I. J. Computer Network and Information Security, **2017, 7, 20-28**
Published Online July 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2017.07.03

Trust Establishment in SDN: Controller and Applications

Bassey Isong
Computer Science Department, North-West University, Mafikeng, South Africa
E-mail: bassey.isong@nwu.ac.za

Tebogo Kgogo and Francis Lugayizi
Computer Science Department, North-West University, Mafikeng, South Africa
E-mail: kgogot@gmail.com, francis.lugayizi@nwu.ac.za

***Abstract*—**Software Defined Networks (SDNs) is a network technology developed to deal with several limitations faced by the current traditional networks. However, SDN itself is confronted with security challenges which emanates specifically from its platform, given the explosive growth in network attacks and threats. Though many solutions have been developed and proposed, the continual lack of trust between the SDN controller and the applications running atop the control plane poses a great security challenge. SDN controller can easily be attacked by malicious/compromised applications which can result in network failure as the controller represents a single point of failure. Though trust mechanisms to certify network devices exist, mechanisms to certify management applications are still not well developed. Therefore, this paper proposes a novel direct trust establishment framework between an OpenFlow-based SDN controller and applications. The objective is to ensure that SDN controller is protected and diverse applications that consumes network resources are always trusted throughout their lifetime. Additionally, the paper introduce the concept of trust access matrix and application identity to ensure efficient control of network resources. We believe that, if the proposed trust model is adopted in the OpenFlow architecture, it could go a long way to improve the security of the SDN**.**

***Index Terms*—**SDN, OpenFlow, Controller, Applications, Trust Establishment.

I. Introduction

Computer networks play a vital role in businesses and have become an integral part of all critical infrastructures in the society, academia and industries [1][2]. However, due to the proliferation in the number of services and devices that the networks support, it becomes complex to manage diverse the tasks a network is designed to handle. In particular, the traditional computer network architecture is crippled with issues such as communication handling between devices with complex protocols, difficulties in administering network polices, issues of security, scalability, robustness, expensive capital cost [1][3]. These challenges have limited the network operations and make it inherently complex to cope with the dynamic environment and significant services deliver to end users [1][4][5]. Thus, to deal with these limitations, a new network technology called Software Defined Networking (SDN) [4][6] is developed, having the capability to manage network resources flexibly and dynamically.

SDN is a network technology which allow networks to be programmable, more manageable, dynamic, modular, abstraction-based, and application aware [4][6][7][8]. SDN operates with an architecture that separates network control from its forwarding. It has a centralized control plane which lies between the application and data planes. The control plane houses the SDN controller which is the brain of the network or control the resources of the entire network [1][6][9]. SDN is gaining considerable attention due to its widespread applicability triggered mainly by its ability to programmatically control the behavior of the network as well as view in real-time, the state of the network control [4][5][10]. Moreover, SDN has greatly improved the use of network resources and provide benefits of lower operating expenses and capital expenditure [4][11].

Given the benefits offered, SDN itself is faced with a host of security challenges emanating from its platform [4][7]. These security challenges has emerged as a bottleneck to its growth and adoption [4][12][13]. One important challenge is the lack trust between the SDN controller and diverse management applications that runs atop then control plane. The OpenFlow-based SDN has a centralized controller, the only decision making entity in the network. This represent a single point of failure whereby, a failure will result to the failure of the entire networks. Thus, the controller poses serious security challenges to the entire network and has become a potential target by attackers. One way the controller can be attacked is through compromised or malicious applications in the application plane which utilize or consume network resources [12][14].These applications present a great and real challenge to SDN and can possibly do anything maliciously in the network such as exhaust resources, abuse control messages and overwrites flow rules if they are malicious or are compromised by an attacker [13].Consequently, the attacker can gain control of the entire network and do whatever pleases them or the network may fail completely.

Today, we live in a world where cyber-attacks are skyrocketing, and are becoming more sophisticated and complex, posing a key concerns for users, organizations and even the governments. Unfortunately, SDN has not yet matured towards mitigating several attacks. Though several security measures have been proposed and developed such as authentication, authorization, auditing and access control [1][12][13], several challenges still exist such as the establishment of trust between SDN entities [12][13]. Presently, mechanisms to certify network devices exist but trust establishment mechanisms to certify SDN applications are still lacking [12][13]. Furthermore, there are no existing mechanisms to identify if an application is a user, third-party or a network service application [12]. Therefore, proactive security measure is indispensable to protect the controller and other critical entities of the networks.

This paper proposes a unique centralized direct trust establishment framework to certify the different management applications that communicate with the SDN controller to assure that these applications are always trusted in their intended function. We also present a theoretical-design framework of the direct trust establishment system as a defensive mechanism to protect the network from unanticipated attacks.

The rest of this paper is organized as follows: Section II is the study background in terms of SDN and its security. In Section III we discuss various related works, Sections IV and V discusses the direct trust establishment system while VI present the proposed SDN controller-application trust framework. Lastly, Section VII is the study conclusion.

