manages the network is analyzed by *Sood et al. in* [36]. Based on direct, indirect, and historical trust values, *Zhao et al.* [37] have proposed a trust evaluation method for the Data Plane nodes.

4. Proposed Framework

The proposed security framework is a modular one that enhances SDN's security by using a trust-based methodology. The goal of a trust-based framework is to create a security solution that can operate on multiple planes. It generates a trust score for the various network entities and offers a standard mechanism to arrive at the right conclusions under the set policies.

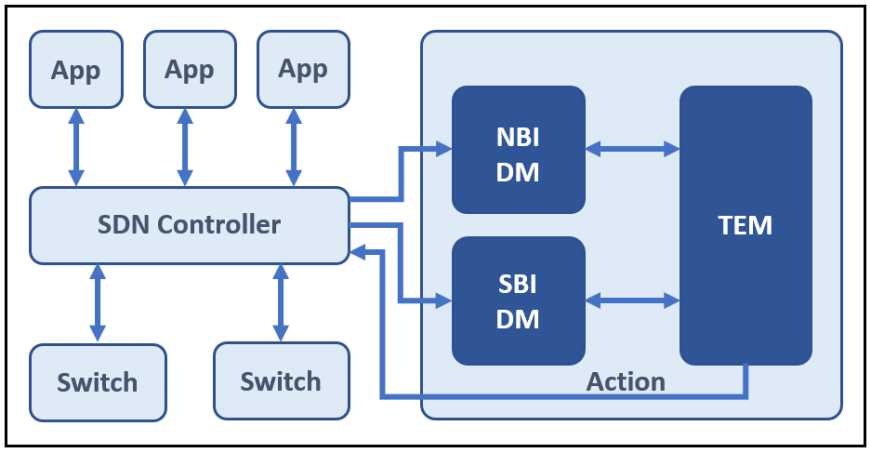

Fig. 2: Proposed Trust-based Security Framework

Architecture of Proposed Framework

The proposed security framework is a modular framework that uses a trust-based approach to add security to SDN. It can easily be integrated with the SDN architecture and enables the computation of trust scores for various network entities, such as hosts and applications, that are deployed in different planes of the SDN architecture. It provides a common mechanism, in the form of a trust score, to make proper decisions as per the established policies, to detect issues or attacks, restrict network entities showing abnormal or malicious behavior from using the network and resolve the issues.

The high-level architecture of the proposed security framework is shown in Fig. 2. The framework is composed of the NB-DM, SBI-DM, and TEM modules discussed below.

North-Bound Interface – Detection Module (NBI-DM)

The Northbound Interface (NBI) in (SDN) is a communication interface between the SDN control layer and the application layer. It allows network applications to interact with the SDN controller.

A malicious or vulnerable SDN application deployed also poses a significant risk to the controller and the data it contains because controllers lack the tools necessary to independently evaluate the eligibility, legality, or reliability of the applications.

One of the major issues with the SDN controllers includes the inability of application authentication and authorization which may also lead to conflicts between applications over the

flow rules, and the potential for malicious applications to input fraudulent traffic. The North-Bound Interface – Detection Module (NBI-DM) addresses these concerns.

NBI-DM is used to identify issues with or attacks on the software applications that interact with the SDN Controller. This is done by authenticating every application whenever the application makes an API request to the controller so that any unknown or new application installed with the approval of the administrator or blocked by the administrator, is stopped from making the request. Further, whether an application is authorized, or permitted, to use a specific API to make a request is also checked against the permissions set by the administrator.

NBI-DM checks for this behavior of each application and notifies the TEM in a binary output – normal behavior or abnormal behavior. Based on this, TEM further computes and manages the trust scores for each application and can take appropriate action, if required. The design and the process of NBI-DM are further discussed in section 5.1.

South-Bound Interface – Detection Module (SBI-DM)

The Southbound Interface (SBI) in software-defined networking (SDN) is a communication interface between the SDN control layer and the data plane. The SBI provides a way for the SDN controller to interact with the network elements, such as switches and routers, and manage the flow of data within the network.

