An Analysis on Software Defined Wireless Network Using STRIDE Model

Arockia Panimalar.S [1], Nishanth.R[2], Sathish.G[3], Manikandan.G[4]

[1] Assistant Professor, Department of BCA & M.Sc SS, Sri Krishna Arts and Science College, Coimbatore, India
[2,3,4] III BCA, Department of BCA & M.Sc SS, Sri Krishna Arts and Science College, Coimbatore, India

---***---

Abstract - *The present mobile and remote system are becoming quicker in size and complex to quantify the administrations. Security is a standout amongst the most essential angles for such complex system and should be checked appropriately to give early identification of security ruptures and Denial of Service assault. Tools that measure such detection of network threats and monitors network services requires interior security in their own particular component. This paper examines two of such checking and estimation apparatuses: sFlow and FlowVisor for hidden Software Defined Wireless Networking (SDWN) condition by applying STRIDE threat model. This analytical study represents that, sFlow requires an external secure deployment environment to ensure security in data flow and data store for SDWN. FlowVisor accompanies secured get to control in information store wherein separated stream cut requires instrument that enhance its security.*

Key Words: **Software Defined Wireless Networking (SDWN), STRIDE, OpenFlow, sFlow, FlowVisor**

1. INTRODUCTION

Wireless Networking turns into the most versatile innovation for adaptability and portability in human life. For the most recent couple of years, Software Defined Wireless Networking (SDWN), a branch of Software Defined Network (SDN) has been a key research innovation to dissect and legitimate administration of the thickly populated cellular network [1] [2]. Programming Defined Wireless Networking (SDWN) guarantees straightforward and adaptable system design and successful portability administration of the IP networks. The Software Defined Wireless Networking (SDWN) automatically concentrates and isolates the control plane (otherwise known as. Network OS) from the data plane (otherwise known as Forwarding plane). A regular engineering of SDN is delineated in Fig. 1. The southbound interface is a medium between the control plane and data plane while northbound is layer between application plane and control plane. The southbound interface prepares the controllers to gather data about Mobile Nodes (MNs) and transmits and gets bundles to and from MNs utilizing SDWN components [3]. To guarantee qulaity of service in SDWN and persistent network services, operators need to monitor the network and do legitimate service measurements from time to time. Such observing will help in in analyzing network parameters, i.e. throughput, roundtrip transmission time, data transfer capacity in the remote connection, mobility frequency and preparing a real-time view of the network service standard on industry level. For such analysis, observing and estimation, different open source and business innovation and instruments are accessible for SDWN including sFlow [4], FlowVisor [5], BigSwitch [6], BigTap [7], SevOne [8] and so on. These tools provide the operator with capabilities to perform troublesome network activities and even monitor, detect and indicate security attack in progress on a certain network entity.

