International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

Securing IoT Devices Through SDN-Based Access Control

Akash.D*[1], Abdul Rahman[1], and Dayakar[1],Mr.S.Praveen Kumar[2],Dr.J.Jayaprakash[2],Dr.G.Victo Sudha George[2]
Mail ID:- akashdara9087@gmail.com, Dept of Computer science and Engineering, *[1] Faculty

Dr. M.G.R Educational and Research Institute, Maduravoyal, Chennai-95, Tamil Nadu, India

ABSTRACT- The Software-Defined Networking (SDN) in a Linux environment is demonstrated in our approach. We set up a 40-node network using basic Tcl commands. The main objective is to allow users to select the beginning and finishing points of a dynamic path. This demonstrates how SDN can adjust to evolving network circumstances. This shortens wait times and speeds up data transfer, demonstrating that SDN may increase Linux efficiency. The central component of this new technology called "SDN-Controller" that is considered to bring about great changes in computer networking, is the software defined networking (SDN). Hence, SDN is a lucid means for IoT to attain high performance and to overcome the current problems.***Keywords—*** *IoT,SDN ,Ubuntu,Tcl.*

1. INTRODUCTION

The security of the Internet of Things (IoT) has drawn particular attention since these devices may be uniquely identified in authentication and identity management because these devices may be uniquely identified, are able to perform meaningful data analysis and decision-making, and can connect to the Internet. Indicative IoT devices are smart home sensors, medical gear, vehicles, aircrafts and even nuclear facilities. Due to the architecture of IoT, many connections can be established without any security or authentication, and leave them open for any kind of attacks.Regardless, the security mechanisms designed for other networks may not be applicable to IoT due to peculiarities of latter. Software-defined networking is a revolutionary technology that was brought about by the recent trends in computer networking. (SDN) which enables the central program, called 'the SDN Controller', to control hardware networking devices and thus routing and switching decisions. Hence, the security of the Internet of Things (IoT) is the one that causes most concerns. These gadgets, however, have some unique features. For instance, they can be uniquely identified; they are intelligent in their data analysis and decision-making; they are network capable such that they can be connected to the internet. Smart devices such as sensors of simple houses, automobiles, aircraft, medical equipment and even nuclear reactors are current Internet of Things (IoT) devices. IoT communication can become exposed without integrity and authentication and in this way it may seek for attacks. In fact, traditional network security measures use intrusion detection/prevention systems and firewalls located at the network perimeter to repel outside attacks. What's more, these techniques are not applied to IoT networks only due to its characteristics. Computer network innovation called software defined networking was brought by recent developments in software defined networking. In comparison to our system model where the network environment was not modifiable, we found that the IoT network had the same technology stack.Jararweh and associates. The proposed SDIoT modifies the IoT security and network administration processes into a centralized iOS stack.At the last, Liu et al. focused their attention on a software-defined based Internet of Things architecture but with emphasis on urban IoT sensors. Our approach and theirs fell in two different categories. First of all, our system models did not require software and hardware modification for either the storage or the network of IoT devices. According to Salman et al. proposes an identity management mechanism that assists to protect the data of IoT supported networks. The proposed structure in their paper is similar to our theoretical system frame.Jararweh along with others. We propose to use SDIoT, a IoT platform with its software-defined network architecture we use to storage and security management easier. In comparison with the two previous ideas, the network

