Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Securing Open Source SDN Controllers- A Comprehensive Review Using the STRIDE Model

Kamal Singh[1], Dr. Brijesh Kumar[2]

[1]MTEch, MSc.(IT), CCIE-SP and DC, MCT, FET, Manav Rachna International Institute of Research and Studies, India, kamallohiaji@gmail.com
[2]Dean Academics, Manav Rachna International Institute of Research and Studies, India, brijesh.fet@mriu.edu.in

Software-defined networking (SDN) can be susceptible to security vulnerabilities, leading to cyberattacks that can affect the entire network. This study aims to thoroughly examine the security of two commonly utilised open-source controllers, OpenDaylight (ODL) and Open Networking Operating System (ONOS), with a particular emphasis on significant security improvements and a comparison of their effectiveness in reducing potential risks. The study also analyses the security principles offered by the Open Networking Foundation (ONF). The comparison evaluates the compatibility of various security enhancements in both controllers with the STRIDE cybersecurity threat model. This model encompasses categories such as Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS), and Elevation of Privilege.

Keywords: SDN, SDN Security, OpenFlow, Network Security, Cyber Security, STRIDE, ONOS, ODL, Open Networking.

1. Introduction

SDN has evolved from innovative to well-established and advanced technology. Although the current era of networking provides numerous advantages to organisations, security remains a significant concern. Incorporating security protocols into SDN is still in the early phases of progress. Due to the growing number of cloud migrations and the rise of the Internet of Things (IoT), organisations encounter more significant difficulties safeguarding their networks.

The SDN architecture comprises various layers within its structure [1]. The SDN controller, an essential part of the control layer, is vital in the SDN architecture [2]. The SDN controller is responsible for configuring and managing the underlying network components. The security of the SDN controller is crucial in the context of SDN architecture [2]. If an SDN controller is compromised, it poses a significant risk to the entire network. Over the past six years, the ONOS and ODL controller communities have strongly emphasised security. They have been dedicated to developing a secure, robust, and resilient SDN controller. ONOS has made significant progress, with notable enhancements to its clustering capabilities and introducing a policy framework. Security is a top priority for ODL projects, as it is carefully integrated into the development process. These projects also ensure the inclusion of essential AAA functionality and strongly emphasise security through effective vulnerability management. This emphasises ODL's commitment to maintaining a secure environment.

According to the information in [3], ODL has a collective total of 22 vulnerabilities. Out of the core ODL code and applications, there are 16 distinct vulnerabilities. In addition, six security advisories have been issued regarding bugs and vulnerabilities discovered in third-party components utilised by ODL. OpenDaylight (ODL) had many reported and denial-of-service (DoS) vulnerabilities. These issues stem from excessive resource consumption. ONOS detected a total of twelve publicly known vulnerabilities. Similar to ODL, the vulnerabilities affect different components, with DoS attacks accounting for just under 50% of all recorded vulnerabilities. In addition, ONOS needs help ensuring precise authentication.

An analysis of the SDN controller is necessary to identify any vulnerabilities, as it serves as the central control node. This study will utilise ODL and ONOS for vulnerability classification, employing the Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS), and Elevation of Privilege (STRIDE) [4] threat model. Additionally, it will include an analysis of the security improvements made to both controllers and a comparison of various security principles utilising the ONF core security standards.

The structure of this work is as follows: Section 2 provides background information on SDN architecture and controllers. In Section 3, the literature review is presented. Section 4 discusses various threats and vulnerabilities following the STRIDE threat model. Section 5 discusses the security initiatives of the ODL and ONOS communities, secure controller architecture recommendations, and a comparison of ODL and ONOS security capabilities. Finally, concluding observations are presented in section 6.

2. BACKGROUND

2.1 SDN Architecture

SDN divides the control plane of a router from the data forwarding plane. The data plane forwards data packets)through the networking device in a traditional networking device, while the control plane makes routing decisions. SDN enables remote decision-making instead of relying on individual devices. Furthermore, the SDN's decoupling enables the new network functions and functionalities in the networking topologies.

The Open Networking Foundation (ONF) defines a three-tiered, high-level architecture for SDN:

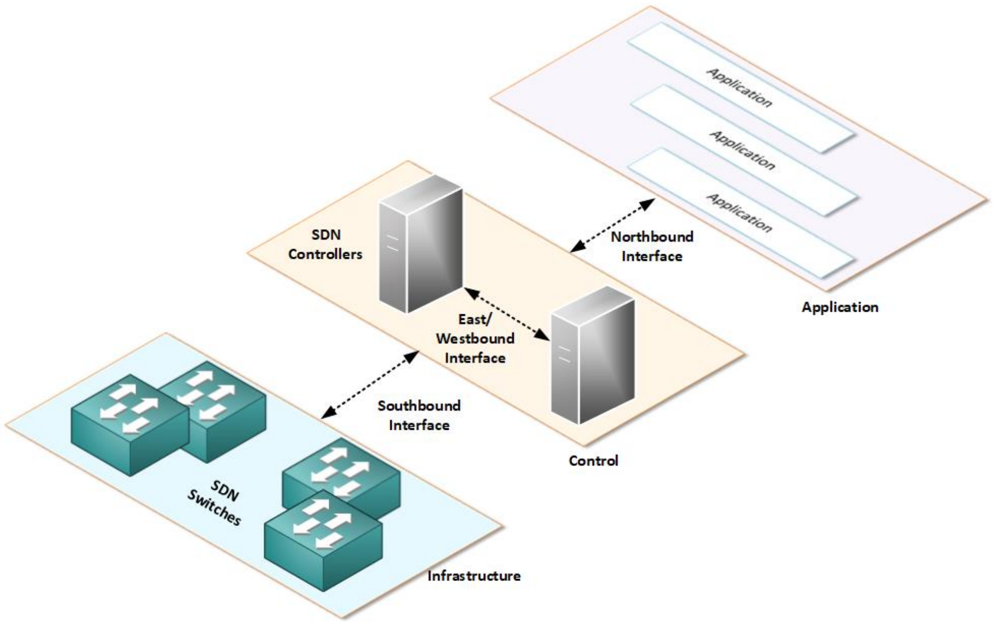

Figure 1: ONF SDN Architecture

Infrastructure Layer: This layer comprises physical and virtual network elements, such as switches, routers, gateways, and servers. These network elements are responsible for forwarding data packets.