In a software-defined network (SDN), the flow table is a data structure stored in the forwarding elements, such as switches and routers, that contains information about the flow of data within the network. The flow table is used to determine the path that packets should take based on the source and destination addresses, and other fields in the packet header. The flow table is managed by the SDN controller and is programmed using the Southbound Interface (SBI), which allows the controller to enforce network policies and manage the flow of data. The flow table contains information about the actions to be taken for each flow, such as forwarding the packet to a specific port, dropping the packet, or modifying certain fields in the packet header.

Flow table monitoring is performed by the SDN controller to get real-time visibility into the flow of data in the network, making it possible to monitor network behavior and implement network policies more effectively. This data, in the form of statistics, can be used to gain insights into network behavior. It contains information about the flow of data in the network, including the

source and destination addresses, packet header fields, and the actions to be taken for each flow. Thus, flow table monitoring can be used to monitor network performance, troubleshoot network issues, implement network policies, and even detect security threats like malware infections, DDoS attacks, and other forms of malicious activity and behavior.

South-Bound Interface – Detection Module (SBI-DM) can identify network attacks by analyzing the flow table entries collected by the SDN Controller from OpenFlow Switches. Due to the modular design of the framework, SBI-DM can detect attacks using any existing method based on machine learning [38], [39], entropy [40], network traffic analysis [41], or any other technique. SBI-DM analyzes the flow table entries of all the switches and inspects the data for unusual behavior, which could be due to an anomaly or a potential attack by a malicious or compromised network entity like a host.

SBI-DM checks for this behavior of each host connected to the network and notifies the TEM in a binary output – normal behavior or abnormal behavior. Based on this, TEM further computes and manages the trust scores for each host and can take appropriate action, if required.

The design and the process of NBI-DM are further discussed in section 5.2.

Trust Evaluation Module (TEM)

SDN architecture makes use of multiple and diverse components, protocols, and technologies at different planes. To overcome this issue and provide a common mechanism for getting a trust score, NBI-DM and SBI-DM are the two essential modules. Both of them provide a binary output, regarding the entities at their plane, whether an abnormal behavior (possibly an attack) is detected or not. This is used as input by the Trust Evaluation Module (TEM) to use a common method to compute and update the trust score for different entities and manage them.

TEM takes appropriate actions such as penalizing the network entity (reducing its trust score) for abnormal behavior or blocking the entity for which the trust score is reduced below the set threshold at the SDN Controller and alerting the administrator.

TEM incentivizes each application for every permitted API request, by incrementing the trust score for displaying normal behavior. For every attempt to make an API request which is not

permitted, the application is penalized by reducing the trust score by a larger value. Whether the behavior of the application is normal or not is detected and reported to TEM by NBI-DM.

Hence, in case NBI-DM detects an anomalous behavior by the application which could also be a potential case of a compromised application making the request, in either case, the behavior displayed by the application is not normal and the trust score is reduced by TEM.

In case of an occasional erroneously made bad request, the application's trust score will suffer but it will get chances to improve its trust score over the period by performing normally. However, if the abnormal behavior of the application continues then in a few time cycles its trust score will go below the set threshold value set by the administrator and the application will be blocked from the network.

TEM also incentivizes each network host for displaying normal behavior by incrementing its trust score. For every instance of abnormal behavior, the host is penalized by reducing its trust score by a larger value. Whether the behavior of the application is normal or not is detected and reported to TEM by SBI-DM.

Hence, in case SBI-DM detects an occasional or erroneous display of abnormal behavior by the host, then its trust score will suffer but it will get chances to improve its trust score over the period by performing normally. However, if the abnormal behavior of the host continues then in a few time cycles its trust score will go below the set threshold value set by the administrator and the host will be blocked from the network by automatically disabling the port on the switch to which it is connected.