and storage setup were similar when they were used to illustrate the models of our system. The identity authentication for the Internet of Things (IoT) using SDN was proposed by Salman et al. However, the identity addresses several communication protocols which are mapped together using the SDN controller via a common identity.Chakrabarti and associates. This paradigm put forward an awareness service which utilizes the SDN functionalities and management interfaces to centralize the IoT network management of IoT resource regions. Tran et al. suggested using algorithms to improve the effectiveness of controller placement in SDN Networks with the distributed IoT. The specific context is the SDN-enabled IoT networks which our approach and proposed algorithms handle their problems of the SDN Controller placement in distributed networks. Purposeful matinent of dynamic AAA (authentication, authorization, and accounting) for Internet of Things security function network using SDN/VNF cyber situational awareness security architecture was appointed by Zarca et al..SDTCP has allowed the creative changes of open gate sequencing of TCP ACK packets to lower the sending rate of bystander flowsSoftThings, a security framework for IoT devices using SDN for detection and neutralization of threats based on SDN has been presented by Zhang et al. Under the current system, any possible suspicious behavior in the future such as DoS attack is counted once a particular type of observed behavior on the Io We have designed the network model and execution with the active use of SDN that helps IoT devices (IoT) while operating HTTP and limit security attacks. from taking place even when the IoT devices have not been modified, and compare our approach with current system and other approaches that are available. Also, we use SDN and DPI (deep packet inspection) for applying traffic separation techniques. The IoT device was a Raspberry Pi, the media center software was Kodi Media Center, and the SDN model was the Openflow Protocol to create the suggested system. The proposed we use even if it is not necessary to modify the IoT devices make the threats ameliorate and ensures of the confidentiality and integrity. In this chapter we will follow this structure. In Section 2 we discuss the IoT and SDN concept and provide its composition and performance overview. Finally, section 3, describes the system architecture, the design process of system implementation, and the man-in-the-middle attack on IoT.

2. LITERATURE REVIEW Internet of Things (IoT) Security Analysis

In their 2011 paper, Gan, Lu, and Jiang delve into the security implications of the (IoT). They meticulously analyze emerging threats since 2012, shedding light on their potential impact, particularly emphasizing the lack of attention given to these threats in existing literature and media. The authors raise concerns about the possibility of IoT involvement in cyber warfare between major powers. They introduce a significant new threat termed "hijack," surpassing previous challenges and posing unique obstacles to traditional mitigation strategies. Proposing behavioral-economic models, they advocate for a nuanced approach to address these threats, stressing the urgency of tackling IoT security vulnerabilities.

Software-Defined Networking (SDN) and Network Security.

The advent of Software-Defined Networking (SDN) heralds a transformative era in networking architecture, as discussed in the SDN White Paper. This document outlines the fundamental shift brought about by SDN, highlighting its core principles of decoupling control and data planes, centralizing network intelligence, and abstracting network infrastructure. It underscores the industry- wide efforts led by the Open Networking Foundation (ONF) in standardizing crucial elements like the OpenFlow protocol.Further, Tariq, Riaz, and Rasheed (2014) showcase the practical application of SDN in network security through their implementation of a Layer 2 firewall. Their work demonstrates the flexibility and programmability of SDN in enforcing network security policies, leveraging a POX controller to control packet flow between hosts based on firewall rules. Michelle et al. (2013) extend this exploration by developing a firewall application over an OpenFlow-based SDN controller. Their study underscores the feasibility of implementing firewall functionalities in software without dedicated hardware, thus enhancing the adaptability of network security solutions.Additionally, Rolbin (2013) explores the potential of SDN in early detection of network threats. Through leveraging SDN with the OpenFlow protocol, Rolbin proposes a cost-efficient and flexible solution for identifying and preventing network vulnerabilities. This master's thesis

underscores the role of SDN in enhancing network security measures, laying the groundwork for proactive threat detection and mitigation strategies.

Integration of SDN and IoT for Network Management

As the Internet of Things (IoT) continues to proliferate, the integration of SDN and IoT emerges as a promising avenue for network management. Sood, Yu, and Xiang (2015) delve into this integration, exploring its potential to simplify wireless network controls while addressing scalability and security concerns in IoT applications. They highlight the synergies between SDN and IoT, envisioning a programmable wireless network architecture tailored to IoT requirements.Similarly, Zhijing et al. (2014) propose a software-defined approach for managing heterogeneous IoT networks. Their work aims to optimize network resource utilization by provisioning different classes of IoT traffic across various wireless communication solutions. This paper underscores the importance of efficient network management in accommodating the diverse needs of IoT applications.

