JES
Journal of Engineering Sciences

Vol 10, Issue 9, Sept /2019
ISSN NO: 0377-9254

DYNAMIC AUTHENTICATION FOR SOFTWARE DEFINED NETWORKS

[1]Sadaf Khaleel, [2]Dr T. Ravi
[1]PG Scholar, MTech, Dept of CSE, Shadan Women's College of Engineering and Technology HYD, T.S.
khaleelsadaf@gmail.com
[2]Professor, Dept of CSE, Shadan Women's College of Engineering and Technology HYD, T.S.

***Abstract*—** Software Defined Networking (SDN) is being broadly executed by basic infrastructure systems, anyway providing security is still a challenge. This work represents SafeFlow an Automatic Trust Negotiation Protocol for SDN, a first line of resistance and fine-grained verification protocol, in order to deny the access of open flow switches without valid digital certifications. Traditional security approaches dependent on personality or capacities are not tackled by the issue of establishing trust between outsiders. One elective way to deal with shared trust foundation is Trust Negotiation, the bilateral trade of advanced credentials is to set up trust bit by bit. The proposed protocol portrays Trust Negotiation in OpenFlow protocol, an extension to the OpenFlow handshake protocol. In this paper, it depicts the usage of SafeFlow. The proposed protocol guarantees the security of the infrastructure itself, as there are there are also other proposals for developing security application on OpenFlow network infrastructure.

Keywords- Security, Privacy, Access Control policy, Software Defined Networks

I. INTRODUCTION

"Providing security is a fundamental need for enterprise networks, data center networks, and critical infrastructure networks". As these networks have many untrusted gadgets that may be vulnerable. At the point when a controller initiates an association with an OpenFlow switch the controller ought to authenticate the OpenFlow change according to a preexisting policy that's determine which OpenFlow change is to be viewed as trustworthy for a giving traffic. Our approach to authentication is called SafeFlow, an extension to the OpenFlow protocol handshake protocol that integrates trust negotiation to give authentication in the SDN infrastructure. Our approach to authentication establishment is based on the exchange of digital credentials to establish trust gradually. Digital credentials contain digitally marked assertions by a credential issuer about a credential proprietor. A credential utilizes name/value pairs to depict at least one attributes of the proprietor. One example of a digitally marked credential is the X.509v3 certificates. Since digital credentials can frequently contain touchy information, an access control policy will control credential disclosure.

As an example of trust negotiation, online car dealers may offer discount to government workers. At the point when a first-time client demands the discount, he won't know about the car dealer's policy for evidence of work status. One approach is for the server to transmit a policy to the customer. Such a policy could distinguish, that the purchaser must present an ID and a social security number (SSN) in request to make the e-purchase and get the discount. The customer thinks about his credentials (ID and SSN) as touchy information and is just willing to disclose his delicate credentials to a trusted server, which is an individual from the Better Business Bureau. The server at that point sends the customer a BBB part credential. Finally, the customer presents his ID information and SSN and gets the discount, Fig.1 portrays the authentication ventures for this example.

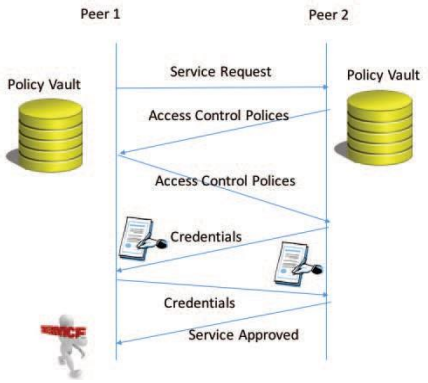

Fig 1: an example of Automated Trust Negotiation

As in the example above, a credential is released just when its access control policy is satisfied. Nonetheless, for a trust negotiation to be effective, some credential must be unprotected and can be released openly.

This paper is organized as the following; area II discusses related work; segment III depicts the representation of Access Control Policy (ACP) utilized in this paper. Area IV depicts SafeFlow protocol. Segment V gives the implementation of SafeFlow. Segment VI contains an example to illustrate the operation of SafeFlow and finally,

area VII has the ends and future works plans.

II. RELATED WORK

The security of Software Defined Networks, particularly the security of OpenFlow networks, is a concentrate completely explored right now as the OpenFlow protocol does not execute security independent from anyone else. There are works for developing applications on OpenFlow networks infrastructure, as there are others that look to guarantee the security of the infrastructure itself. As far as we could possibly know SafeFlow is the principal authentication protocol that introduces Trust Negotiation for authentication in SDN that actualize security in OpenFlow protocol itself, existing authentication protocols utilized in software defined networks, depend on application executed over OpenFlow networks infrastructure. AuthFlow is executed with an OF controller, an authenticator, and a RADIUS server. In the issue of exposing the full benefit of OpenFlow to each application without insurance is recognized, the authors propose PermOF with a lot of permission and isolation mechanism to apply at the API section. Resonance and Ethane are two proposals for authentication in a Software Defined Network. Both claim that hub authentication must be done through a site, in which the client must present their credentials. These approaches present a basic limitation that is the need for a hub has a program installed to accesses web content. In addition, another disadvantage of these proposals is to restrain authentication to username and password technique, while SafeFlow enable the authentication by exchanging of digital credentials where both OpenFlow switches and OpenFlow controllers can indicate and personalize their own attribute release policy.