5. Implementation Design Of The Proposed Framework

The proposed framework is modular, hence, the different modules can be implemented independently. The framework gives the flexibility to select the attack detection techniques making it flexible to be used in different environments, and it can be a policy decision based on the requirements.

North-Bound Interface – Detection Module (NBI-DM)

The NBI-DM checks for APIs requests from applications (App) that are permitted to make API requests. The API request is denied if the application is not registered or if it is registered but

blacklisted or blocked by the administrator. The API Checker then verifies that the administrator has granted permission for the specified API request. Each application may use any of the permitted APIs to request the SDN Controller, and API Checker keeps a database of all those APIs.

NBI-DM notifies TEM of whether API requests made by the applications are accepted or discarded to update the trust score of the applications linked to the SDN Controller, which is covered in the discussion of TEM later.

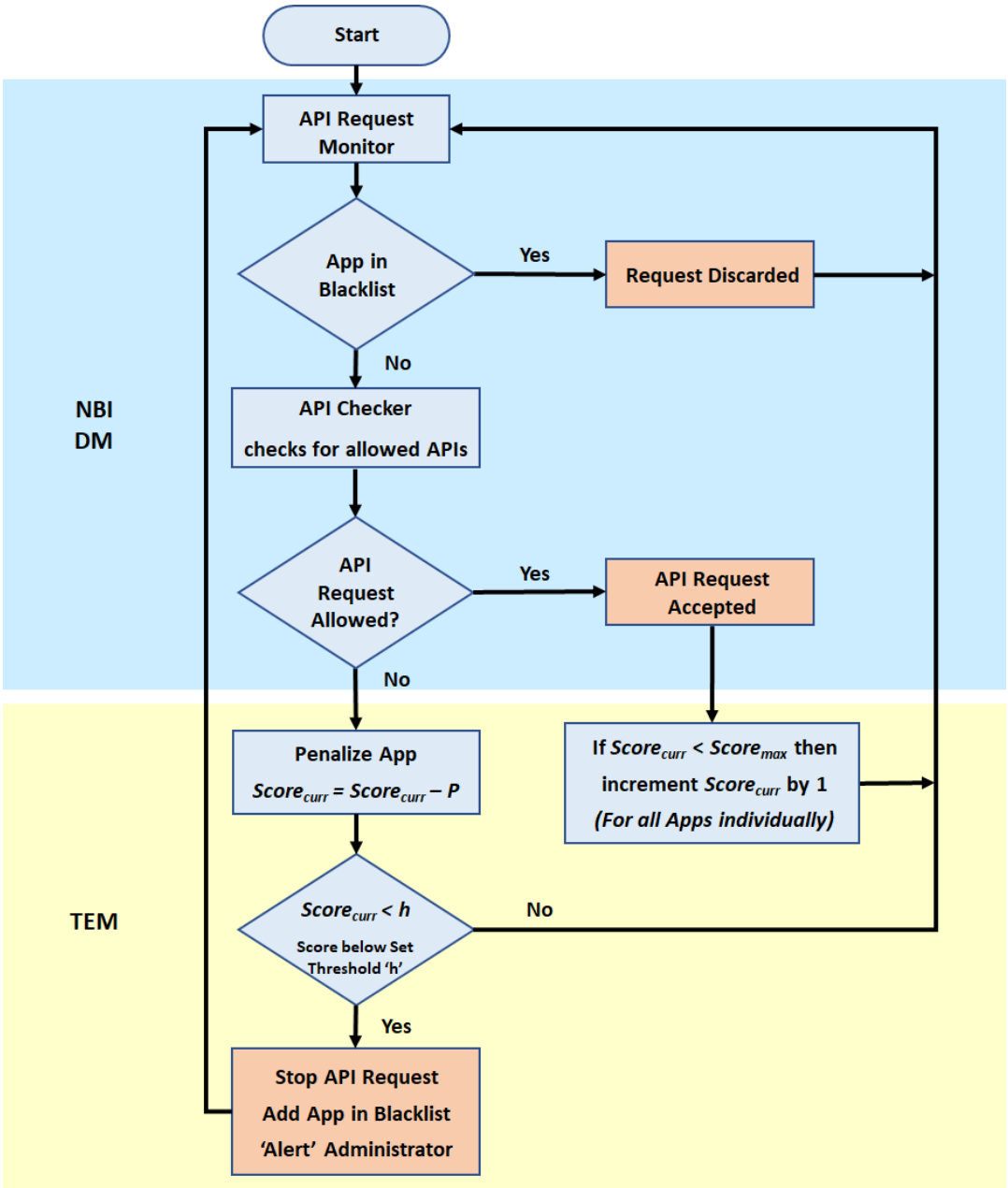

Fig. 3: Working of NBI-DM with TEM

South-Bound Interface – Detection Module (SBI-DM)

SBI-DM analyzes the flow entries collected by the SDN Controller from OpenFlow Switches to detect network attacks. The modular design of the framework enables SBI-DM to detect attacks using any existing method based on Machine Learning based [38], [39], Entropy-based [40], network traffic analysis [41], or any other technique, to detect the attack.

Irrespective of whether an attack is detected or not, SBI-DM notifies the output to TEM to update the trust score of the hosts connected to the network.

