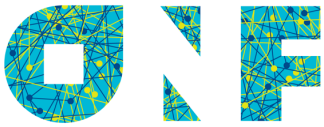

OPEN NETWORKING FOUNDATION

Principles and Practices for Securing Software-Defined Networks

January 2015

ONF TR-511

ONF Document Type: TR (Technical Recommendation)
ONF Document Name: Principles and Practices for Security Software-Defined Networks

Disclaimer

Open Networking Foundation
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303
www.opennetworking.org

Table of Contents

1 Introduction

Security challenges for Software-Defined Networks differ in some respects from those of a classical network due to the specific network implementation and SDN's inherent control and programmability characteristics. For instance, the concept of logically centralized control may expose a series of high-value assets to attackers while the ability to directly access the control plane results in a new attack surface (i.e. the Application-Control Programming Interface (A-CPI)) for adversaries.

For Software-Defined Networking (SDN), multiple vulnerability analyses have been performed [1-6], and several of these focus on the OpenFlow protocol. However, none of them attempts to extensively analyze the security issues with the SDN architecture and provide systematic methods to instruct the design of SDN solutions with the required security strength to tolerate threats. This is the intention of the Open Networking Foundation (ONF) security project. The project is initiated by defining a series of security principles that provide a reference point for the security work developed independently by different groups within the ONF. The application of these generic principles in the work proposed by ONF will ensure that ONF outputs have similar security features and sufficient capacity to deal with attacks arising in the operational environment.

1.1 Purpose:

Based on the unique SDN security challenges, the Open Networking Foundation (ONF) Security Discussion Group proposes a set of core security principles that provide criteria and instructions for designing and developing ONF specifications in which the security of the overall system is foundational. The principles are broadly defined and may cover different security issues depending on the context e.g., when securing an SDN protocol, an SDN component or an SDN interface.

In order to illustrate the application of these principles, this document presents a security analysis of a core protocol of the ONF, OpenFlow Switch Specification v1.3.4 [7]. As such, a set of security requirements is defined with respect to the security principles. The switch specification example then demonstrates how the security principles and requirements apply in a given situation, and how to use technology to support security. The recommended corrective measures are detailed in Section 5 with respect to the relevant security requirement.

1.2 Scope:

This document includes SDN security principles, security requirements for ONF protocols and a security analysis of the OpenFlow switch specification v1.3.4. Within the scope of the protocol-specification analysis, we determine potential vulnerabilities in the protocol and suggest updates to the specification.

1.3 Audience:

This document is conceived for use by ONF Projects in their design and development work. It is intended that individual ONF WGs will evaluate the security of their proposals and

specifications against the principles and requirements laid out within this document in order to deliver technologies that can be deployed and operated in a secure manner. It is anticipated that Section 5 of the document will be of particular interest to the ONF Extensibility WG for future versions of the OF Switch Specification.

Aspects of this document might also prove helpful in guiding member companies and SDN technology vendors in developing their platforms in a secure manner.

1.4 Document Structure:

The remainder of the document is organized as follows: Section 2 introduces the architecture of SDNs and the security challenges associated with this architecture. A set of SDN Security Principles are presented in Section 3 with security requirements derived from these principles detailed in Section 4. The OpenFlow Switch Specification v1.3.4 analysis is presented in Section 5. Section 6 concludes the document.

1.5 Terminology

1.5.1 Definitions:

Availability: The readiness for providing correct service to authorized parties.

Confidentiality: Limiting information access and disclosure to authorized parties.

Integrity: The trustworthiness of information resources.

Reference Data: The data objects that are related to state, configuration or status that are used by the logic of a security control.

Trust Boundary: The boundary of an area between components where the privilege level changes or where data is received from or sent to an untrusted or external source.