II. Background Information

In this section, we discuss in-depth, the SDN, its security and trust issues.

A. *Software Defined Networks*

SDN is an emerging network paradigm designed to get rid of the complexity but simplify network control and management. Its development is considered to be a kind of disruptive and innovative force in the realm of computer networks that has positively impacted all actors such as network devices, operators, service providers [1][2]. SDN technique is based on the programmability of the network and its services by decoupling the data plane and the control plane [1][13]. The core idea behind the decoupling in the SDN architecture stems from the fact that devices of the network such as switches, routers, etc. will be used for forwarding while the centralized layer, SDN controller in particular, will deal with the intelligences of the entire networks. The controller is positioned to provide an abstract view of the entire network through an interface, manages distributed network resources as well as decides on the control of the resources in a logically centralized fashion [1][9].

A typical SDN architecture consist of three key layers and two communication interface as shown in Fig. 1. The layers are the application layer, control layer and the data layer, while interfaces are the northbound and southbound [1][3].We provide a high level explanation as follows:

1) *Application layer:* It host and manages the execution of application and services that runs atop the network infrastructure which consume the network resources such as network virtualization, provisioning, intrusion detection/prevention system, security monitoring, load balancers, etc [1][12]. Applications are allowed to modify network resources such as policies and routers behavior with some degree of human intervention.
2) *Data layer:* It host the forwarding hardware that are responsible for the forwarding of data in the network such as the routers and switches. In this layer, the management functionality of the controller is implemented via the SDN-enabled switch [12].
3) *Control layer:* It has a logically centralized but physically distributed component called the SDN controller [1]. The controller manages the intelligences of the whole network, its programmability as well as the interactions between the application and the data layer. It provides network-oriented high level function to applications and maintain effective communication with network devices to perform specific tasks such as rules of forwarding and routing. Additionally, it provides a global view of the network.

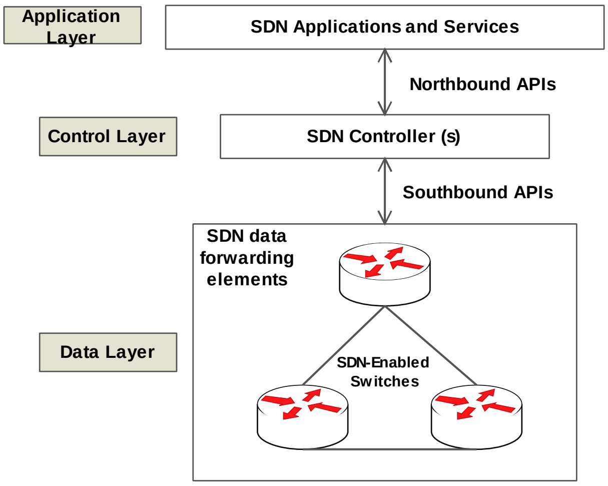

Fig.1. High-level view of SDN architecture

4) *Northbound SDN interface:* This interface are mainly application programming interface (API) designed to implement application-to-control layer communication. It implement the needed abstraction to facilitate the abstract view of the core network [1].
5) *Southbound interface:* This interface implements the control-to-data plane communication. It implement protocols required for effective communication with diverse devices of the network. In particular, the

communication between the controller and the switch. This interface facilitates the handling of traffic of existing switching devices of the data layer by administrators simply by pushing out the controller decisions [1][12].

Fig. 1, shows the decoupling nature of the SDN architecture that is not found in the conventional networks. Moreover, they have wide spread applicability such as in cloud computing and Internet of Things (IoT), as well as has the potential to reduce capital cost due to inexpensive components like switches, etc. SDN is gaining momentum and currently, the OpenFlow [1][9][15] is the ideal architecture used for implementing a prototype network application. However, the OpenFlow SDN-based architecture is faced with several challenges which are dominated by security issues [1][8]. In addition, several solutions have been developed and proposed with the goal of designing a secure and dependable SDN.

B. *SDN Security and Trust*

In this section, we provide discussion on both general security issues in SDN and trust management.

1) *SDN security:* Security can be viewed as an assurance of data authenticity, integrity and confidentiality [12]. Security is one of the driving force of SDN due to the sharp rise in sophisticated attacks on the networks [16]. However, SDN is an emergent technology for which security poses a great challenge. SDN raises new security and data protection threats originating from the programmability and the decoupling nature of its architecture [1][3][8][13][17]. Despite SDN benefits, it also brings about opportunities for attacks on the networks. This is due to the fact that security was not considered in its initial architecture [1] and security issues have not been thoroughly studied [18]. Consequently, the three layers of the OpenFlow-based SDN architecture are seen as the primary targets by potential attacks.