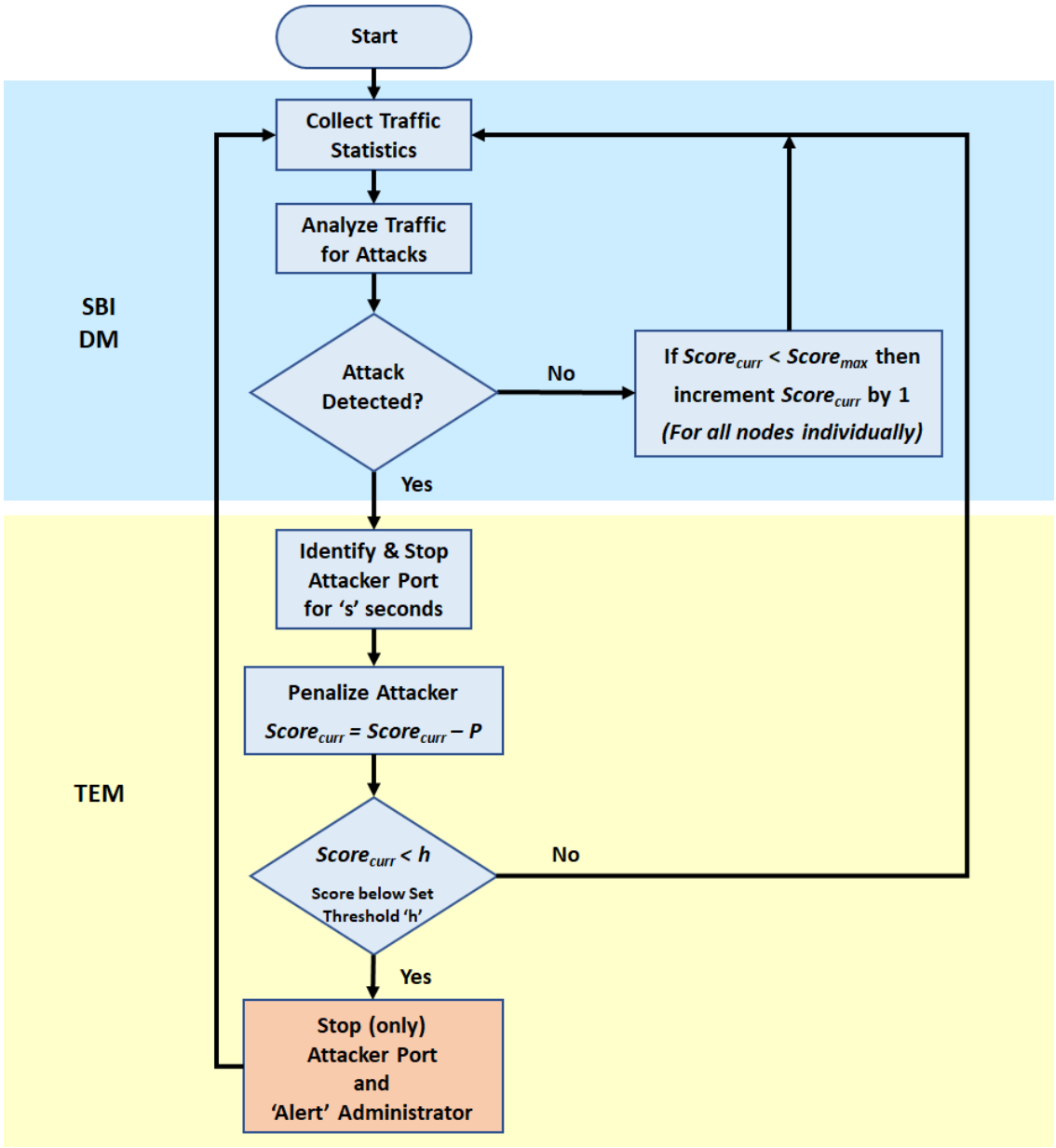

Fig. 4: Working of SBI-DM with TEM

Trust Evaluation Module (TEM)

TEM is the most important module of the proposed framework. This is the module that ensures that different entities in the SDN get a common Trust Score. TEM detects network attacks by analyzing trust scores from the first 2 modules - NBI-DM and SBI-DM.

Trust Computation for Attack Detection by NBI-DM

The trust score for every application is computed based on its behavior. The two possible cases of behavior displayed by the applications are discussed below.

Case 1: Application making an API request which is permitted:

If the API request is permitted then the request is accepted by the SDN Controller.

And, if,

$$Score_{curr} < Score_{max}$$

Then,

$$Score_{curr} = Score_{curr} + 1$$

where $Score_{max}$ is the maximum score that can be attained.

The increase in the current trust score ($Score_{curr}$) increases the trustworthiness of the App for behaving as permitted.

Case 2: Application making an API request which is NOT permitted:

This could be a case when the application (App) erroneously, due to compromise or an attack behaves abnormally.

If the API request is not permitted then the request is discarded.

And, the App is penalized by decrementing the current trust score by 'P' (penalty) set by the Administrator.

$$Score_{curr} = Score_{curr} - P$$

If the current trust score of the App goes below the set threshold 'h',

$$Score_{curr} < h,$$

the App will not be allowed to make any further API requests till the Administrator resets the $Score_{curr}$ above the Threshold 'h'