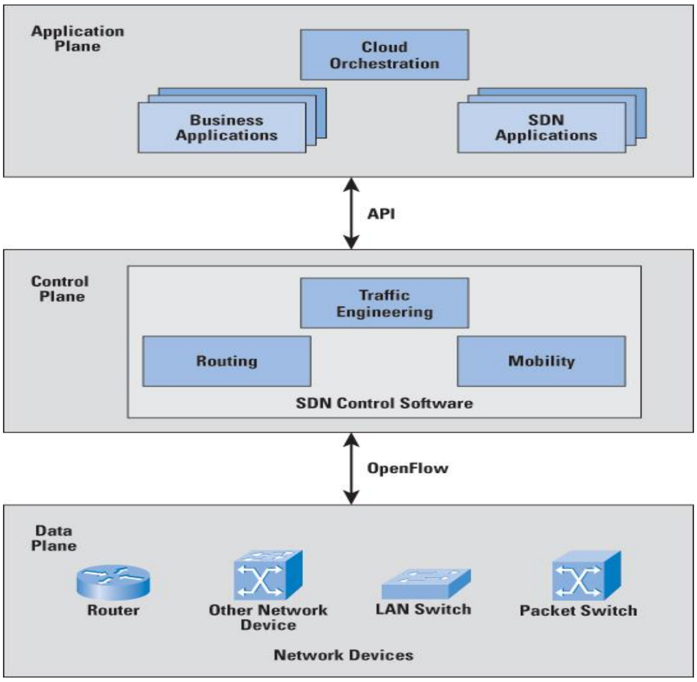

Fig 1: SDN Architecture with Control and Data Plane

Beforehand a few research works is performed on dissecting the security of OpenFlow-based SDN environment. Analysis on 4D, PCE and SANE-based SDN architectures is performed in paper [9], security use of SDN is completely investigated and assessed in [10]. First, sFlow was represented as an effective and scalable vulnerability mitigation mechanism for SDN [11]. FlowVisor turned a better solution for network virtualization [12] and powerlessness answer for flow isolation is proposed and assessed nearby [13]. Among the tools that screen and measure the SDN, a comparative study between sFlow (Open-Source) and BigTap (commercial) is illustrated in paper [14]. Be that as it may, a security and risk characterized ponder in SDWN monitoring and measurement tools, those focuses on Open-Flow flow entries and communication among multiple controllers in wireless platform, is the primary goal of this perspective study. Consequently, sFlow and FlowVisor are decided for above stream conditions.

The structure of this paper is as per the following. In Section II, the STRIDE and Data Flow approach is portrayed. In Section III and IV, security and threat risks of sFlow and FlowVisor are broke down. Segment V represents a comparative report between the two monitoring tools and along these lines closing the paper in Section VI with future prospects of this study outcome.

2. METHODS FOR IMPLEMENTING SDN

Several approaches are available for implementing SDN concept including OpenFlow that separates the control and forwarding plane in the network architecture. SDN approaches were generalized using a concept of OpenFlow and were introduced in mid-1990s.

A. OpenFlow

According to OpenFlow specification in [15], any OpenFlow switch holds flow entries that contain incoming packet header information, packet handling action for matched packet entries in the list and statistics of number of bytes, packets in a particular flow and time since last pass. Packets as arrive at any OpenFlow switches, it executes the packet header information and try matching the existing flow entries. When the information does not match any of the flow tables, switch then pass the packet to the controller to take action and update the flow entries accordingly with required information of the packet. When it's a match switch performs and forwards the packet to its next destination on the basis of routing flow table information in it.

B. Software Defined Wireless Network

As SDN brings more advantage in connecting into the internet, Software Defined Wireless Network (SDWN) has got much importance and emerging research field with attention. SDWN is oriented towards the mobile and wireless network devices and aims at the research and study of crucial technologies for the future mobile and wireless network. This SDWN architecture is composed of both North-South and East-West network dimension where East-West operates for wireless and mobile devices using intercontroller protocols such as Border Gateway Protocol (BGP) [16]. Hence, security of the underlying network depends on the secured flow information and control plane. Tools that monitor and measure and flows between SDWN entities, therefore, requires security from external access and service oriented attacks. This study is concerned about sFlow and FlowVisor as one of these tools.

C. Threat Modelling and STRIDE

Threat Modelling used to refer to analyzing any software or system or organizational network. Threat Modeling encompasses a wide variety of activities in the elicitations and analysis of security mechanisms in deployed designs and network [17]. Some of the mostly applied models include DREAD [18], Octave [19], STRIDE [20], Generic Risk Model, Guerilla Threat Modelling, Process for Attack Simulation and Threat Analysis (PASTA), Trike etc [21]. DREAD provides threat identification rate as SQL injections and provides the subjective assessments by the threat reporter. Octave model is best suited for complex and larger system where STRIDE focuses on network based application and systems. Trike helps security auditing process with distinct risk-based implementation than others, however, is yet in experimentation stage and lacks proper documentation and support. PASTA includes risk management steps in the final stage of the process and is not limited to a specific risk calculation formula [22]. Thereby, introduced by Microsoft, STRIDE model method is used to identify and evaluate the security threats on OpenFlow based SDWN network measurement and monitoring tools: sFlow and FlowVisor. STRIDE threat model reveals if a system or software in concern is vulnerable to Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS) and Elevation of Privilege threat [20]. Each of the STRIDE threats can be mapped to one security property as shown in Table 1. and described in the following:

a)Spoofing: In spoofing malicious user or program masquerades gain illegal access in privileged data by falsifying user information.