Several studies have highlighted the different security challenges in the SDN's OpenFlow architecture where different threats have been identified. Such threats include attacks on communication, controller – applications vulnerabilities, lack of trust between SDN controller and applications or switches [13][14]. Others include securing and protecting the controller, ensuring application integrity, unauthorized access, distributed denial of service (DDoS)[18][19]. For instance, the SDN can be attacked through the SDN API in the northbound or via compromised or malicious network applications running atop the controller [13][18]. Consequently, danger can be posed when a hacker exploit the network programmability to take over the control of the entire networks in which traffic can be rerouted or even duplicated to carry out unauthorized operations. In particular, the interaction between the SDN controller and the various applications (either user, third-party or network service applications [12][13][19]) that runs atop the network infrastructure poses great risks. Applications like intrusion detection system could request the network to filter traffic or a MapReduce-style application could request bandwidth guarantees to improve performance [1][12]. An attacker can exploit the vulnerabilities of these management applications to compromise the controller in terms of its confidentiality, integrity and availability [12]. If applications are malicious or compromised by an attacker, the control of the entire network can be gained to cause conflicting configuration and delicate users information can be accessed to forge authorized flows or even fail the entire networks [13][19]. Other challenges posed by these malicious or compromised applications include service chain interference, gateway to unauthorized access, manipulation of SDN internal storage, abuse of SDN control messages, exhaustion of resources, and so on [12][19].

With these security challenges faced by SDN, many research efforts have been channeled to improving them. This is because, threats emanating from cyber activities are growing exponentially and more sophisticated where the potential impact of a successful attack on network like the SDN may be catastrophic [20]. Thus, it is critically essential that various SDN security threats should be attended to at the early stage in order to reap the benefits it offers.

2) *Trust establishment:* To ensure networks reliability and eliminate attacks especially in an open and dynamic setting like SDN, trust and reputation are indispensable [21][22]. They are important in emergent network technologies like the SDN to defend and boost end-users confidence.Trust can be defined as the degree of belief that an entity (trustor) and other entities (trustees) are consistent in their behaviors in a normal way for any intended functions [5][22][23]. It is established based on one's historical experience and observation of others actions. Trust plays a critical factor in decision-making and its relationships between peers help establish confidence. On the other hand, reputation constitute the expectations one has about the behaviour of an entity based on previous behaviour. Reputation is sometimes confused with trust but is used to determine trust. Accordingly, trust model is the method used to specify, assess, setup, and ensure trust relationship among entities [23]. Trust management offers an effective way to assess trust relationships between entities and will help them in making intelligent decision to communicate and collaborate.

In the light of the above discussion where malicious applications can be used to compromise the SDN controller, it is important that some elements of trust exist between SDN applications and the controller in order to communicate in a trustworthy manner. Several studies have recommended the establishment of trust between

SDN controller and the network applications [8][12][13][14][19][24]. With the existence of dynamic trust model, malicious/compromised applications can be rid of and entities identity authenticated before control messages are exchanged [19][24]. In this case, the SDN controller can be protected by using trust to assess the trustworthiness of the SDN applications using behavior observation. However, trust establishment between SDN entities are still lacking. Most studies on trust establishment have been geared towards SDN controller-switch [12][13] while few exist for controller-applications such as in [18]. Thus, this paper proposes a theoretical-design framework for trust establishment between the applications and the controller to ensure that applications are trusted during their life-time.

III. Related Works

In this section, we highlight some of the research efforts that have been made to address security challenges in the SDN. We group these works into two categories: survey works that have made these challenges known and works that have already addressed some of the issues.

Akhunzada et al. [12] conducted a comprehensive survey on the state-of-the–art in the OpenFlow-based SDN security solutions on different layers/interfaces, current security issues, challenges and research directions for a secure and dependable SDN. The studies offered promising possible solutions such as trust establishment mechanisms between entities in the SDN and strong encryption. The survey found that SDN security is still at its nascent stage. Similar studies have been performed by Wenjuang et al. [13], Ali et al. [20], Scott-Hayward et al. [19] and Chen et al. [24]. Moreover, Kreutz et al.[14] strongly advocated for the importance of integrating security and dependability into SDN design and outline various threats categories which SDN can be attacked with such as, forged traffic flows, attacks on vulnerabilities in controllers and switches, attacks on control plane communication, lack of trust between controller and management applications and so on.

Wen et al. [25] provided a defensive measure against potential network attacks launched via the controller API interface to protect the networks from misconfiguration by malicious applications. In the approach called PermOF, kernel modules of the controller are isolated at runtime which prevent applications from calling the controller directly with a set of fine-grained permission. Porras et al. [26] also proposed a security enforcement kernel solution known as FortNOX to deal with the issues of malicious, compromised applications based on a role-based authentication. However, the approach is limited by not being able to determine appropriate security authorization levels. In a related study, Shin et al. [27] proposed an approach called ROSEMARY to advance the resilience of the control plane to both error-prone and malicious applications using a micro- network operating system (NOS) architecture. The solution splits network applications from the trusted computing end of the NOS, monitors and control both network resources each application consumed and application operations and so on. Chandrasekaran and Benson [16] also worked on an approach to deal with failures consequence of SDN application on the reliability of the controller. To avoid SDN application failure affecting the controller negatively, isolation layer and fault tolerance layers to ensure both security and availability were recommended.