(ideally to $Score_{max}$ after taking appropriate actions against the threat).

Trust Computation for Attack Detection by SBI-DM

The trust score for every network host is computed based on its behavior. The two possible cases of behavior displayed by the applications are discussed below.

Case 1: No attack is detected:

If the host is behaving on the network normally and is not involved in an attack,

And, if,

$$Score_{curr} < Score_{max}$$

Then,

$$Score_{curr} = Score_{curr} + 1$$

where $Score_{max}$ is the maximum score that can be attained.

The increase in the current trust score ($Score_{curr}$) increases the trustworthiness of the network host for behaving as permitted.

Case 2: A network attack is detected:

This could be a case when the network host erroneously, due to compromise or an attack behaves abnormally.

In this case, the host is penalized by decrementing the current trust score by '*P*' (penalty) set by the Administrator.

$$Score_{curr} = Score_{curr} - P$$

If the current trust score of the host goes below the set threshold '*h*',

$$Score_{curr} < h,$$

the host will not be blocked at the switch and will not be allowed to participate over the network till the Administrator resets the $Score_{curr}$ above the Threshold '*h*'

(ideally to $Score_{max}$ after taking appropriate actions against the threat).

Summary

TEM computes the trust score for applications as well as network hosts by analyzing their behavior with the help of NBI-DM and SBI-DM.