b)Tampering: Data tampering involves malicious modification of information and resources i.e. alteration of data as it streams between two PCs over an open network called the Internet.

c)Repudiation: Repudiation threats are associated with malicious users and masquerades who performs an action and deny without other parties having any way to prove otherwise—for example, an attacker controller performs an illegal operation in a SDN that lacks the ability to trace the prohibited operations.

d)Information Disclosure: This treat means the illegitimate availability of resource information of the system or network or software to malicious and unauthorized users or programs.

e) Denial of Service: This treat causes service unavailability to the authorized legitimate users or programs.

f)Elevation of Privilege: In this type of threat, an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system.

This treat can cause penetration of all system or network defense and declares it a trusted system.

Table 1 presents the STRIDE threat categorization model, based on the above definitions, which includes the corresponding security property and default controls associated with the threat type.

TABLE 1: Threat Categorization, Security Properties and Controls [17]

Threat	Property	Controls
Spoofing	Authentication	Authentication Stores, Strong Authentication mechanisms
Tampering	Integrity/ Access Controls	Crypto Hash, Digital watermark/ isolation and access checks
Repudiation	Non Repudiation	Logging infrastructure, full packet-capture
Information Disclosure	Confidentiality	Encryption or Isolation
Denial of Service	Availability	Redundancy, failover, QoS, Bandwidth throttle
Elevation of Privilege	Authorization /Least Privilege	RBAC, DACL, MAC, Sudo, UAC, Privileged account protections

Data Flow Diagrams (DFD) are used to graphically represent any system [17]. DFDs use a standard set of symbols consisting of four elements: data flows, data stores, processes, and interactors [17]. In Table 2, DFD elements are identified as a means of eliciting information which can be used to drive STRIDE threat analysis. As illustrated in Table 3, each DFD elements can be vulnerable to one or many STRIDE threats.

TABLE 2. DFD elements and their representation [17]

Name	Representation	Definition
Data Flow	Directed Arrow	Data sent among network elements
Data Store	Parallel Lines	Stable Data
Process	Circle	Programs or applications that configures the system
Interactors	Rectangular Box	Endpoints out of system scope to control
Trust Boundaries	Dotted Line	Separation between trusted and untrusted elements of the system

TABLE 3. STRIDE Threats per DFD element [17]

Threat	Data Flow	Data Store	Process	Interactors
Spoofing			Yes	Yes
Tampering	Yes	Yes	Yes	
Repudiation		Yes	Yes	Yes
Information Disclosure	Yes	Yes	Yes	
Denial of Service	Yes	Yes	Yes	
Elevation of Privilege			Yes	

3. sFLOW

sFlow is an open source sampling technology and traffic measurement and monitoring tool for OpenFlow network [4]. It is a traffic monitoring solution embedded with switch and router of any possible OpenFlow based SDWN. Primary elements of sFlow system consists sFlow agents and sFlow collector, illustrated in Fig. 2. Agent is the software process that is remotely configured using a Management Information Base (MIB) within the device. Consolidating the interface counters and flow tests into sFlow datagrams, these datagrams are sent to the sFlow collector installed in the checking host through the SDWN environment utilizing Simple Network Management Protocol (SNMP) [23]. Including sFlow's own collector sets: sFlow-RT, sFlow-Trend, sflowtool, this sampling tool also support the third party collectors: VitalSuit, Peakflow, Kentik Detect and FlowTraq - those handle more details of sFlow datagrams [4]. Illustration in Fig. 2 represents the Data Flow Diagram (DFD) of sFlow that uncovers the crucial security risk. sFlow doesn't provide any security mechanism for data flow rather depends on secure third party management environment for sFlow agents.