III. ACCESS CONTROLPOLICY

In this paper, we speak to assets as propositional images. Each asset has a policy which is spoken to in the disjunctive normal form (DNF), for example the following is the policy (ACP) that administer when the asset C1 will be safely released, c1 ß s1V s2 which indicates that to release asset c1 either s1 or s2, must be released. Boolean TRUE means that the plan of action is free or unprotected and can be released immediately. Fig.2 demonstrates an example of strategies to be utilized in this paper, assume c1, c2, c3, c4, and c5 are OpenFlow Controller's assets while s1, s2, s3, s4, and s5 are OpenFlow switch's assets.

OpenFlow Switch's policies:	OpenFlow controller's policies:
s1 : (c1Ac5) ⩛(c2Ac3) s2 : (c2Ac3) s3 : (c3V c4) s4 : c3 s5 : (True)	c1 : (s3V s4) c2 : (s2As3) c3 : s4 c4 : (s1As5) c5 : (True)

Fig.2: an example of Access Control Policy (ACP)

IV. SAFE FLOW PROTOCOL

The SafeFlow protocol, appeared in Fig.3, is intended to help authentication between an OpenFlow switch and OpenFlow controller in Software Defined Networks, at whatever point an OpenFlow switch has mentioned access to a delicate asset from the OpenFlow controller. The controller will send a Negotiation Request message to the switch. The switch reacts with another Negotiation Request message that contains the mentioned asset in the event that it is free or unprotected, or it will counter demand with another asset from the controller. In request to lead an effective trust negotiation, the OpenFlow switch and the OpenFlow controller should initially agree on a trust strategy. In this work, we assume that both the switch and the controller agree on using the SPAN strategy.

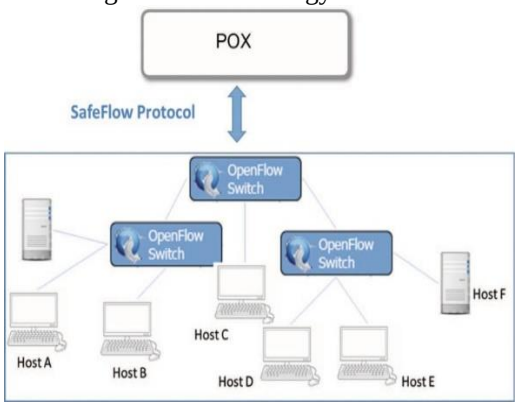

Fig.3: SafeFlow protocol in SDN

Vol 10, Issue 9, Sept /2019
ISSN NO:0377-9254

V. IMPLEMENTATION

SafeFlow is an extension to the Open Flow protocol, however SafeFlow introduces one new message type, the Negotiation message. The following syntax describes the new message type in SafeFlow that is simply for authentication purposes.

Struct {} Negotiation Request

There are two approaches to support authentication using trust negotiation: build a module above the OpenFlow protocol, or integrate trust negotiation into the OpenFlow authentication (TLS). SafeFlow is an example of the later approach. SafeFlow is implemented in Python and to be added to the OpenFlow v1.5 implementation

VI. EXAMPLE

As a proof of concept, we include this example to illustrate the operation of SafeFlow. Suppose a switch is trying to establish a connection with the OpenFlow controller, following the exchange of the Hello message, the SafeFlow protocol enters the negotiation phase in which the controller and the switch will take turns disclosing digital credentials until the negotiation succeeds or terminated. Fig.4 shows the SafeFlow protocol for trust negotiation.

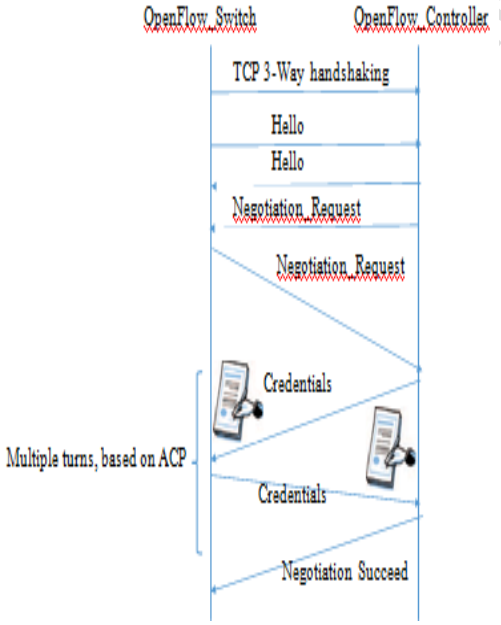

Fig.4: shows the SafeFlow protocol for trust negotiation.