Control Layer: The SDN controller is part of this layer and is responsible for managing and controlling the network components in the Infrastructure Layer. When the SDN controller communicates with network elements, it uses a southbound Application Programming Interface(API) like OpenFlow. It uses northbound APIs like Representational State Transfer (REST) when communicating with applications in the application layer.

Application Layer: This layer consists of the applications that use the network services provided by the SDN solution.

2.2 SDN Controller

The OpenFlow SDN controller manages and controls data traffic flow between network devices and is an essential element of an SDN network. ONOS and ODL are widely used open source controllers, and ODL is widely used as an open-source controller.

It encompasses essential components that contribute to its functionality:

Southbound API: The Southbound Application Application Programming Interface (API) is an interface between the OpenFlow controller and network devices, enabling efficient message exchange, traffic handling, and forwarding.

Northbound API: The Northbound API connects the OpenFlow controller to applications, allowing them to request network services like load balancing, security implementations, and Quality of Service (QoS) provisions.

East/Westbound API: East/Westbound APIs facilitate communication between SDN controllers and distributed controllers, promoting coordination and information exchange among network-distributed controllers.

Security Module: The OpenFlow SDN controller ensures network security by providing user authentication, encryption protocols, and access control functionalities to only authorised users.

Management Module: The management module configures and manages the OpenFlow controller, network devices, and applications, ensuring operational efficiency in the SDN ecosystem.

Open Network Operating System

The Open Network Operating System (ONOS) [5] is an open-source SDN controller platform developed by the Open Networking Foundation. ONOS has a modular, distributed architecture.

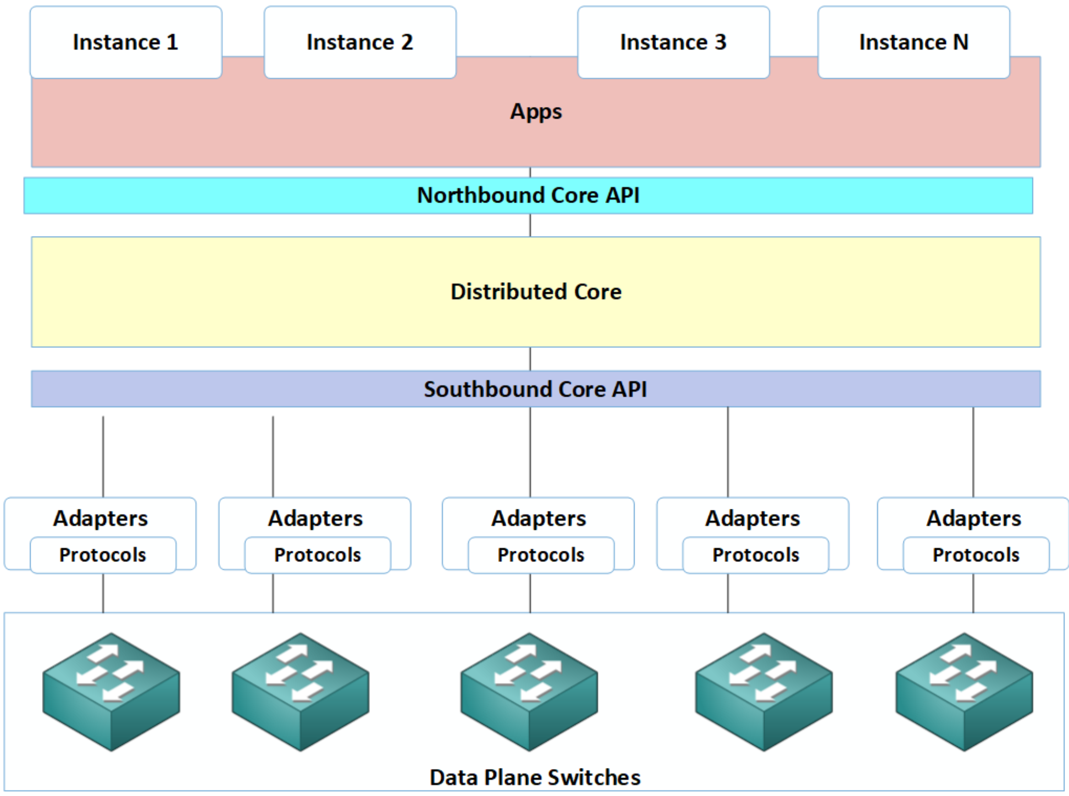

Figure 2: SDN ONOS Controller

ONOS uses microservices, which are modules for different functions and services. These modules can run on cluster nodes for flexibility and scalability. ONOS provides northbound and southbound interfaces for efficient network device and application communication. Communication between network switches and routers relies on the southbound interface.