The scoring system and TEM ensure that the network entity is not immediately blocked when abnormal activity is detected.

The trust-based technique makes it possible to determine over several cycles (depending on the penalty and threshold value) whether the attack is genuine or just a minor, random anomalous behavior on the part of the entity.

6. Conclusion And Future Scope

The proposed trust-based security framework is a novel framework that identifies the trustworthiness of all the entities in an SDN network and multiple planes. The framework provides a common trust score to Apps (NBI) and Hosts (SBI). The trust scores for each network entity give its trustworthiness which can be high if the network entity behaves normally, but can reduce in case of an attack or a display of anomalous behavior. It can also mitigate the attacks by blocking the entities with trust scores below a threshold value.

The framework allows the network administrator to select the attack detection technique and set the penalty and the threshold values, to block the entities, for the environment. More attacks and security issues can be detected using the framework by using different attack detection techniques.

The flexible nature of the framework will help the security team to identify the low-trust network entities, and further investigate the threat.

This framework can work with SDN and can be extended to traditional networks and Hybrid-SDN network deployments.

Supplementary Information: The patent for the proposed framework is filed and approved by the Patent Office of India on the title "A Trust-based Attack Mitigation Framework (TAMF) framework for Software Defined Networks", Patent Application No: 202221065221, Date of Filing: 14/11/2022.

References

1. Open Networking Foundation (ONF), "SDN Architecture 1.0 Overview", Published: November 2014, Accessed: November 2022, Available at: https://opennetworking.org/wp-content/uploads/2013/02/SDN-architecture-overview-1.0.pdf

2. "Using HTTP Methods for RESTful Services", Accessed: November 2022, Available at: http://www.restapitutorial.com/lessons/httpmethods.html
3. Diego Kreutz, Fernando M.V. Ramos, Paulo Verissimo, "Towards Secure and Dependable Software-defined Networks", Proceedings of the second ACM SIGCOMM workshop on Hot Topics in Software Defined Networking (HotSDN' 13), Pages 55–60, August 2013.
4. Changhoon Yoon, Seungsoo Lee, Heedo Kang, Taejune Park, Seungwon Shin, Vinod Yegneswaran, Phillip Porras, Guofei Gu, "Flow Wars: Systemizing the Attack Surface and Defenses in Software-Defined Networks", IEEE/ACM Transactions on Networking, Volume: 25, Issue: 6, Page(s): 3514 – 3530, December. 2017
5. Sandra Scott-Hayward, Sriram Natarajan, and Sakir Sezer, "A Survey of Security in Software Defined Networks", IEEE Communication Surveys and Tutorials, Volume: 18, Issue: 1, First Quarter 2016.
6. Adnan Akhunzada, Abdullah Gani, Nor Badrul Anuar, Ahmed Abdelaziz, Muhammad Khurram Khan, Amir Hayat, Samee U. Khan, "Secure and Dependable Software Defined Networks", Elsevier, Journal of Network and Computer Applications, Volume 61, Pages 199-221, February 2016.
7. Wenjuan Li, Weizhi Meng, Lam For Kwok, "A Survey on OpenFlow-based Software Defined Networks: Security Challenges and Countermeasures", Elsevier, Journal of Network and Computer Applications, Volume 68, Pages 126-139, June 2016.
8. O. Yurekten and M. Demirci, "SDN-based Cyber Defense: A Survey", Elsevier Journal: Future Generation Computer Systems, vol. 115, pp. 126-149, Feb. 2021
9. J. C. C. Chica, J. C. Imbachi, and J. F. Botero, "Security in SDN: a comprehensive survey," Journal of Network and Computer Applications, vol. 159, p. 102595, 2020.
10. Ankur Chowdhary, Sandeep Pisharody, Adel Saeed Alshamrani, Dijiang Huang, "Dynamic Game based Security framework in SDN-enabled Cloud Networking Environments", Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization (SDN-NFVSec '17), Pages 53–58, March 2017, Scottsdale, USA.
11. Christian Banse and Julian Schuette, "A Taxonomy-based Approach for Security in Software-Defined Networking", Proceedings of IEEE International Conference on Communications (ICC), May 2017, Paris, France.
12. Marcin Niemiec, Piotr Jaglarz, Marcin Jekot, Piotr Chołda, and Piotr Boryło, "Risk Assessment Approach to Secure Northbound Interface of SDN Networks", Proceedings