Furthermore, Yaser et al. (2015) present a comprehensive framework for managing IoT networks based on software-defined systems. Integrating SDN, software-defined storage, and software-defined security, their framework offers a unified approach to IoT network management

3. EXISTING SYSTEM

Additionally, we used SDN in conjunction with deep packet inspection (DPI) to apply traffic separation techniquesIn our approach, the Cuckoo Search Optimization (CSO) based Energy and Delay Aware Routing Algorithm is used to resolve the variable delays, packet losses and ensuring scalability of SD-WBANs. SDN and IoT concepts are introduced in Sec.2. They will be analyzed in terms of their interactions and operations in this section. A system model which includes system implementation & man-in-the-middle attack are defined in the third sectionThe flexibility of SDN system even allows it to fit into current security frameworks, helping achieve conformity and boosting security status. It enables preventive mitigation of security threats through adaptive response to threats and continuous monitoring, making IoT devices and networks resistant to the cyber-attacks that are varied by nature. It seems that the current SDN system for IoT security gives network-wide control to the security function, advanced threat detection, and smooth integration with current security infrastructure; thus, it provides a comprehensive approach for improving the security of IoT setups.

4. PROPOSED SYSTEM

In execution of Installing TCL and related packages on Ubuntu creates the development environment, which is the initial step in running an SDN system for IoT security. Create an SDN controller with TCL by starting from scratch or combining it with pre- existing frameworks such as OpenDaylight. Create TCL scripts that implement security regulations and use features like as encryption and authentication to enable safe communication between IoT devices. Use TCL to interact with Internet of Things devices to set up security and communication protocols. Use TCL for automated response mechanisms, anomaly detection, and real-time network traffic monitoring. To enable comprehensive testing and validation, set up a test environment with virtualized IoT devices. Use TCL to record the design, implementation, and maintenance processes. Update and maintain the system often to guarantee dependability and adjust to new security risks. Although TCL is flexible and simple to script, think about the performance and scalability implications. For more complicated functionalities, you might want to add additional languages to TCL.

The algorithm used: Ubuntu Linux contains CSO optimization in the SDN architecture which includes routing, traffic engineering, load balancing and dynamic flow control algorithms. CSO stands as a central optimization process that sequentially refines routing paths from source to destinations. Path selection algorithms such as Dijkstra's and dynamic routing protocols choose the best paths. Traffic engineering algorithms will guarantee proper function of resources allocation and meet the QoS requirements. The load balancing algorithms smoothen the process of traffic by redirecting it in such a way that the system does not get overloaded. Dynamic flow control algorithms use adaptive routing strategies that tailor flow paths in real time based on the current network conditions. Through this combined function, these algorithms address two common issues of network optimization;efficiency, reliability, and responsiveness.

METHODOLOGY USED

The SDN system focused on IoT security in Ubuntu through TCL scripting involves various key

methodologies.Several important approaches are involved in the SDN system that uses TCL scripting to focus on IoT security in Ubuntu. First, particular security needs for IoT devices and the network architecture are analyzed. Next, design principles like flexibility, scalability, and modularity are applied. The SDN controller, the IoT device communication module, and the security enforcement module are examples of manageable parts of the system that are further separated into smaller units. Iterative development is essential because it integrates feedback and addresses changing security challenges, allowing for incremental architecture refinement. Prototyping is used to evaluate system performance and verify design choices. Best practices and security standards are adhered to at every stage of design and execution. An essential first step is documentation, which records architecture specifics, component interactions, security measures, and deployment considerations for later use.

6. SYSTEM ARCHITECTURE

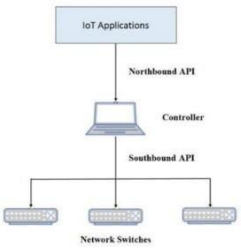

Fig:1 Architecture

The SDN architecture for IoT security as shown in figure 1 with Ubuntu using TCL scripting employs modular design principles, iterative development, and adherence to security standards. Components include the SDN controller, IoT device communication module, and security enforcement mechanisms. Prototyping and documentation ensure robustness and scalability of the system.

7. IMPLEMENTATION

The implementation of the SDN architecture for IoT security in Ubuntu using TCL scripting involves several key steps. First, the SDN controller is developed, utilizing TCL to handle network management and security policies. This controller communicates with IoT devices through TCL scripts that establish secure communication channels using encryption algorithms like AES or RSA.Next, the IoT device communication module is implemented, enabling seamless interaction between the SDN controller and IoT devices. TCL scripts handle message passing and protocol translation, ensuring compatibility with various IoT communication protocols such as MQTT or CoAP.Security enforcement mechanisms are integrated into the architecture using TCL scripts, which dynamically adjust security policies based on real-time network conditions and threat intelligence. Anomaly detection algorithms, implemented in TCL, monitor network traffic for suspicious behavior, triggering automated responses to mitigate potential security threats.Throughout the implementation process, iterative development and testing are employed to validate system functionality, identify potential vulnerabilities, and optimize performance. Prototyping is used to simulate real-world scenarios and the effectiveness of the security measures need to be evaluated.Finally, comprehensive documentation is created to outline the architecture, including component interactions, security mechanisms, and deployment considerations.