Joeng et al. [18] proposed a trust support approach between the controller and applications in the SDN. It uses several 'redundant' controllers capable of running in different execution environments other than relying on one controller. All forms of network configuration requests emanating from the controllers are subject to comparison and only sent to the network if found sufficiently consistent and trustable. Accordingly, Yan et al. [28] also proposed a framework for 5G security and trust in a virtualized networking setting. Trust in computing platform and SDN security is ensured by applying adaptive trust evaluation, management and viable trusted computing technologies. Trustworthy services across the virtualized networks is securely deploy via cloud.

Sun and Yang [15] developed a trust establishment model and proposed approaches to analyze its process to protect distributed networks against several malicious attacks. The basic components of the model are shown on Fig. 2. The proposed analysis methods were implemented and validation performed using simulations. The analysis approaches are specifically used for comprehending trust establishment process as well as to quantitatively compare different trust models.

In these studies [12][13][14][19][20] and [24] brought together the state-of-the-art in OpenFlow-based SDN security challenges, solutions and research directions. [25][26][27] and [16] addressed the protection of the controller from compromised/malicious applications. In particular, while [26] enforces application authorization to protect the network, [27] and [16] protect the network against the threats from compromised/malicious applications. In addition, [18] and [28] deals with the use of trust between SDN entities to protect the SDN. Though the approach in [15] is not related to SDN, this paper is going to adopt the trust establishment system as a guide to the design of our trust model for SDN controller-applications.

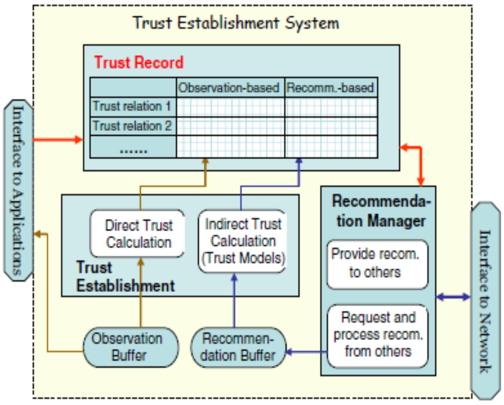

Fig.2. Elements of Trust Establishment System [15]

IV. Proposed Trust Establishment Framework

Confidentiality, integrity, availability supported by methods of authentication, authorization and so on constitutes the basic properties of a secure communication networks [12][17][20][29]. They combine together to provide a network in which its assets, data and communication connections through the network are made secured and protected against all forms of intentional or unintentional attacks and damage [29][30]. Trust establishment constitute an important security mechanism to defend against malicious attacks that has been common in today's networks. It has played critical roles in several network paradigms such as sensor networks [15]. The goal of trust establishment in networks especially in distributed networks is to assist in detecting computing and communicating entities that are not trustworthy as well as helps in decision making [15]. Based on the nature of the SDN architecture where only one controller controls the entire network, this paper deemed it important to protect the controller from various malicious/compromised applications. Therefore, this section discusses the proposal of a direct trust establishment model between the SDN controller and applications.

A. *Trust Model Design*

In order to model the establishment of trust between the SDN controller and its management applications, we adopted the trust framework proposed by [15] which deals with direct and indirect trust establishment and analysis designed as shown in Fig. 2. The choice of the trust model in [15] is based on the fact that it is effective for distributed networks and provides a guide for designing a better trust establishment approaches. To this end, we performed modification on the model to derive the trust model captured in Fig. 3. In particular, our approach rely only on the direct trust and the core elements are discussed.

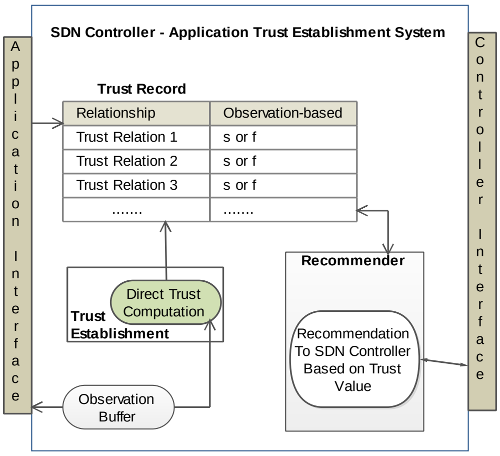

Fig.3. Trust establishment model

1) *Trust record:* This is a component of the trust model where information about trust relationship and the trust values are collected and stored. In the perspective of SDN, we assumed a trust relationship is established between the controller and the applications. Emphasis therefore, is placed on checking if the controller trust the applications to perform different tasks/actions with regards to network resource consumption. The relationship can be represented as: {controller: *application*, *action*}. Moreover, the measure of the trustworthiness in the trust relationship is given by a trust value.