OpenFlow and Network Configuration Protocol (NETCONF) protocols help it do this. The southbound interface establishes and maintains network device communication channels using OpenFlow and NETCONF protocols. The northbound interface connects network applications using REST APIs. ONOS provides comprehensive support for a variety of SDN capabilities. These include network topology detection, administration, setup, data flow control, load distribution, and segmentation. Virtualisation and multi-tenancy allow multiple users and applications to use a single network infrastructure. ONOS can run on physical servers, virtual machines, and cloud infrastructure. ONOS integrates seamlessly with Docker, Kubernetes, and Apache Mesos. Java was the primary programming language for the platform. Python and C++ were used to implement some components to improve its functionality. Version 2.0 of the Apache License governs ONOS. It debuted in December 2014.

OpenDaylight (ODL)

ODL is an open-source software-defined networking (SDN) controller platform introduced in 2013 [6]. The Linux Foundation manages the project, with contributors actively involved in its development.

Similar to ONOS, ODL also follows a distributed architecture, allowing for the deployment of functions and services as modules. It also facilitates communication between network devices and applications in both northbound and southbound directions with different protocols.

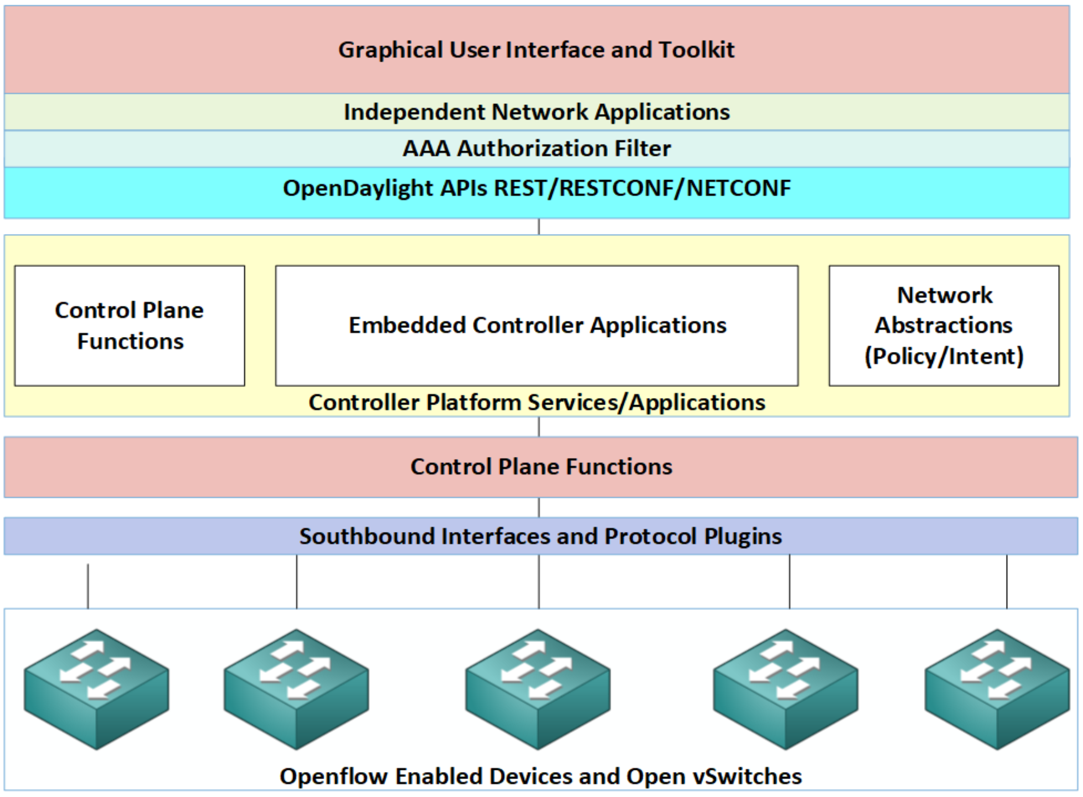
Graphical User Interface and Toolkit
Independent Network Applications
AAA Authorization Filter
OpenDaylight APIs REST/RESTCONF/NETCONF
Control Plane Functions
Embedded Controller Applications
Network Abstractions (Policy/Intent)
Controller Platform Services/Applications
Control Plane Functions
Southbound Interfaces and Protocol Plugins
Openflow Enabled Devices and Open vSwitches

Figure 3: ODL SDN Controller

ODL can be deployed on various infrastructure options, such as commodity servers, virtual machines, Docker containers, and Kubernetes clusters. The system incorporates various virtualisation and cloud technologies, including OpenStack and Amazon Web Services. The ODL framework is governed by the Eclipse Public License, version 1.0, a permissive open-source license. Several industry leaders, including Cisco, IBM, and Ericsson, back the ODL project.

3. LITERATURE REVIEW

The study [7] of ONOS and ODL, two open-source SDN controllers, aimed to uncover the progression of security measures for these platforms over time. Both communities demonstrated a strong dedication to security by consistently implementing software updates that included improved security features. Nevertheless, the study highlighted that security should have been prioritised regarding these controllers, as they only incorporated a few recommended secure features. The study highlighted the importance of conducting thorough security assessments on these controllers before they are deployed in public networks.

ODL experienced security concerns, including the vulnerability of the controller's login details being easily obtained and the potential for a server to be impacted by a Denial-of-Service (DOS) attack. The study [8] emphasised the importance of addressing these issues through updates. The necessary modifications were implemented, including adding encryption and the direct inclusion of Defense4All in the package instead of being optional. The study [9] investigated ODL and ONOS, two widely used SDN controllers. The authors utilised four machines with various configurations in their study. Open-source tools were utilised to launch Distributed Denial-of-Service (DDoS) traffic, which exposed the ONOS controllers and ODL-3 node cluster to various attack scenarios. The results revealed the susceptibility of SDN controllers to DDoS attacks. During attack scenarios, the ODL-3 node cluster outperformed ONOS regarding disk, memory, and CPU. Compared to the ONOS controller, the ODL-3 node cluster controller demonstrated a more extended operational period before encountering any failures.