1.5.2 Abbreviations:

A-CPI	Application-Controller Plane Interface
BGP	Border Gateway Protocol
D-CPI	Data-Controller Plane Interface
DDoS	Distributed Denial-of-Service
DNS	Domain Name Server
DPID	Datapath ID
I-CPI	Intermediate-Controller Plane Interface
LAG	Link Aggregations
MAC	Medium Access Control
MITM	Man-in-the-Middle
NAT	Network Address Translation

QoS	Quality-of-Service
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTL	Time-to-Live

2 SDN Architecture and Security Challenges

In this section, we briefly introduce the SDN architecture and the security challenges associated with this architecture. For further detail regarding the SDN architecture, we refer the reader to [8].

2.1 SDN Architecture

The SDN model (Figure 1) proposed by the Architecture and Framework working group is composed of the application plane, the controller plane and the data plane [8]. A fundamental concept of the SDN architecture is the separation of the controller plane from the data plane. Network switches become simple forwarding devices and the control logic is implemented in a logically centralized controller (in practical implementation, the control function is distributed for resilience). The SDN controller controls data plane resources via D-CPI (Data-controller plane interface). A-CPI (Application-controller plane interface) is used to realize communication between applications and controllers, and management functions are orchestrated through the management interface. With programmability and flexibility, new algorithms and applications can be implemented and verified efficiently. This configuration also supports higher-layer applications that deal with multi-tenant issues.

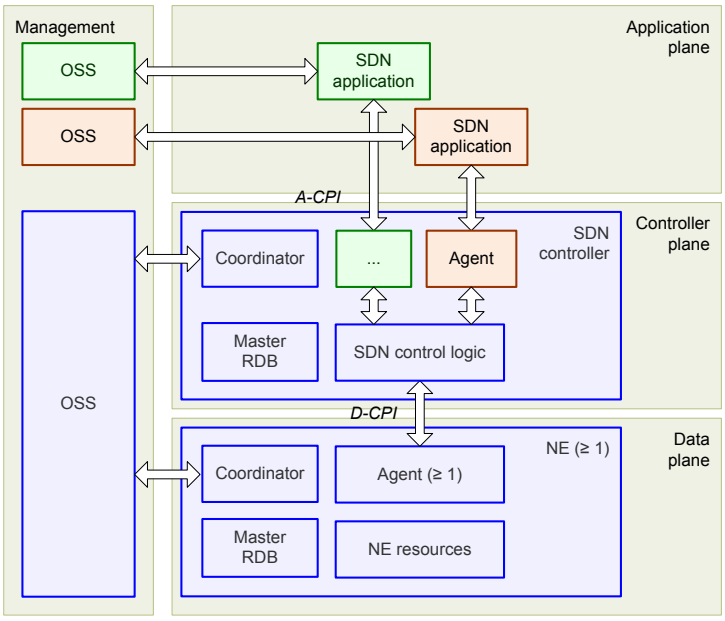

Figure 1: SDN Overview, with physical data plane [8]

These SDN features of programmability, flexibility and the support of 3rd party applications enable advanced networking functionality. However, they also introduce several new security issues.

2.2 SDN-Specific Security Challenges

New features and new network deployments can introduce faults and risks that open the door for threats that did not previously exist or are more serious than before. For example, in one configuration outlined in [8], one provider's SDN controller can directly access and manipulate another provider's SDN switches. This configuration is not recommended for deployment in practice. In addition to the traditional attack vectors on traffic flows, switches, administrative stations, and recovery and fault diagnosis, the controllers and the communications related to the Controller plane result in new security issues that are specific to SDN.

2.2.1 Centralized Control

Centralized control or logically centralized control (i.e. distributed but coordinated control function) exposes a high-value asset to attackers. Attackers may attempt to manipulate the common network services or even control the entire network by tricking or compromising a controller. This is distinct from a larger number of autonomous assets in a completely distributed control domain.

2.2.2 Programmability

New types of threats arise due to the explicit programmatic access SDN offers to clients that are typically separate organizational or business entities. This new business model presents requirements that do not exist within closed administrative domains in terms of protecting system integrity, third-party data and open interfaces.