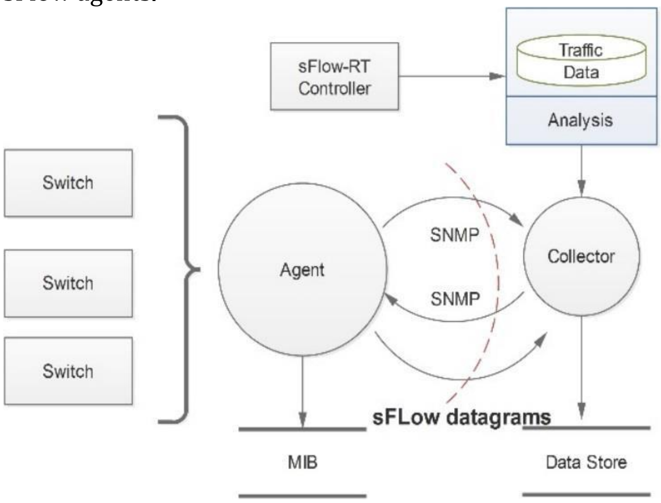

Fig. 2: sFlow Data Flow Diagram

A. Data Flows

Data flows are vulnerable to Tampering, Information Disclosure and DoS attack in absence of proper security mechanism. A physical interface of switch, routers is potential data sources in the underlying SDWN. These provides sampled data packet to sFlow agents for measurement. sFlow agents combine packet flow sampling and counter sampling to sFlow datagrams. The sFlow Datagrams are used to immediately forward the sampled traffic statistics being unencrypted to a sFlow Collector for analysis [24]. As collectors can be vendor provided, security of the received datagrams depends on vendor's will of deployment and how they process the data. Hence, sFlow doesn't provide any security mechanism. For security reasons SNMPv3 should be used to configure and control the sFlow agents to encrypt and authenticate the datagrams before transmitting to the collector [24].

B. Data Stores

According to Table 1 data store are prone to Tampering, Information Disclosure and DoS attack vulnerabilities alike data flows. MIB contains information about sFlow agents, collector ports and even IP addresses. Using SNMP, sFlow agents can be configured through a local Command Line Interface (CLI) or SNMP commands. In order to decline any anonymous actions, switches and routers in the network should have some Access Control (AC) mechanisms, i.e. Discretionary Access Control (DAC), Role-Based Access Control (RBAC) to ensure the interface's security [14]. In inverse case, if CLI is available from unapproved client, MIB in sFlow is powerless against data Tampering, traffic Information Disclosure and even DoS assault, holds the MIB flow enteries and authority subtle elements open for unapproved get to and considerably aggressor can alter the information. If there should be an occurrence of SNMPv1, SNMPv2 communication with collector is at comparable dangers.

C. Interactors

sFlow agents performs one-way communication with the sFlow collectors and sends the combination of packet based and time-based sampled traffic data [24]. According to Table 1. they are not considered as interactors.

D. Processes

sFlow agent processes are not accessible through interfaces, therefore the STRIDE method is not applied. The collector should check the time-based counter number of the sFlow datagrams to provide a security mechanism against spoofing attacks [24].

E. Summary

Above analysis clarifies that sFlow requires a third party deployment environment for security needs. However, ensuring the Transport Layer Security (TLS) among sFlow agents and collector, sFlow itself can emerge as a secured SDWN monitoring and measurement tool for wireless and Table 4 shows the probable vulnerabilities of sFlow agents. Adapting an access control mechanism can eliminate the security risks to a certain level of tampering and MIB information disclosure.

TABLE 4. sFlow Vulnerabilities

Threat	Data Flow	Data Store	Solution
Tampering	Yes	Yes	ACL/RBAC/DAC for CLI, NPMv3, TLS
Information Disclosure	Yes	Yes	TLS
Denial of Services(DoS)	Yes	Yes	AC in CLI for MIB Security, TLS

4. FLOWVISOR

FlowVisor is an OpenFlow controller works as a proxy in between the OpenFlow switches and several multiple OpenFlow controllers, allowing visualization of OpenFlow physical infrastructure into different virtual networks [5]. Using OpenFlow protocol, FlowVisor controls underlying network, dividing the resources into slices isolated from each other. And delegates control of each slice to a different controller [5]. FlowVisor provides isolation for topology and addressing space. FlowVisor is architecturally a neutral transparent proxy and makes no assumption about the functions and operations of the switches and controllers. FlowVisor sits between each of the controllers and switches making sure that the guest controller has full accessibility of the switches maintaining the flows that define the corresponding slice. The DFD in Fig. 3 represents data flow between OpenFlow enabled switches and controllers where messages are intercepted through FlowVisor.