Controller Dynamic Access Control (DAC) [10] protects SDN controllers using dynamic access control from API misuse. OpenDaylight can implement the controller-agnostic solution without modifying its source code. They showed that their prototype provides efficient and adaptable dynamic access control to improve SDN controller security, with a latency of less than 0.5% for API requests. The results show that the proposed approach protects SDN controllers from API misuse with minimal performance impact. The article [11] discussed SDN-enabled network security. They discussed their research on a complex and reliable SDN architecture that can simplify the controller's functionality and allow switches to make application-aware decisions. They strengthened the defensive data plane to improve SDN architecture security. Meta-data, memory, and advanced analysis improvements made the SDN architecture safer by strengthening the defensive data plane.

The authors considered ODL the best full-featured SDN controller due to its extensive range of applications and stable environment [12]. ODL surpassed other controllers in latency, detention, and unpleasantness for real-world events and simulations. To test Ubuntu 20.04.1

on Windows, the writers utilised VMware VirtualBox. Network managers can utilise different software and hardware to manage controllers and run various programs, enhancing the dynamic and lifelike nature of the results and making the research valuable for monitoring network performance. The ODL controller can also provide SDN network security, enabling firewall applications.

To address SDN security issues, DSF [13], a control plane framework for distributed SDN, uses data-centric Real-Time Publish/Subscribe (RTPS) communication. Control plane entities shared peer-to-peer or parent-to-child link discovery updates. The participants could route data plane packets between domains by synchronising their holistic topology. The Floodlight and ONOS controller platforms assessed the east/west interface in homogeneous and heterogeneous networks. Performance metrics showed consistent interface behaviour and real-time topology synchronisation as the number of controllers increased. In this research [14], the researchers secured the communication between the SDN controller and OpenFlow switch using IPsec. The authors developed a scalable and lightweight cryptographic service that utilises Free-to-Add (FTA) to invoke Internet Protocol Security (IPsec). They achieved a balance between communication performance and link security through feedback-based scheduling. This approach diminishes the effects of IPsec cryptography on network throughput and latency when network traffic fluctuates. This guarantees the efficient utilisation of system resources in order to process critical data.

SDN relies on network virtualisation; therefore, SDN hypervisor security assessment is essential. FlowVisor and OpenVirteX (OVX), two popular SDN hypervisor platforms, were analysed for security in this article [15]. Both hypervisors had new vulnerabilities that might let attackers damage networks. Due to the increasing importance of SDN hypervisors in large-scale networks, organisations should thoroughly test and analyse hypervisor code before deploying it in production systems.

4. ONOS AND ODL CONTROLLERS SECURITY FEATURES

4.1 ONOS and ODL Security Features

Mitre Common Vulnerabilities and Exposures (CVE) reports [16] and [17] determine that ONOS has approximately 41 instances of CVEs. The most recent report indicates that the OpenDaylight ODL platform has 20 CVEs.

The ONOS and ODL communities have taken many initiatives to add new security features. Below is a list of some security features:

Table 1: ONOS and ODL security features

Security Feature	SDN Controller	Description
Security-Mode ONOS [18]	ONOS	ONOS employs three access control mechanisms to enhance security: bundle-level role-based access control, application-level role-based access control, and API-level permission-based access control.
ONOS Access Control Based on Dynamic Host Configuration Protocol (DHCP) [19]	ONOS	It uses DHCP snooping to dynamically control network access based on DHCP messages exchanged between devices and DHCP servers.

ONOS Access Control List (ACL) [20]	ONOS	The ONOS platform incorporates a native application that facilitates the generation of access control lists by utilising 5-tuple rules to either permit or restrict IP traffic.
ONOS Authentication, authorisation, and accounting (AAA) [21]	ONOS	The RADIUS server analyses incoming traffic, enabling the blocking of unauthorised traffic or establishing flows on the switch to allow authenticated and authorised traffic.
Policy Framework for ONOS [22]	ONOS	This system offers network administrators a centralised approach to defining, applying, and enforcing network policies, thereby resolving policy conflicts.
ONOS ARTEMIS [23]	ONOS	A security application that uses machine learning algorithms and real-time data analysis to detect and mitigate Border Gateway Protocol (BGP) prefix hijacking automatically.
ODL Defense4All [24]	ODL	Designed to detect and mitigate DDoS attacks in ODL environments.
ODL Controller Shield [25]	ODL	It provides a secure framework for ODL controllers, protecting them from external attacks.
ODL Unified Secure Channel (USC) [26]	ODL	Enables secure communication between ODL and network devices using industry-standard Transport Layer Security (TLS).
ODL AAA [27]	ODL	Provides secure access control, ensuring only authorised users can access network resources.
Cardinal: ODL Monitoring as a Service [28]	ODL	Monitoring and analytics capabilities for ODL deployments, ensuring network performance and availability.
ODL Secure Network Bootstrapping Interface (SNBI) [29]	ODL	Secure discovery of controllers and network devices.

Security-Mode ONOS

ONOS (SM-ONOS) provides application authentication and access control mechanisms to protect sensitive information and prevent unauthorised access. It offers three tiers of access control: bundle-level role-based access control, application-level role-based access control, and API-level permission-based access control. Bundles are designated "applications" or "non-applications" and can access northbound API bundles and utility APIs. Application-level Role-Based Access Control (RBAC) is implemented by considering the policy file of an application bundle. An ONOS application with the " admin " role can access admin and regular services. In contrast, an application with the role of "non-admin" is limited to regular services. API-level permission-based access control uses permissions specified in the policy file, representing specific network operations and application programming interfaces.