2) *Direct trust and computation:* Trust establishment in computer networks can either be done by direct or indirect approach [15]. In this paper, we assume the direct trust establishment which is based on observation of the activities performed by SDN applications when comsuming network resources. That is, whether the interaction between controller-application is successful (***s***) or a failure (***f***) [15]. Thus, the direct trust value can be computed by $D_{Trust}(s,f)$ while DT_{Capp} represents controller-application the direct trust where DT_{CApp} can be a vector.

$$DT_{CApp} = D_{TRrust}(s,f) \tag{1}$$

We assume that the SDN controller is protecting itself from compromised/malicious applications. Thus, the choice of the direct trust is to enable the controller directly observe the behavior of the different and diverse applications to determine if they are malicious or not. Direct trust computation is the component designed for computing the trust value of the relationship between the two SDN entities, in terms of *success*, s or *failure*, f.

B. *Trust Establishment Analysis*

The components of the trust establishment analysis as adopted from [15] is shown in Fig. 4. They include:

- Direct trust calculation for trust evaluation based on previous interaction.
- The direct interaction between SDN controller and applications as well as their trust establishment.
- Direct trust analysis which is the core of trust establishment analysis.
- Metrics set for security and performance trust evaluation approaches.

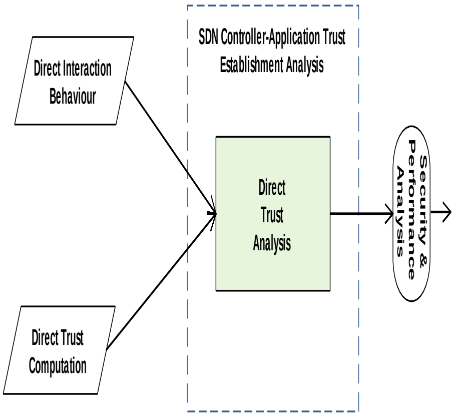
SDN Controller-Application Trust Establishment Analysis
Direct Interaction Behaviour
Direct Trust Computation
Direct Trust Analysis
Security & Performance Analysis

Fig.4. Trust Establishment Analysis

As shown in Fig. 4, both direct trust computation of the trust record and the interaction between SDN controller-applications are inputs to analyze trust in the system. The end-product will be set of metrics that will be used to assess the security/performance of the SDN controller. We provide an in-depth explanation as follows.

1) *Input:* To analyze the trust establishment, we employ the direct interaction between SDN controller and applications as well as the direct trust computation.

- Direct trust comuputation has already been discussed in Section IV A. It will be used to establish trust for applications that had previously interacted with the controller using the trust value *s* or *f*.
- Direct interaction and behaviour model: We assume there is interaction bewteen SDN applications and the controller. The probability of such interaction is thus, the trust establishment factor. This is used to establish trust for the first time interaction and it describe the behavior of good and bad applications. For clarity, we assumed applications are of three types: *the good* (g), *the bad* (b) and *the malicious* (m). For a given interaction between the two entities, result can either be ***s*** or ***f***. The probability of interaction with a good application which is success is ***αp*** and with a bad application which is also success is ***βp***. The probability of interaction with a malicious application which is success is ***γp***. The model can be described by the three parameters: αp, βp and γp. as shown in TABLE I.

Table 1. Model Inputs

Input Perspective	Parameters
Direct Interaction behaviour	α_p, β_p, γ_p
Direct Trust calculation	$D_{\text{Trust}}(s,f)$

V. Direct Trust Establishemnt

In the event of communication between the SDN controller and the applications, the approach to establish direct trust is shown in Fig. 3. To avoid malicious/compromised applications crashing the SDN controller, we suggest that a direct trusted connection be established and each entities (i.e. applications) authenticated before control messages are exchanged between them. In this case, we assumed the controller is active and continuously observing the behavior of each application. At the end of each interaction, the controller can update the trust record in the trust model which is used as a ticket for next interaction admission. To this end, we introduce the concept of Trust Access Matrix (TAM), application identity and secure handshake connection.

Table 2. Trust Access Matrix

S_U ID	R_1	R_2	R_3	R_n
APP_1	x	-	x	-
APP_2	-	x	-	x
APP_3	x	-	-	-
-	-	-	-	-
APP_{n-1}	x	x	-	x
APP_n	x	x	x	x

Obi=Ri= resource, X = read, write, modify, view, etc, APPi=application

A. *Trust Access Matrix*

TAM is the access matrix which is composed of the subjects, S_u and the Objects, O_b. The subjects are the various applications that will access the SDN controller while the Objects, O_b are the network resources managed by the SDN controller. The relationship between the S_u and the O_b is captured by TAM with rights drawn from a set of rights, X in each entry given by TAM[s_u, o_b], where s_uϵS_u, o_bϵO_b, and TAM[s_u, o_b] ⊆ X. The subject s_u has the set of rights TAM [s_u, o_b] over the object, o_b. Hence, the set of protection states of the system is governed by the triple (S_u, O_b, TAM). TAM is shown in TABLE II.