2.2.2.1 Traffic and resource isolation

Operators must ensure that business management and real-time control information of one entity is fully isolated from that of all others. Best practices from existing automated interfaces between customer and provider business support systems may be of use here. This element extends to the existing security issue of multi-tenant traffic and resource isolation to avoid interference and misuse. Due to the new business model for SDN described above, there may be additional dynamic interactions introducing further requirements for isolation in order to meet different SLAs, private addressing issues, etc.

2.2.2.2 Trust between third party applications and the controller

Programmability is a double-edged sword; it offers flexibility to implement newly innovated market-driven applications but it also opens the door to malicious and vulnerable applications. Authentication and different authorization levels should be enforced at the point of application registration to the controller in order to limit the controller exposure.

2.2.2.3 Interface Security protection on A-CPI and I-CPI

Beyond the communication with applications through A-CPIs (see Figure 1), a controller may be controlled either by an upper layer controller or may work in tandem with another controller at the same hierarchical level. Lack of protection across these interfaces may lead to malicious attacks on the SDN. Security attributes and operation checkpoints should therefore be defined for securing A-CPIs and I-CPIs (Intermediate-controller plane interface).

2.2.3 Challenge of Integrating Legacy Protocols

SDN interfaces and protocols are being developed in the recognized context of escalating exploitation of technical and process deficiencies, with increasingly severe consequences that could lead to security issues. However, experience has demonstrated the difficulty of retrofitting security capabilities into existing technologies (Domain Name Server (DNS) and Border Gateway Protocol (BGP) are notable examples). It is critical that compatibility be checked before implementing legacy protocols (e.g., Network Address Translation (NAT), BGP) into SDN. It is also important that weaknesses previously addressed by legacy architectures not be repeated or even inflated when building the SDN framework.

2.2.4 Cross Domain Connection

An additional requirement of SDN implementation requires that infrastructure of different domains can be connected. This can be realized by connecting controllers of different providers via the I-CPI. The mechanisms to establish trust relationships, to determine authorization level in order to prevent abuse and secure channel setup should all be considered.

3 Security Principles

As detailed in Section 1, the 8 security principles outlined here apply to all protocols, components, and interfaces of the SDN architecture in Figure 1. In Section 4, these principles will be linked to the security requirements for SDN protocols.

3.1 Principle 1: Clearly Define Security Dependencies and Trust Boundaries

When specifying a security mechanism for SDN networks, security dependencies between different components must be clarified. Circular dependencies must be avoided. The clear definition of trust boundaries allows for targeted risk analysis and security control evaluation. Trust boundaries should be defined based on areas of privilege change, information flow across domains (i.e. ingress and egress direction), and dependency on data where confidentiality and integrity cannot be verified.

At a minimum, any external dependency should represent a trust boundary as it is reasonable to assume that attacks may arise from external systems. The interface to external environments should therefore provide sufficient security functionality to prevent or mitigate externally initiated attacks. External systems should be limited in access via a method of least privilege to reduce the risk to the system. In addition, the management or containment of internally initiated attacks should be considered to prevent impact on the external environment.

3.2 Principle 2: Assure Robust Identity

The basis for effective security is the ability to uniquely identify all components and users of a system and verify identities with a trusted source. Without a strong identity framework, the ability to build effective authentication, authorization, and accounting implementations will be limited.

A robust identity should have the following properties:

- Ability to distinguish its owner from other entities within a pre-defined scope.
- Ability to be generated, updated, and revoked.
- Impersonation prevention, preferably through strong cryptographic mechanisms.

Analysis of the SDN architecture identifies numerous means for elements inside the system's trust boundary to compromise the availability of the logically centralized control. Strong authentication based on assured identity is, therefore, critical to the security of the system.

There are several use cases for which elements external to the SDN system (e.g., network applications) will require access to a subset of system resources through defined interfaces. For such circumstances, access control mechanisms with various privilege levels should be employed to authorize external parties and authenticate their access to the system—e.g., role-based access control [9]. During communications, the identity of a device can be indicated explicitly by the information (e.g., identifiers, credentials, IP addresses, etc.) transferred with the packets, or implicitly by the key used to secure the packets.