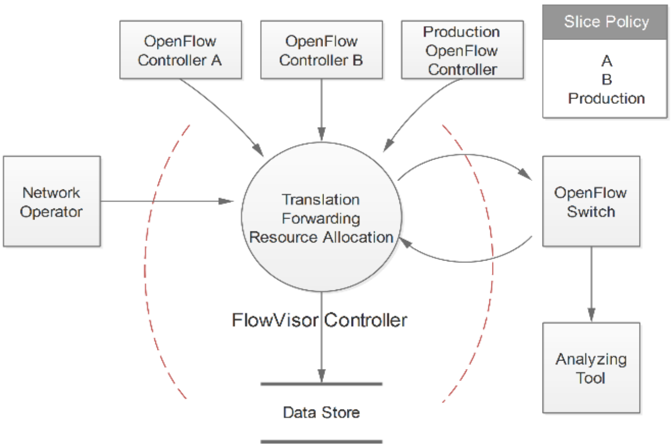

Fig. 3: FlowVisor Data Flow Diagram

FlowVisor partitions the base transmission transfer speed for each slice doling out particular data rate to a set of flows from that slice. FlowVisor screens each flow-entry for each guest controller and portions the flow table among the switches. Switches are arranged by the resource allocation and directing approaches cuts of the FlowVisor controller. Slices are isolated and have their own 'flowspace' or set of region of data flows. These isolated slices can be broken allowing different attacks.

A. Data Flows

FlowVisor adopts slicing policies for each guest controller. Activity sent from the production network and guest controllers if matches the sending enteries in the FlowVisor are sliced for relevant switches as per 'flowspace'. Diverse slices having adaptable and distinctive flow policies are emphatically separated. Any traffic that does not coordinate the current sending enteries are sent to the production controller for inclusion. Production controller subsequently revises the relevant slice. Assault on slice policy reworks from assaulting entity can make vulnerabilities such system with FlowVisor. Assuming, in this manner, data is sent from an aggressor, the controller can't identify as a result of policy revise and causes altering of flow rules and the system data and even DoS dangers.

B. Data Stores

The switch arrangement is put away in the flow enteries of the cuts by the respective guest controllers. This permits data movement validation to flow between the controllers and switches inside the wireless OpenFlow network even under portability circumstances. This mechanism ensures that data is secured against Tampering, Information Disclosure and Spoofing threats.

C. Interactors

FlowVisor's Command Line Interface (CLI) provides control access to users for data and slice configuration. CLI uses user-authentication in terms of username, host name and port number on accessing the interface and slices and therefore secure from Spoofing, Tampering, Repudiation and Elevation of Privilege threats.

D. Processes

Slice processes are owned by the admin and groups of the network operators and thereby Spoofing, Repudiation, DoS and tampering threats are unable to make the network vulnerable in FlowVisor's process.

E. Summary

FlowVisor is open source to access controller's processes, data flow and action support for slices. Although this tool has separate production controller and isolated slices to perform the flow independently against any attacking entity, FlowVisor is vulnerable to different threats at different flow status described in Table 5.

TABLE 5: FlowVisor Vulnerabilities

Threat	Data Flow	Solution
Tampering	Yes	TLS
Information Disclosure	Yes	TLS
Denial of Services (DoS)	Yes	Access Control in CLI for policy rewrite, TLS

Adjusting Transport Layer Security can safeguard the arrangement revise production controller for unmatched packets where virtual controller can't change the MAC and IP address for the packets uninhibitedly.