ONOS Access Control Based on DHCP

The ONOS Access Control Based on Dynamic Host Configuration Protocol (DHCP) uses DHCP snooping to manage network access dynamically by leveraging DHCP messages exchanged between devices and DHCP servers. The ONOS application uses DHCP snooping to monitor DHCP messages exchanged between clients and servers, enabling customer identification. By default, every customer port is configured in a restricted state, allowing only DHCP traffic. The ONOS application uses DHCP snooping to observe shared DHCP messages between the client's device and the DHCP server. When a DHCP ACK is issued to a customer, the port is authorised, allowing unrestricted traffic. Access authorisation determines whether network resources are allowed or denied, granting complete access to resources. The ONOS application enforces access control policies using DHCP messages, including MAC address, IP address, and device lease duration, to allow or limit network access according to predetermined policies.

ONOS ACL

The ACL is an integrated feature within ONOS that facilitates the generation of access control lists by utilising 5-tuple rules to authorise or prohibit IP traffic. This application's integration allows for the seamless association of ACL rules with the corresponding network devices. Furthermore, the ACL application offers various services, including mapping deny rules to allow rules and determining the priority of newly added ACL rules.

ONOS AAA

The primary objective of the AAA application is to examine incoming traffic as the initial means of defense, utilising the RADIUS server to either block unauthorised traffic or allow authorised traffic to pass through the switch. Currently, the AAA application exclusively offers authentication services. Nevertheless, the ONOS community has stated it is a provisional resolution for future progress.

Policy Framework for ONOS

The policy framework for ONOS is a network policy framework explicitly designed for ONOS. The policy framework aims to address a fundamental security concern faced by SDN controllers: resolving policy conflicts. The mechanism lets administrators define, apply, and enforce network policies centrally. The framework has been specifically developed to be compatible with various policy languages and facilitate the management of networks based on policies. This, in turn, improves the network's overall security, reliability, and manageability. The Policy Framework for ONOS aims to possess extensibility and flexibility to cater to various policy use cases.

ARTEMIS

ONOS ARTEMIS uses advanced machine learning algorithms and real-time data analysis to find and stop BGP prefix hijacking. The system meticulously analyses BGP messages, identifying anomalies indicative of a potential hijacking attempt. When a potential hijack is detected, the system takes immediate proactive measures to mitigate the threat, such as alerting network administrators or filtering the identified malicious BGP messages. The objective of ONOS ARTEMIS is to furnish an automated and preemptive defense against BGP prefix hijacking, thereby minimising the likelihood of network disruptions and enhancing the overall security posture of the network.

ODL Defense4All

Defense4All is a solution designed to detect and counter DDoS attacks efficiently. The Defense4All application within ODL offers network administrators a comprehensive suite of security services. These services include monitoring the conduct of protected traffic and rerouting attacked traffic to designated Attack Mitigating Systems (AMSs). In pursuit of these objectives, the application strategically configures flow entries in specific network locations to monitor traffic statistics for each protected network segment (PN), initiating redirection to assigned AMSs upon detecting an attack.

Defense4All can establish communication with specified AMSs. This communication capability allows for dynamic configuration, continuous monitoring, and the comprehensive collection and utilisation of attack statistics from the AMSs. It is crucial to emphasise that the application programming interface for autonomous AMSs needs to be more standardised and fall beyond the scope of the OpenDaylight project. Defense4All incorporates a reference implementation pluggable driver to overcome this challenge, streamlining communication specifically with Radware's DefensePro AMS.

ODL Controller Shield

The ODL Controller Shield is a security solution designed to enhance the security of the ODL controller platform in SDN. It includes the Unified Security Plugin (USecPlugin), which provides security information to applications and helps with tasks like identifying attack sources and configuring firewalls. The Controller Shield also focuses on secure communication between the controller and SDN switches through encryption protocols.

ODL AAA

OpenDaylight's ODL AAA implementation provides a robust approach to user authentication and resource access control. Authentication involves entering a valid username and password with unique criteria for access. The AAA framework compares user credentials with database data, granting authorised access. Authorisation involves permissible actions for successful authentication, including tasks, APIs, and additional features. The implementation also includes an accounting process that records verified user actions, including data volume, API usage, and other details, for audits and network utilisation monitoring.

Cardinal - ODL Monitoring as a Service

The ODL’s Cardinal project provides a fault and health monitoring service for ODL and the underlying software-defined network. It enables remote monitoring through deployed Network Management Systems (NMS) or the analytics suite OpenDaylight MIB. The service allows ODL diagnostics and monitoring to be accessible through Simple Network Management Protocol (SNMP) version 2 and 3 and REST northbound interfaces. The ODL Cardinal enhances the overall health of the ODL System, provides information on Karaf parameters and features, improves the scalability of ODL plugins, and optimises network parameters. It facilitates autonomous notifications using SNMP traps, such as providing CPU and memory usage data for monitoring network attack-related resource utilisation. By utilising the monitoring capabilities of ODL Cardinal, security services could be expanded or merged to offer a more all-encompassing security solution for the software-defined network.

ODL Secure Network Bootstrapping Interface (SNBI)

The Secure Network Bootstrapping Interface (SNBi) within the ODL framework is a method that enables network devices to be securely bootstrapped without any manual intervention.

The utilisation of manufacturer-installed IEEE 802.1AR certificates is employed to ensure the security of initial communications between devices and controllers.