3.3 Principle 3: Build Security based on Open Standards

Using open standards can bring benefits in both portability and interoperability. Wherever possible, proven protocols and methodologies should be implemented in favor of developing or designing new ones. New protocols and algorithms are created as a last resort when existing requirements cannot be met. For example, transport layer protection is required to secure the OpenFlow™ communication channel for both the Transmission Control Protocol (TCP) traffic header and payload. Various TCP enhancement techniques have been previously proposed for this purpose and are widely deployed [10]. It is, therefore, recommended to adopt such an existing technique rather than to develop a new transport layer solution.

The concept of protocol/algorithm reuse is particularly important in the case of security functionality such as encryption, authentication, and integrity, the solutions for which require significant vetting to prove their strength. Note that the use of legacy protocols or algorithms (e.g., MD5, Transport Layer Security (TLS) 1.0) that have been proved to be insecure and are no longer recommended by standards organizations should be avoided.

3.4 Principle 4: Protect the Information Security Triad

Although security controls by nature should increase the confidentiality, integrity, and availability (CIA) of a system, the security posture of the control should be evaluated for its impact on the overall architecture. An effective method for evaluating new controls is to determine whether the overall system availability might be reduced as a result. The control should not introduce new vulnerabilities or exploits.

Any reduction in the effectiveness of the core pillars (CIA) should be identified and mitigated. For example, the introduction of a centralized security server into the SDN architecture must be carefully evaluated in case the server's potential vulnerability to denial-of-service (DoS) attacks might impact system availability. If it might, then a suitable mitigation to this problem must be identified. In addition, security controls should be constructed in a way that they do not unnecessarily degrade system performance or impose additional system complexity which will likely introduce new security vulnerabilities. In practice, the eventual solution of a security control is synthetically affected by security requirements, cost, and manageability.

3.5 Principle 5: Protect Operational Reference Data

The effectiveness of a security control is directly impacted by the integrity of the reference data (e.g., credentials and sequence numbers), which is a key requirement in making operational decisions. Incorrect information can lead to unexpected system behavior that can result in a loss of confidentiality, integrity, and/or availability. In addition, the leakage of certain sensitive reference data such as cryptographic keys will cause potential security breaches of the security control. Operational reference data for all security controls should be clearly defined and protected to a level of continuity consistent with the security policy and the security architecture assumptions.

Reference data must be generated, processed, maintained, and transported securely in expected operational states, state transitions, and during the system lifecycle—i.e. system initialization, normal system operation, system standby, system failover and system recovery states, and during transitions between these states.

As an example, several security protocols use monotonically increasing sequence numbers to detect replay attacks. Any uncontrolled rollback of these numbers—particularly following system failure—must be avoided. This is of particular importance when automated key management is not supported.

3.6 Principle 6: Make Systems Secure by Default

Security controls should provide multiple security levels to meet the requirements of all potential system use cases. These levels may vary from a state in which a control is disabled to a state that can satisfy the most rigorous security requirements (e.g., deny by default). Regardless of the intended use case, the system should define a minimum level in which the majority of primary security controls are enabled by default. In addition to being enabled, these controls should be configured in a manner that meets minimum criteria to ensure that the control is effective. Security controls should have the ability to be reconfigured or even disabled, but this should be a conscious decision of the system owner/operator.

For example, when implementing an authentication control, it is important to ensure that there is some form of authentication by default. To make the control effective, the authentication should not be set to null or disabled entirely. Similarly, the key security properties (which could be various in different cases) of a system should be ensured across updates, recovery from failures, restarts, etc.

3.7 Principle 7: Provide Accountability and Traceability

All security controls should be auditable for the state and actions critical to system security. Logged data should contain sufficient information for auditing purposes. Based on the logged data, an auditor should be able to not only uniquely identify the entity on whose behalf an action has been carried out but also find out the relevant sequence of the action. Principle 2 aids in tracing the actions to particular entities.

However, it is also important to ensure that the audited data should not contain redundant information and the actions of auditing will not lead to violation of security policy.

The security properties of logged data should be protected to a level of continuity consistent with the security policy and assumptions during its lifecycle. Basically, the data should be protected against unauthorized access and modifications.