of the International Conference on Computing, Networking and Communications (ICNC), April 2019, Honolulu, USA.

13. Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, Anca Delia Jurcut, "Machine-Learning Techniques for Detecting Attacks in SDN", Proceeding of IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), October 2019, Dalian, China.
14. Nasrin Sultana, Naveen Chilamkurti, Wei Peng, Rabei Alhadad, "Survey on SDN based Network Intrusion Detection System using Machine Learning Approaches", Peer-to-Peer Networking and Applications, Springer, Volume 12, Issue 2, Pages 493–501, March 2019.
15. Narmeen Zakaria Bawany, Jawwad A. Shamsi, Khaled Salah, "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions", Arabian Journal for Science and Engineering, Springer, Pages 425–44, February 2017.
16. K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," Computer Networks, vol. 62, pp. 122–136, 2014.
17. A. Zaalouk, R. Khondoker, R. Marx and K. Bayarou, "OrchSec: An orchestrator-based architecture for enhancing network-security using Network Monitoring and SDN Control functions," 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, Poland, 2014, pp. 1-9, doi: 10.1109/NOMS.2014.6838409.
18. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "Flowsense: Monitoring network utilization with zero measurement cost," in International Conference on Passive and Active Network Measurement. Springer, 2013, pp. 31–41.
19. X. T. Phan and K. Fukuda, "SDN-Mon: Fine-grained traffic monitoring framework in software-defined networks," Journal of Information Processing, vol. 25, pp. 182– 190, 2017.
20. L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, "Tennison: a distributed SDN framework for scalable network security," IEEE Journal on Sel. Areas in Commun., vol. 36, no. 12, pp. 2805–2818, 2018.
21. P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, and C.-S. Yang, "Network Monitoring in Software-Defined Networking: A Review," IEEE Systems Journal, 2018.
22. Zhen Yao, Zheng Yan, "A Trust Management Framework for Software-defined Network Applications", Special Issue Paper, Concurrency and Computation: Practice and Experience, 2018; e4518, Published online in Wiley Online Library, March 2018.

23. Phan The Duy, Do Thi Thu Hien, Nguyen Van Vuong, Nguyen Ngoc Hai A, and Van-Hau Pham, "Toward a Trust-Based Authentication Framework of Northbound Interface in Software Defined Networking", International Conference on Industrial Networks and Intelligent Systems (INISCOM 2019). Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 293. pp 269-282, Springer, Cham, August 2019.
24. Ankur Chowdhary, Dijiang Huang, Adel Alshamrani, Myong Kang, Anya Kim, Alexander Velazquez, "TRUFL: Distributed Trust Management Framework in SDN", Proceedings of IEEE International Conference on Communications (ICC), July 2019, Shanghai, China.
25. Sandeep Pisharody, Janakarajan Natarajan, Ankur Chowdhary, Abdullah Alshalan, Dijiang Huang, "Brew: A Security Policy Analysis Framework for Distributed SDN-Based Cloud Environments", IEEE Transactions on Dependable and Secure Computing, July 2017.
26. Aliyu Lawal Aliyu, Peter Bull and Ali Abdallah, "A Trust Management Framework for Network Applications within an SDN Environment", 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), March 2017, Taipei, Taiwan.
27. Stéphane Betgé-Brezetz, Guy-Bertrand Kamga, Monsef Tazi, "Trust Support for SDN Controllers and Virtualized Network Applications", Proceedings of the 1st IEEE Conference on Network Softwarization (NetSoft), April 2015, London, UK.
28. Christian Banse and Sathyanarayanan Rangarajan, "A Secure Northbound Interface for SDN Applications", Proceedings of IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), August 2015, Helsinki, Finland.
29. Bassey Isong, Tebogo Kgogo, Francis Lugayizi and Bennett Kankuzi, "Trust Establishment Framework between SDN Controller and Applications", 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), June 2017, Kanazawa, Japan.
30. Svetlana Burikova, JooYoung Lee, Rasheed Hussain, Iuliia Sharafitdinova, Roman Dzheriev, Fatima Hussain, Salah Sharieh, Alexander Ferworn, "A Trust Management Framework for Software Defined Networks-based Internet of Things", IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), October 2019, Vancouver, Canada.
31. Hongyan Cui, Zunming Chen, Longfei Yu, Kun Xie, Zongguo Xia, "Authentication Mechanism for Network Applications in SDN Environments", Proceedings of the 20th International Symposium on Wireless Personal Multimedia Communications