SNBi facilitates the automatic discovery of devices and controllers, unveiling the network's physical topology, revealing the various types of devices, and allocating a domain to each device. After the IP addresses are allocated to the devices and a secure IP connection is established, SNBi sets up a fundamental infrastructure to support various network functions on a network device. These functions include performance measurement, traffic-sniffing capability, and traffic transformation capability.

5. ONOS AND ODL CONTROLLERS SECURITY FEATURES REVIEW WITH STRIDE MODEL

5.1 SDN Threats Classification with STRIDE

The STRIDE threat model is a well-established and widely used cybersecurity framework that is valuable for identifying and categorising various security threats. With a comprehensive examination of the six distinct threat categories, specifically spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege, vulnerabilities can be systematically classified and subsequently leveraged to formulate robust strategies to mitigate the associated risks.

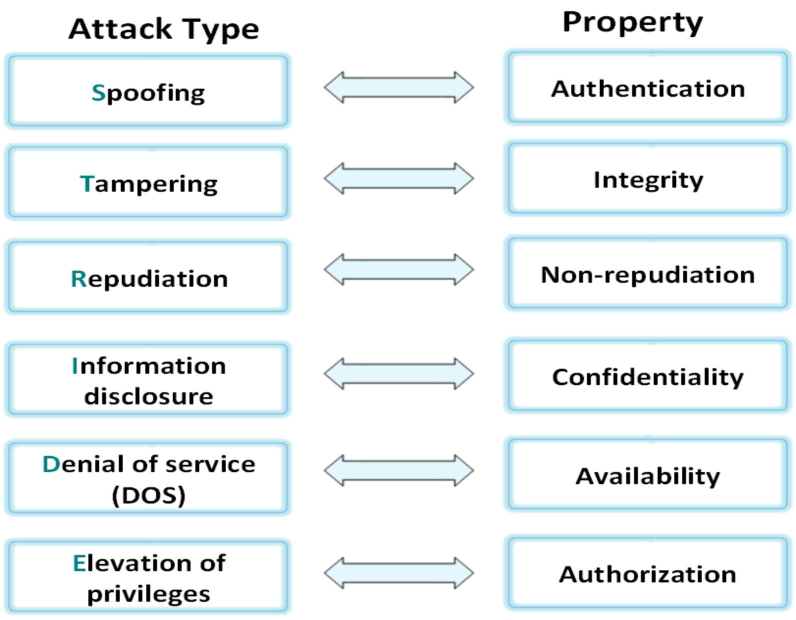

Figure 5: STRIDE attacks and security property [4]

Spoofing

Spoofing involves a deceitful act where a malicious actor imitates a legitimate user or system to obtain sensitive information or carry out harmful actions. In SDN, individuals with malicious objectives can manipulate packet information by using deceptive ports or system source addresses. This allows them to gain unauthorised access to the controller. Attackers can exploit this vulnerability to manipulate configuration settings, network management, topology, log files, backup flow table contents, and other vital data.

Tampering

Unauthorised modifications to data or code within a system constitute tampering; the resultant adverse effects include data loss, system instability, and security vulnerabilities. In the OpenFlow-based SDN domain, malicious actors can manipulate controller software, thereby obtaining access to update services or the ability to perform unauthorised package updates. This form of manipulation enables attackers to make unauthorised changes to critical elements, including logs, backup flow tables, policies, and configuration settings.

A significant concern arises when attackers manipulate controller policies to redirect network traffic, thus potentially exposing sensitive data through eavesdropping.

Repudiation

Repudiation occurs when a user declines to acknowledge or assume accountability for a specific action performed within a system. This behaviour may complicate identifying the origin of a security breach.

SDN controllers that occupy subordinate positions in a hierarchy, such as upstream/downstream controllers or applications, can launch attacks against the controller.

Information Disclosure Information disclosure

Information disclosure refers to the intentional or unintentional release of confidential data. Information disclosure is a risk associated with unauthorised access to controller data, which includes backup flow tables, configuration, and topology data. Protocol exchange can expose log data, statistics, and unsecured backup flow entries. Critical cryptography keys include the private key for the account digital certificate, the encryption key, and the root key. Disclosure of the encryption keys compromises cryptography. The identity of the controller is validated using their key. Generally, the transmission of private keys through messaging platforms is prohibited. Data leakage may occur due to sharing hardware and software resources, given that multiple applications utilise the same controller to store their data.

Denial of service

A Denial of Service (DoS) attack is a deliberate attempt to obstruct authorised users' access to a system or resource. This is typically accomplished through vulnerabilities or the overwhelming influx of requests onto the system. SDN involves the management of switches and applications by a central controller. Memory overload by an adversary may compromise the SDN controller's ability to allocate resources efficiently. Operating applications without appropriate access restrictions may prevent the Central Processing Unit (CPU) and other resources from overloading. Due to the unpredictability of network traffic in complex networks with numerous flows, packets that do not match flow table entries are forwarded to the controller to initiate new flows. DoS attacks may transpire when compromised switches overwhelm the controller with unsolicited packets and communication capacity may be impeded due to the intensive processing required for encryption and decryption, as well as the communication between the switches and the controller.

Elevation of privileges

Elevation of privileges occurs when attackers gain unauthorised access to privileged system resources, functionalities, or sensitive data within a system.

By utilising the API of SDN controllers, third-party applications can regulate their functionality. The API lacks privilege control, which allows malicious applications to exploit its weaknesses. This may grant unauthorised access to maintenance, services, system debugging, and other operational functions. SDN controllers facilitate the installation, testing, maintenance, debugging, and monitoring of applications developed by third parties. Nonetheless, API integration vulnerabilities may compromise the security of the entire controller.