B. *Application Identity and Handshake Connection*

In order to facilitate the authorization of network resources for application to ensure easier access to the controller in terms of priority, configuration requests and others, the present SDN OpenFlow-based architecture has no mechanism in place to differentiate between user, third-party and network service applications that request or consumes network resources. Thus, to bridge this gap and to facilitate efficient control of resource allocation and utilization, we recommend the categorization of applications into types with identities, IDs. That is, different applications should be identified as follows: user applications with ID =, APP_u, Third-party applications with ID =APP_{tp} and the network service applications with ID = APP_{ns}. To control the access to and the consumption of network resources, each IDs will be allocated privileges/rights that are appropriate in the TAM as shown in TABLE II. Based on the privileges/rights given, the behavior of each application will be monitored whenever there is interaction and be used for the basis of trust establishment and onward permission for resources consumption if found trustworthy. The process involve is shown in Fig. 5.

```
      Process of Trust Establishment
i.    For a given $Su$ do
ii.   SDN Controller determine applications, $Su = \{S_{u1}, S_{u2}.... S_{uk}\}$ which it will directly interact with in a given time, t.
      The interaction is guided by the TAM maintained by the controller.
iii.  The interaction between the controller and the $Su$ may succeed or fail based on the direct interaction behavior and direct trust computation.
          Success indicate an APP accessed the right R designated for it, otherwise, fail.
iv.   SDN controller can updates its direct trust record about $Su$ based on the outcome: $f$ or $s$
v.    End for
```

Fig.5. Trust Establishment Process

To further ensure that the controller is protected from all forms of malicious attack, a secure connection between the SDN controller-applications is recommended. Thus, before a control messages are exchanged, the controller has to certify the identity of each application. This is possible by assuming a handshake connection as shown in Fig. 7.

VI. Proposed SDN Controller-Applications Direct Trust Framework

This section presents the architecture of the SDN and the trust establishment model between the SDN controller and the applications. As a core contribution of this paper, we present the proposed basic improvements to the existing SDN OpenFlow-based architecture as follows:

A. *Trust Establishment Model*

This model will be used to establish trust between the application and the controller to ensure that the applications are trustworthy enough to gain access to utilize network resources. The operations and the idea behind this model has been discussed in Section IV. The SDN trust framework is captured in Fig. 6.

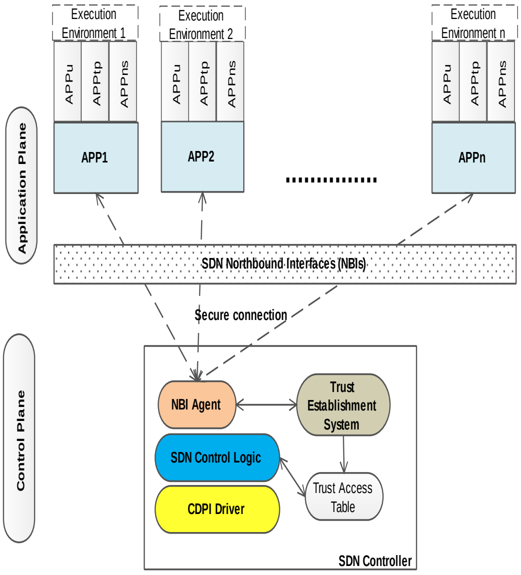

Fig.6. SDN Controller-Applications Trust Framework

B. *Trust Access Table*

In the SDN control plane, this table is proposed to store TAM. It identifies each application, SDN resources and the given rights or actions a particular application can exercise when interacting with the controller. The design of TAM has already been discussed in Section IV.

C. *Basic Operations*

In Fig. 6, we present the structure of the SDN controller-applications with a trust computing capability and secure connection. It shows how SDN applications will communicates with the controller to make request to use the network resources. Given the SDN architecture, the framework indicates that trust must be established first between controller-applications before an interaction or control messages can take place. The process involve is represented by a handshake transaction shown in Fig. 7.

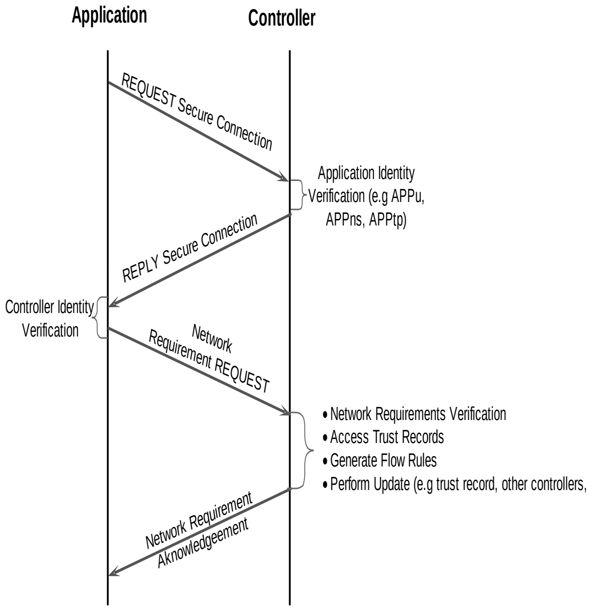
Application
Controller
REQUEST Secure Connection
Application Identity Verification (e.g APPu, APPns, APPtp)
REPLY Secure Connection
Controller Identity Verification
Network Requirement REQUEST
• Network Requirements Verification
• Access Trust Records
• Generate Flow Rules
• Perform Update (e.g trust record, other controllers,
Network Requirement Aknowledgeement