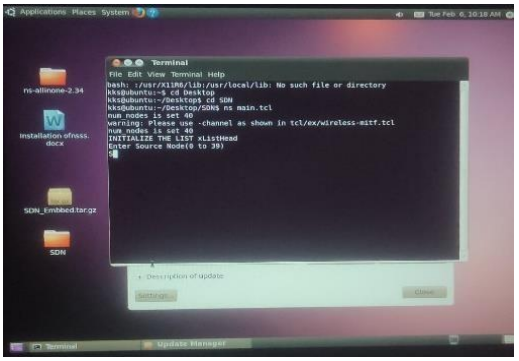

Fig:2 New User Login

Figure 2 describes the front end New user logindetails .

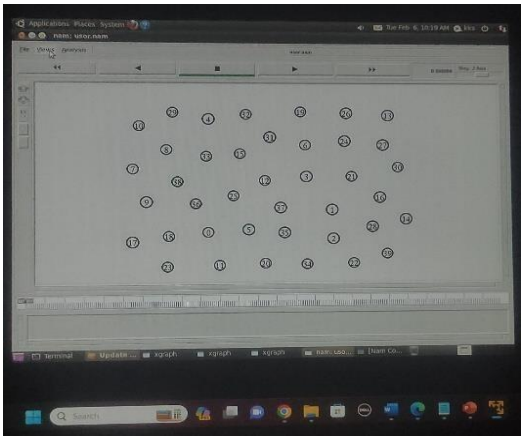

Fig : 3 SDN Portal

Figure 3 describes the SND portal where the processing of finding the path between the sourceand destination nodes

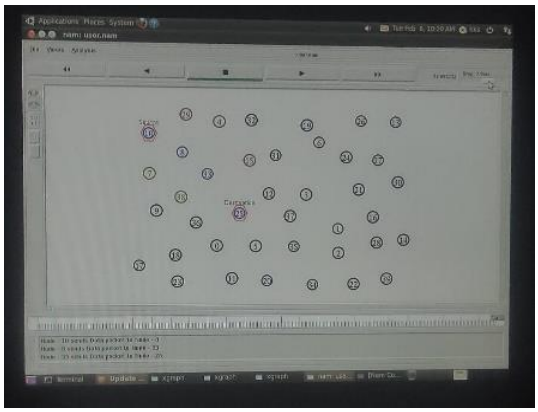

Fig : 4 Destination

Figure 4 shows the final output where the source node and destination node is found and the path isclearly dipicted

8. MODULES CONCLUSION

we successfully implemented Software-Defined Networking (SDN) within a Linux environment using Tcl commands, constructing a network with 40 nodes. The dynamic establishment of paths from user-specified sources to destinations showcased the adaptability and effectiveness of SDN. By leveraging open-source tools and Linux as the underlying operating system, our approach prioritizes accessibility and flexibility in deploying SDN solutions. This study contributes valuable insights into the practical application of SDN, emphasizing real-time, user-driven network optimization in an open-source setting. These findings pave the way for further research, encouraging the seamless integration of SDN into mainstream networking practices.