SDN controller software is susceptible to design and code errors that may grant elevated privileges to malicious actors.

It is critical to implement privilege management within the controller's API in order to enhance SDN security. This requires the implementation of robust accounting, authentication, and authorisation procedures to prevent unauthorised access or actions.

5.2 SDN Controller Security Requirements Review with STRIDE

Secure controller design principles are essential to secure the entire SDN environment. The authors in [30] outlined eleven critical features of a secure network controller, which include control process isolation, policy conflict resolution, multiple instances of both the controller and applications for resilience, secure storage, secure communication interfaces, REST API security, integration with Intrusion Prevention System (IPS) for network security, authentication and authorisation for access control, resource monitoring, and logging/audit services for security. These features protect against single points of failure, prevent unauthorised access and resource consumption, and collect appropriate log information for security auditing.

In [31], the authors suggested nine rules for evaluating the security of five open-source and active SDN controllers: OpenDaylight, ONOS, Ryu, Floodlight, and OpenContrail.

The ONF proposed the following set of fundamental security requirements [32] for SDN controllers, consisting of 31 security parameters to secure the controller in light of the ONF's security evaluation of prospective threats. Using the STRIDE model, the security requirements are determined by thoroughly analysing potential threats to the SDN controller.

ONF has defined the security foundation requirements for controller security in [33]. The table below provides the security requirement comparison of two open SDN controllers:

Table 4: Security features comparison of ONOS and ODL with STRIDE

S. No.	Security Feature	STRIDE Category	ODL	ONOS
1	Authentication on Interfaces of SDN Controllers	Spoofing	✓	✓
2	IP check	Spoofing	✓	✓

3	User Authentication	Spoofing, Repudiation	✓	✓
4	Account Management	Spoofing		
5	Hardware Consistency	Spoofing, Information Disclosure, and Elevation of Privileges		
6	Hypervisor Security	Spoofing, Information Disclosure, and Elevation of Privileges		
7	Software package integrity	Tampering		
8	Protecting the Integrity of Data in Transit	Tampering	✓	✓
9	Protecting Reference Data from Unauthorised Modification	Tampering		
10	Log Function	Repudiation		
11	Log Files Access Protection	Repudiation	✓	✓
12	Log modification protection	Repudiation		
13	Authorisation for access to sensitive data	Information Disclosure		✓
14	Protecting the Confidentiality of Data in Transit	Information Disclosure	✓	✓
15	Hiding Password and Key Display	Information Disclosure		
16	Application Isolation	Information Disclosure		
17	Traffic separation	Information Disclosure		
18	Access control on the GUI	Information Disclosure	✓	✓
19	VM Security	Information Disclosure		
20	Closing unnecessary ports/services	Information Disclosure, Denial of Service		
21	Physical Host Security	Denial of Service		
22	Restriction for Forwarding Packets from Switches	Denial of Service		
23	Authorisation for flow table creation	Denial of Service		
24	Anti-DoS from Computing Capacity Exhaustion	Denial of Service		
25	Anti-DoS from Northbound/Southbound Interfaces	Denial of Service		
26	Anti-DoS from Excessive Resource Consumption	Denial of Service		
27	Privileged Control of Applications	Elevation of Privileges	✓	
28	Policy Conflict Resolution	Elevation of Privileges	✓	
29	Authorisation for Using System Functionalities	Elevation of Privileges		
30	Interface Authorisation for Third Parties	Elevation of Privileges		
31	Security of the hosting OS	Elevation of Privileges		

ODL and ONOS communities have improved their security features to combat modern cyber attacks. However, they do not only comply with some of the foundation security requirements compared with the STRIDE model. They still need to make more enhancements in their foundation summary for inbuilt security features for the secure controller framework for their products' ready commercial use.

The ODL and ONOS communities have significantly enhanced their security features to address modern cyber-attacks. However, it is worth noting that they still need to fully meet all the security requirements outlined by the STRIDE model. Further improvements are required in the foundation summary to incorporate enhanced security features into the secure controller framework, ensuring the product is ready for commercial use.

6. CONCLUSION

This paper explored the critical security features of ONOS and ODL SDN controllers. We delved into the STRIDE framework, mapping potential attacks and vulnerabilities within SDN controllers. Through STRIDE threat modelling, we scrutinised the security requirements outlined by ONF and assessed their coverage in ODL and ONOS security solutions. Despite commendable efforts to enhance security by ODL and ONOS, the two leading open-source SDN controllers, it is essential to note that they only partially encompass all the fundamental security principles necessary for a secure SDN environment. Notably, security is not a primary focus during the controller's design.

To fortify the SDN architecture against growing cyber threats, it is crucial to integrate third-party security services before deploying the controller in production networks. Collaborative community efforts should prioritise additional initiatives and incorporate enhanced security features into the controller. This proactive approach is essential for ensuring the security of the SDN architecture in the face of an escalating number of cyberattacks.