Fig.7. SDN Controller-Applications Handshake Secure Connection

As shown in Fig. 7, for any transaction between controller-application, the application will have to sends a secure connection request to the SDN controller which is managed by the NBI Agent. On receipt of the request, the NBI agent of the controller will have to verify the applications' identity to authenticate if is a valid application (e.g. APP_u, APP_{tp} and APP_{ns}) or unrecognized application. Upon verification, the controller sends response back to the application confirming if connection is granted or not. Then, the application authenticates the controller's identity and finally sends its network requirements request (e.g. network statistics, configuration, bandwidth, latency, and so on) depending on the response from the controller. At this juncture, upon the receipt, the controller NBI agent will authenticate the network requirement requests, check and resolve any conflicting policy based on the following factors:

- If the request is for the first time, in accordance with TAM, the controller will generate the necessary flow rules to implement the application service. Moreover, it will update the trust record based on the interaction and behavior of the application with ***s*** or ***f***, updates other controllers if available and finally sends the acknowledgment back to the SDN application.
- If the request is not for the first time, the controller will first check the trust record for previous connection, interaction and behavior. In this case, if the trust value is ***f***, the SDN controller will halt the application from performing any operation. But if ***s***, the controller will generate the necessary flow rules to implement the application service in accordance

with TAM. Furthermore, the trust record will be updated based on the outcome of such transaction: *s* or *f*, updates other controllers if available and finally sends the acknowledgment back to the SDN application. Note, at the end of each interaction, the trust record is updated with the current trust value and overrides any previous record.

VII. Conclusion

Software defined network is a network paradigm with an innovative improvement that enable network administrators to configure and manage network resources very quickly. It also assist in meeting changing needs of the users/devices dynamically with flexible network traffic flow adjustment. Nevertheless, in spite of the benefits and SDN being at its nascent stage, SDN is faced with series of challenges where security is at the top. Therefore, in this paper we identified one security challenges faced by OpenFlow-based SDN and proposed a countermeasure. This paper proposed and presented a novel trust establishment framework to ensure the communication between the SDN controller and the different applications are trusted and safe. We presented the design of the trust framework and theoretically discussed the operations of each component. With the framework, we believe that the controller being a single point of failure in the SDN can be protected given different sophisticated attacks that exist today. By having trusted applications, the controller can be protected and intelligent decisions can be made to communicate and collaborate effectively. Though the framework is theoretical, we consider it an important aspect of our future works to implement the idea discussed in this paper and evaluate its effectiveness and performance on a real-world network. The importance of this paper is that it serve as a stepping stone in the design and implementation of an effective security countermeasure in the OpenFlow-based SDN.

References

[1] Da Silva, AS, Smith, P., Mauthe, S. "Resilience support in software-defined networking: A survey." *Computer Networks* 92 (2015): 189-207.

[2] Ding, AY, Crowcroft, J. Tarkoma,S. Flinck,H. Software defined networking for security enhancement in wireless mobile networks. Computer Networks 66 (2014) 94–101

[3] Benabbou, J. and Idboufker, N. Software-Defined Networks, Security Aspects Analysis. 2015 11th International Conference on Information Assurance and Security (IAS), 2015.

[4] Sezer, Sakir, et al. "Are we ready for SDN? Implementation challenges for software-defined networks." *IEEE Communications Magazine* 51.7 (2013): pp.36-43.

[5] Grandison, Tyrone, and Morris Sloman. "A survey of trust in internet applications." *IEEE Communications Surveys & Tutorials* 3.4 (2000): 2-16.

[6] Sezer, Sakir, et al. "Are we ready for SDN? Implementation challenges for software-defined networks." *IEEE Communications Magazine* 51.7 (2013): 36-43.

[7] Bakshi, Kapil. "Considerations for software defined networking (SDN): approaches and use cases." *Aerospace Conference, 2013 IEEE*. IEEE, 2013.

[8] Jeong, J. Seo, J. Cho, G. Kim, H. Park, J. A Framework for Security Services Based on Software-Defined Networking," *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, Gwangiu, 2015, pp. 150-153.

[9] Bakshi, Kapil. "Considerations for software defined networking (SDN): approaches and use cases." *Aerospace Conference, 2013 IEEE*. IEEE, 2013.

[10] Govindarajan, Kannan, et al. "Realizing the quality of service (QoS) in software-defined networking (SDN) based cloud infrastructure." *Information and Communication Technology (ICoICT), 2014 2nd International Conference on*. IEEE, 2014.

[11] Raza, Muhammad H., et al. "A comparison of software defined network (SDN) implementation strategies." *Procedia Computer Science* 32 (2014): 1050-1055.