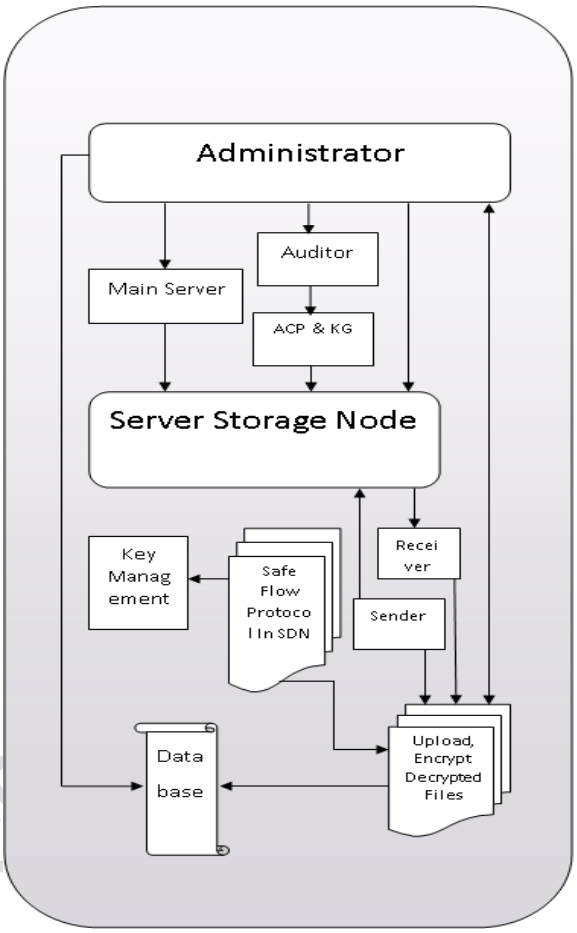

Fig.5: System Architecture

VII. RESULT

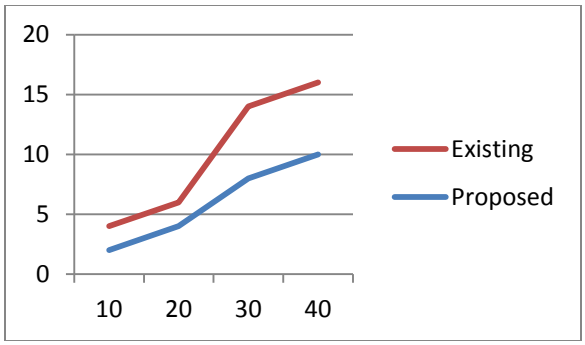

Fig 6. Time taken for transfer of data in proposed is lesser than existing system

VII. CONCLUSION AND FUTUREWORK

In this paper, we present a distributed negotiation protocol SafeFlow for authentication in software-defined networks (SDN). This algorithm is an extension to the OpenFlow protocol. As a future work, we will compare the performance of Safe Flow to other authentication protocols in SDN.

Vol 10, Issue 9, Sept /2019
ISSN NO: 0377-9254

REFERENCE

[1] Khan, J., Bobade, K., Hardas, M. "Negotiation Based on Individualization: Incorporating Personalization into Federation" in 5th International Conference on information and communication Technology, 2007, IEEE press, December 2007, Cairo, Egypt, pp 309-314.

[2] Winsborough, W.H., Seamons, K.E., Jones, V.E., " Automated trust negotiation" , In DARPA Information Survivability Conference and exposition. Jan 2002.

[3] D.M.F. Mattos, L. H. G. Ferraz, and O.C.M.B. Duuarte." Auth Flow: Authentication and access control mechanism for software defined networking" Univ. Federal Rio Janeiro, Riodejaneiro, Brazil 2014

[4] X. Wen, Y, Chen.C Hu, C.shi, and Y Wang, "Towards a secure controller platform for openflowapplications." In proc. 2ndACM SIGCOMM workshop Hot Topics so ft w: Defined Net w., 2013, pp 171-172.

[5] A. K. Nayak, A. Reimers, N. Feamster, and R. Clark, "Resonance: Dynamic access control for enterprise networks," in *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*, ser. WREN '09. New York, NY, USA: ACM, 2009, pp.11–18.

[6] M. Casado, M. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 1–12,2007.

[7] https://www.opennetworking.org/wp-content/uploads/.../openflow-switch-v1.5.1.pdf

AUTHOR'S PROFILE

Ms. SADAF KHALEEL has completed her B.Tech (CSE) from Shadan Women's College of Engineering and Technology, Khairtabad , JNTU University Hyderabad. Presently, she is pursuing her Masters in Computer Science and Engineering from Shadan Women's College of Engineering and Technology, Khairtabad, Hyderabad, TS. India.

Dr T. RAVI, Professor of Shadan Women's College of Engineering and Technology, Hyderabad. He is a graduate in computer science and engineering from Maguraj Kamaraj University, Masters and Ph.D. in computer science and engineering from Jadavpur University, Kolkata. He has more than 25 years of teaching experience in various engineering institutions in Tamil Nadu, Telangana and AP. More than 35 research papers are published in national and international journals and conferences and also 5 text books are published through various publications. He is the Recognized Research Supervisor in anna university and Satyabhama University, Tirunelveli