(WPMC2017), December 2017, Bali, Indonesia.

32. Sandra Scott-Hayward, Christopher Kane and Sakir Sezer, "OperationCheckpoint: SDN Application Control", IEEE 22nd International Conference on Network Protocols, October 2014, Raleigh, USA.
33. Seugwon Shin, Phillip Porras, Vinod Yegneswaran, Martin Fong, Guofei Gu, and Mabry Tyson, "FRESCO: Modular Composable Security Services for Software- Defined Networks," Published in Internet Society (ISOC) 20th Annual Network and Distributed System Security Symposium (NDSS), February 2013.
34. Phillip Porras, Seungwon Shin, Vinod Yegneswaran, Martin Fong, Mabry Tyson, Guofei Gu, "A Security Enforcement Kernel for OpenFlow Networks", Proceedings of the first workshop on Hot Topics in Software Defined Networks (HotSDN '12), Pages 121–126, August 2012.
35. Christopher C. Lamb, Gregory L. Heileman, "Towards Robust Trust in Software Defined Networks", Proceedings of IEEE Globecom Workshops (GC Wkshps), December 2014, Austin, USA.
36. Keshav Sood, Kallol Krishna Karmakar, Vijay Varadharajan, Uday Tupakula, and Shui Yu, "Analysis of Policy-Based Security Management System in Software- Defined Networks",
37. Zhao B, Liu Y, Li X, Li J, Zou J, "TrustBlock: an adaptive trust evaluation of SDN network nodes based on double-layer blockchain", PloS one 15(3) (2020). https://doi.org/10.1371/journal.pone.0228844
38. R. Santos, D. Souza, W. Santo, A. Ribeiro, E. Moreno, "Machine learning algorithms to detect DDoS attacks in SDN", Concurrency and Computation: Practice and Experience., vol. 32, no. 16, pp. 1-14, August. 2020.
39. Jin Ye, Xiangyang Cheng, Jian Zhu, Luting Feng, Ling Song, "A DDoS Attack Detection Method Based on SVM in Software Defined Network", Security and Communication Networks, vol. 2018, Article ID 9804061, pp. 1-8, 2018.
40. Jiang, Y., Zhang, X., Zhou, Q., Cheng, Z., "An Entropy-Based DDoS Defense Mechanism in Software Defined Networks.", In: Chen, Q., Meng, W., Zhao, L. (eds) Communications and Networking. ChinaCom 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 209. Springer, Cham., pp. 169–178, 2018.
41. Anil, A., Rufzal, T.A., Adat Vasudevan, V., "DDoS Detection in Software-Defined Network Using Entropy Method", In: Giri, D., Raymond Choo, KK., Ponnusamy, S., Meng, W.,

Akleylek, S., Prasad Maity, S. (eds) Proceedings of the Seventh International Conference on Mathematics and Computing. Advances in Intelligent Systems and Computing, vol 1412. Springer, Singapore, 2022.