REFERENCES

1. Gan, G.; Lu, Z.; Jiang, J. Internet of Things Security Analysis. In Proceedings of the 2011 International Conference on Internet Technology and Applications, Wuhan, China, 16–18 August 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1–4. [Google Scholar]
2. Software-Defined Networking: The New Norm for Networks. Available online: https://www.opennetworking.org /images/stories/downloads/sdn- resources/white-papers/wp-sdn-newnorm.pdf (accessed on 5 January 2020).
3. Tariq, J.; Riaz, T.; Rasheed, A. A Layer2 Firewall for Software Defined Network. In Proceedings of the 2014 Conference on Information Assurance and Cyber Security (CIACS), Rawalpindi, Pakistan, 12–13 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 39–42. [Google Scholar]
4. Michelle, S.; Park, S.H.; Lee, B.; Yang, S. Building Firewall over the Software- Defined Network Controller. In Proceedings of the 16th International Conference on Advanced Communication Technology, Chennal, India, 27–28 February 2013; IEEE: Piscataway, NJ, USA, 2014; pp. 744–748. [Google Scholar]
5. Pena, J.G.V.; Yu, W.E. Development of a Distributed Firewall Using Software Defined Networking Technology. In Proceedings of the 2014 4th IEEE International Conference on Information Science and Technology, Busab, Korea, 26 February–1 March 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 449–452. [Google Scholar]
6. Rolbin, M. Early Detection of Network Threats Using Software Defined Network (SDN) and Virtualization. Master's Thesis, Carleton University, Ottawa, OT, Canada, 2013. [Google Scholar]
7. Sood, K.; Yu, S.; Xiang, Y. Software- Defined Wireless Networking Opportunities and Challenges for Internet- of-Things: A review. *IEEE Int. Things J.* **2015**, *3*, 453–463. [Google Scholar][CrossRef]
8. Zhijing, Q.; Denker, G.; Giannelli, C.; Bellavista, P.; Venkatasubramanian, N. A Software Defined Networking Architecture for the Internet-of-Things. In Proceedings of the 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, Poland, 5–9 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–9. [Google Scholar]
9. Yaser, J.; Al-Ayyoub, M.; Benkhelifa, E.; Vouk, M.; Rindos, A. SDIoT: A software defined based internet of things framework. *J. Ambient. Intell. Humaniz. Comput.* **2015**, *6*, 453–461. [Google Scholar]
10. Liu, J.; Li, Y.; Chen, M.; Dong, W.; Jin, D. Software-defined internet of things for smart urban sensing. *IEEE Commun. Mag.* **2015**, *53*, 55–63. [Google Scholar] [CrossRef]
11. Salman, O.; Abdallah, S.; Elhajj, I.H.; Chehab, A.; Kayssi, A. Identity-Based

Authentication Scheme for the Internet of Things. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Wrocław, Poland, 7–9 September 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1109–1111. [Google Scholar]

12. Chakrabarty, S.; Engels, D.W.; Thathapudi, S. Black SDN for the Internet of Things. In Proceedings of the 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems, Dallas, TX, USA, 19–22 October 2015; IEEE: Piscataway, NJ, USA; pp. 190–198. [Google Scholar]
13. Theodorou, T.; Violettas, G.; Valsamas, P.; Petridou, S.; Mamatas, L. A Multi-Protocol Software-Defined Networking Solution for the Internet of Things. *IEEE Commun. Mag.* **2019**, *57*, 42–48. [Google Scholar] [CrossRef]
14. Tran, A.K.; Piran, M.; Pham, C. SDN Controller Placement in IoT Networks: An Optimized Submodularity-Based Approach. *Sensors* **2019**, *19*, 5474. [Google Scholar] [CrossRef] [Green Version]
15. Molina Zarca, A.; Garcia-Carrillo, D.; Bernal Bernabe, J.; Ortiz, J.; Marin-Perez, R.; Skarmeta, A. Enabling virtual AAA management in SDN-based IoT networks. *Sensors* **2019**, *19*, 295. [Google Scholar] [CrossRef] [PubMed] [GreenVersion]
16. Lu, Y.; Ling, Z.; Zhu, S.; Tang, L. SDTCP: Towards datacenter TCP congestion control with SDN for IoT applications. *Sensors* **2017**, *17*, 109. [Google Scholar] [CrossRef] [PubMed]
17. Zhang, A.; Lin, X. Security-Aware and Privacy-Preserving D2D Communications in 5G. *IEEE Netw.* **2017**, *31*, 70–77. [Google Scholar] [CrossRef]
18. Raspberry pi—Teach, Learn, and Make with Raspberry pi. Available online: https://www.raspberrypi.org/ (accessed on 5 January 2020).
19. Kodi j Open Source Home Theatre Software. Available online: http://kodi.tv/ (accessed on 5 January 2020).
20. Overview of the Internet of Things. Available online: https://www.itu.int/rec/T-REC-Y.2060–201206-I (accessed on 5 January2020).