References

1. Internet Research Task Force SDN Layers. https://datatracker.ietf.org/doc/html/rfc7426. Accessed 6 Jan 2024
2. SDN Architecture Overview - Open Networking Foundation. https://opennetworking.org/wp-content/uploads/2013/02/SDN-architecture-overview-1.0.pdf. Accessed 6 Jan 2024
3. Security and performance comparison of ONOS and ODL controllers. https://opennetworking.org/wp-content/uploads/2019/09/ONOSvsODL-report-4.pdf. Accessed 6 Jan 2024
4. Jegeib Threats - Microsoft threat modeling tool - azure. Threats - Microsoft Threat Modeling Tool. https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats#stride-model. Accessed 6 Jan 2024
5. Open network operating system (ONOS) SDN Controller for SDN/NFV Solutions. Open Networking Foundation. https://opennetworking.org/onos/. Accessed 6 Jan 2024
6. OpenDaylight. https://www.opendaylight.org/. Accessed 6 Jan 2024
7. Scott-Hayward S (2017) Trailing the snail: SDN controller security evolution. arXiv.org. https://arxiv.org/abs/1711.08406. Accessed 6 Jan 2024

8. Unger W, Izzat Alsmadi (Mentor)Evaluating the security of software-defined network controllers. https://scholarworks.boisestate.edu/icur/2015/Poster_Session/145/. Accessed 6 Jan 2024
9. Badotra S, Tanwar S, Bharany S, et al (2022) A ddos vulnerability analysis system against distributed SDN controllers in a cloud computing environment. Electronics 11:3120. doi: 10.3390/electronics11193120
10. Tseng Y, Pattaranantakul M, He R, et al (2017) Controller DAC: Securing SDN Controller with Dynamic Access Control. 2017 IEEE International Conference on Communications (ICC). doi: 10.1109/icc.2017.7997249
11. Raghunath K, Krishnan P (2018) Towards a secure SDN Architecture. 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). doi: 10.1109/icccnt.2018.8494043
12. Gupta N, Maashi MS, Tanwar S, et al (2022) A comparative study of software defined networking controllers using mininet. Electronics 11:2715. doi: 10.3390/electronics11172715
13. Almadani B, Beg A, Mahmoud A (2021) DSF: A Distributed SDN Control Plane Framework for the east/west interface. IEEE Access 9:26735–26754. doi: 10.1109/access.2021.3057690
14. Yang X, Wang D, Tang W, et al (2020) IPsec Cryptographic Algorithm Invocation Considering Performance and security for SDN Southbound Interface Communication. IEEE Access 8:181782–181795. doi: 10.1109/access.2020.3028603
15. Alharbi T, Portmann M (2019) The (in)security of virtualisation in software defined networks. IEEE Access 7:66584–66594. doi: 10.1109/access.2019.2918101
16. Search results. MITRE ONOS CVE. https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=ONOS. Accessed 6 Jan 2024
17. Search results. MITRE ODL CVE. https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=OpenDaylight. Accessed 6 Jan 2024
18. Security-mode onos. Wiki. https://wiki.onosproject.org/display/ONOS/Security-Mode+ONOS. Accessed 6 Jan 2024
19. Access control based on DHCP. Wiki. https://wiki.onosproject.org/display/ONOS/Access+Control+Based+on+DHCP. Accessed 6 Jan 2024
20. Security and performance comparison of ONOS and ODL controllers. https://opennetworking.org/wp-content/uploads/2019/09/ONOSvsODL-report-4.pdf. Accessed 6 Jan 2024
21. Demo AAA/radius server testing. Wiki. https://wiki.onosproject.org/pages/viewpage.action?pageId=6357336. Accessed 6 Jan 2024
22. Policy Framework for ONOS. Wiki. https://wiki.onosproject.org/display/ONOS/POLICY+FRAMEWORK+FOR+ONOS. Accessed 6 Jan 2024
23. Artemis: Automatic and real-time detection and mitigation system against BGP prefix hijacking. Wiki. https://wiki.onosproject.org/pages/viewpage.action?pageId=13995462. Accessed 6 Jan 2024
24. defense4all. Defense4All - OpenDaylight User Guide. https://nexus.opendaylight.org/content/sites/site/org.opendaylight.docs/master/userguide/manuals/userguide/bk-user-guide/content/_defense4all.html. Accessed 6 Jan 2024
25. Controller shield. OpenDaylight. https://wiki.opendaylight.org/display/ODL/Controller+Shield. Accessed 6 Jan 2024
26. Unified secure channel. Unified Secure Channel - OpenDaylight Documentation Nitrogen documentation. https://test-odl-docs.readthedocs.io/en/latest/getting-started-guide/project-release-notes/usc.html. Accessed 6 Jan 2024
27. Authentication, authorisation and accounting (AAA) services . Authentication, Authorisation

and Accounting (AAA) Services - Developer guide - AAA master documentation. https://docs.opendaylight.org/projects/aaa/en/latest/dev-guide.html. Accessed 6 Jan 2024

28. Cardinal: OpenDaylight monitoring as a service. Cardinal: OpenDaylight Monitoring as a Service - OpenDaylight Documentation Nitrogen documentation. https://test-odl-docs.readthedocs.io/en/latest/user-guide/cardinal_-opendaylight-monitoring-as-a-service.html. Accessed 6 Jan 2024
29. Securenetworkbootstrapping. OpenDaylight. https://wiki.opendaylight.org/display/ODL/SecureNetworkBootstrapping. Accessed 6 Jan 2024
30. Scott-Hayward S (2015) Design and deployment of secure, robust, and Resilient SDN Controllers. Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft). doi: 10.1109/netsoft.2015.7258233
31. Tseng Y, Naït-Abdesselam F, Khokhar A (2018) A comprehensive 3-dimensional security analysis of a controller in software-defined networking. SECURITY AND PRIVACY. doi: 10.1002/spy2.21
32. Threat analysis for the SDN architecture - open networking foundation. https://opennetworking.org/wp-content/uploads/2014/10/Threat_Analysis_for_the_SDN_Architecture.pdf. Accessed 6 Jan 2024
33. Security Foundation Requirements for SDN controllers. https://opennetworking.org/wp-content/uploads/2013/05/Security_Foundation_Requirements_for_SDN_Controllers.pdf. Accessed 6 Jan 2024