[12] Akhunzada, A, Gani, A.,Anuar, N.B, Abdelaziz, A. Khan, M.K, Hayat, A., Khan, S.U. Secure and dependable software defined networks. *Journal of Network and Computer Applications* 61(2016) pp.199–221.

[13] Li, W., Meng, M., Kwok, L.M. A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures. *Journal of Network and Computer Applications* Issue 68, pp.126–139, 2016.

[14] D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, 2013, pp.55-60.

[15] Y. L. Sun and Y. Yang, "Trust Establishment in Distributed Networks: Analysis and Modeling," *2007 IEEE International Conference on Communications*, Glasgow, 2007, pp. 1266-1273.

[16] B. Chandrasekaran and T. Benson, "Tolerating SDN application failures with LegoSDN," in *Proc. 13th ACM Workshop Hot Topics Netw.*, 2014, p. 22.

[17] Hu, Zhiyuan, et al. "A comprehensive security architecture for SDN."Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on. IEEE, 2015.

[18] Betgé-Brezetz, S. Kamga, G. Tazi, M. "Trust support for SDN controllers and virtualized network applications," *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, 2015, pp. 1-5.

[19] S. Scott-Hayward, S. Natarajan and S. Sezer, "A Survey of Security in Software Defined Networks," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 623-654, Firstquarter 2016.

[20] S. T. Ali, V. Sivaraman, A. Radford and S. Jha, "A Survey of Securing Networks Using Software Defined Networking," in *IEEE Transactions on Reliability*, vol. 64, no. 3, pp. 1086-1097, Sept. 2015.

[21] Ganeriwal, S and Srivastava, M.B. "Reputation-Based Framework for High Integrity Sensor Networks," In the Proceedings of ACM Workshop Security of Ad Hoc and Sensor Networks, October 25- 29, 2004, Washington, DC, USA, pp. 66-67.

[22] Mármol, F. G., & Pérez, G. M. Security Threats Scenarios in Trust and Reputation Models for Distributed Systems. Elsevier Computers & Security, 28(7), 545–556, 2009.

[23] Z. Yan and C. Prehofer, "Autonomic Trust Management for a Component-Based Software System," in *IEEE Transactions on Dependable and Secure Computing*, vol.

8, no. 6, pp. 810-823, Nov.-Dec. 2011.

[24] J. Chen, X. Zheng, and C. Rong. “Survey on software-defined networking”. In: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 9106.1 (2015), pp. 115–124. issn: 16113349.

[25] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, “Towards a secure controller platform for OpenFlow applications,” in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics in Software Defined Networking (HotSDN)*, 2013, pp. 171–172.

[26] P. Porras *et al.*, “A security enforcement kernel for OpenFlow networks,” in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 121–126.

[27] S. Shin *et al.*, “Rosemary: A robust, secure, and high-performance network operating system,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2014, pp. 78–89.

[28] Z. Yan, P. Zhang, and A. V. Vasilakos, “A security and trust framework for virtualized networks and software-defined networking,” Security and Communication Networks, 2015.

[29] Scott-Hayward, Sandra, Gemma O'Callaghan, and Sakir Sezer. "Sdn security: A survey." *Future Networks and Services (SDN4FNS), 2013 IEEE SDN For*. IEEE, 2013.

[30] Chourishi, Dharmendra, et al. "Role-based multiple controllers for load balancing and security in SDN." *Humanitarian Technology Conference (IHTC2015), 2015 IEEE Canada International*. IEEE, 2015.

Authors’ Profiles

Bassey Isong received B.Sc. degree in Computer Science from the University of Calabar, Nigeria in 2004 and M.Sc. degrees in Computer Science and Software Engineering from Blekinge Institute of Technology, Sweden in 2008 and 2010 respectively. Moreover, he received a PhD in Computer Science in the North-West University, Mafikeng Campus, South Africa in 2014. Information Systems. Currently, he is a Senior Lecturer in the Department of Computer Sciences, Mafikeng Campus, North-West University. He is also a member of IEEE Computer and Communication Societies. His research interests include Software Engineering, Requirements Engineering, Software Maintenance, Cybersecurity, Software-Defined Networks, and Cloud Computing.

Tebogo Kgogo is a postgraduate student of the department of Department of Computer Sciences, FAST, North-West University, Mafikeng Campus. He is also CSIR research student. His research interests include: Software Defined Networks, Computer Security and Cloud Computing.

Francis Lugayizi holds B.Sc. (Hons) degree in Computer Science (2011), M.Sc. degree in Computer Science (2014) and PhD in Computer Science (2016) at the North-West University, Mafikeng, South Africa. Currently, he is a Lecturer in the Department of Computer Sciences and a Faculty member of FAST, North-West University, Mafikeng Campus. His research interests include: Cloud Computing, Networks and Databases.

How to cite this paper: Bassey Isong, Tebogo Kgogo, Francis Lugayizi,"Trust Establishment in SDN: Controller and Applications", International Journal of Computer Network and Information Security(IJCNIS), Vol.9, No.7, pp.20-28, 2017.DOI: 10.5815/ijcnis.2017.07.03