5. COMPARISON BETWEEN sFLOW AND FLOWVISOR

sFlow and FlowVisor both provide different network monitoring and measurement functionalities. The comparative threat model analysis of them is illustrated in Table 6. Above investigation holds sFlow giving no security in data flow and data store in DFD wherein FlowVisor acquires security threat vulnerabilities in disengaged cuts. This makes FlowVisor defenseless against Spoofing, Tampering and Information disclosure, even postponement and Denial of Service dangers in data flow. However, FlowVisor guarantees security of switch information put away in its own controller where sFlow relies upon external secure environment to guarantee security in MIB data storage and flow entry information. This makes sFlow helpless against spoofing, DoS and information divulgence risk as switching operators send decoded datagrams to the collector. Utilizing Transport Layer Security (TLS) in sending the datagrams to the collectors can take out information exposure threats wherein tampering can be handled utilizing access control mechanism in CLI, agent arranging SNMPv3 protocols. FlowVisor includes access control in CLI for slice information which protects it from spoofing, repudiation and elevation of privilege attacks from any kind of malicious user or masquerades.

TABLE 6: Comparison of FlowVisor and sFlow tools

Threat	Data Flow	Data Store
Tampering	FlowVisor, sFlow	sFlow
Information Disclosure	FlowVisor, sFlow	sFlow
Denial of Services (DoS)	FlowVisor, sFlow	sFlow

6. CONCLUSION

In this paper, an analysis on wireless SDN monitoring tools, sFlow and FlowVisor, in terms of STRIDE security threat model where both has different functionalities and vulnerabilities in handling data traffic flows and network entities. This analysis will provide suggestions in handling the above mentioned security threats in SDWN using existing well-to-do mechanisms. These study fall in the category of security-centric SDWN and will be viable in doing research on OrchSec wireless architecture [25].

7. FUTURE ENHANCEMENT

The future prospects of this SDWN security analysis will lead to persistent research on assessment of SDWN appliance in data center, cognitive networks and mobile communication. As future work, the researchers would plan to study FlowVisor topology isolation mechanism and queue-based bandwidth isolation mechanism in securing the underlying SDWN network. Prototyping the network in real time SDWN network devices and environment will be interesting and a big challenge ahead.

8. REFERENCES

[1] Bernardos et al., "An architecture for software defined wireless networking".

[2] M. R. Sama, L. M. Contreras, J. Kaippallimalil, I. Akiyoshi, H. Qian, and H. Ni, "Software-defined control of the virtualized mobile packet core," IEEE Communications Magazine, vol. 53, no. 2, pp. 107–115, Feb. 2015.

[3] Y. Wang, J. Bi, and K. Zhang, "Design and implementation of a software-defined mobility architecture for IP networks,"

Mobile Networks and Applications.

[4]sFlow, "Making the network visible," 2003. [Online]. Available: http://www.sflow.org/.

[5]"FlowVisor,".[Online].Available:https://openflow.stanford . edu/display/DOCS/Flowvisor.

[6]T. Turner, "Big switch networks, Inc," Big Switch Networks,2014.http://www.bigswitch. com/

[7]BigSwitch Networks, "Big tap monitoring fabric," Big Switch Networks, 2014. [Online]. Available: http://www. bigswitch .com/topics/big-tap-monitoring-fabric.

[8] S. Inc, "SevOne: The digital infrastructure management company". [Online]. Available: https://www.sevone.com/.

[9]P. Dauer, R. Khondoker, R. Marx, and K. Bayarou, "Security analysis of software defined networking applications for monitoring and measurement.

[10]N. A. Jagadeesan and B. Krishnamachari, "Software-defined networking paradigms in wireless networks: A survey.

[11]A. Shostack, "Experiences Threat Modeling at Microsoft", [Online]. http://ceur-ws.org/Vol-413/ paper12. pdf.

[12]Shawn Hernan and Scott Lambert and Tomasz Ostwald and Adam Shostack, Uncover Security Design Flaws Using The STRIDE Approach.

[13]sFlow.org, sFlow Version 5 Specification, [Online]. Available: http://www.sflow.org/sflow_version_5.txt.

[14]A. Zaalouk and R. Khondoker and R. Marx and K Bayarou, “OrchSec: An Orchestrator-Based Architecture for Enhancing Network-Security Using Network Monitoring and SDN Control Functions”