3.8 Principle 8: Properties of Manageable Security Controls

In addition to the seven principles specified above, when introducing new controls into an architecture or a standard, the following properties of the control should be considered:

- Prior to designing or introducing a security control, the security objectives and assumptions should be clarified;
- Security controls should be scalable and designed to support installations from the smallest reference system to the largest deployment without introducing undue complexity;
- When introducing new controls, the impact of the solution implementation and lifecycle management should be considered. New security functions should only introduce minimal complexity to the implementation. A good implementation should be extensible so that additional security control functions can be introduced in the future;
- Security controls should be easy to implement, maintain, and operate;
- Ensure that controls are backward-compatible, or provide an upgrade path that allows current and legacy controls to coexist;
- Ensure that controls are well documented and based on well-defined standards;
- It should always be possible to revoke and modify security credentials as part of a system's lifecycle;
- Wherever possible, all security controls should support automation to ensure that controls are properly implemented. In many cases, manual processes may lead to improper configuration, which may reduce the effectiveness of a control;
- The ability to monitor, troubleshoot, and debug any system is fundamental to its successful adoption.

4 Security Requirements for ONF Protocols

The principles described in Section 3 can be used to direct all of the security-related work in ONF. As such, they are defined in a relatively abstract way, i.e. a principle may cover different security issues when it is applied in different contexts. In this section, a set of security requirements are derived from each security principle introduced in the previous section. These security requirements specifically relate to the design and development of ONF protocols.

Goal: The security requirements are intended to help the designers of security mechanisms to:

- Address or mitigate the potential for malicious exploitation of ONF protocols; and
- Evaluate and control the negative effects (e.g., overheads, new security weaknesses) that may be introduced by the deployment of security mechanisms.

The following issues are out of scope:

- Security issues caused by improper implementation of security mechanisms;
- Security issues caused by the system configuration or operation in-service which is not in accordance with the system security recommendations;
- Physical attacks against SDN network assets (e.g., disabling network devices or breaking the cables connecting them).

4.1 Clearly Define Security Dependencies and Trust Boundaries

Before designing the security solution for a SDN protocol, the application scenarios in which the protocol will be used and potential threats associated with its use must be carefully analyzed. In each scenario, authentication and authorization must be performed between network elements on each side of the trust boundary before signaling packets are exchanged. In addition, packet level security protection must be provided for signaling packets.

REQ 4.1.1: The security solution of an SDN protocol should support mutual authentication between two SDN components running the protocol.

REQ 4.1.2: The security solution of an SDN protocol should provide the authorization function for the SDN components running the protocol in the case where an SDN component is only approved (based on certain security policies) to perform a limited set of operations on the resources of another SDN component.

REQ 4.1.3: The SDN protocol processing components should agree upon the security associations (e.g., key materials, algorithms etc.) for securing their communications before exchanging any protocol packets.

REQ 4.1.4: In the case that a protocol exchange could be accessed by an attacker, the security mechanism should be able to provide integrity protection (and optionally provide confidentiality protection) for protocol packets.

In practice, confidentiality protection can be optional and provided only when the protected content is sensitive.

4.2 Assure Robust Identity

REQ 4.2.1: Each entity (SDN devices or users) running the ONF protocol should have an ID that distinctly identifies the owner of the ID within a required scope. The possession of the identity should be verifiable through cryptographic methods during authentication.

REQ 4.2.2: In the protocol specification, the issues related to management of IDs during their lifecycle (including generation, distribution, maintenance, and revocation) should be considered.

It is not intended that a complete solution for ID management be specified in each protocol specification. However, ID management should be specified and provided as a fundamental service in ONF security solutions.

4.3 Build Security based on Open Standards

REQ 4.3.1: Existing security protocols/mechanisms should be applied first.

Security extensions to the base ONF protocols or new security protocols are proposed only when there is no existing security protocol meeting all the security requirements.

REQ 4.3.2: Non-standard or vulnerable algorithms/protocols should not be adopted.

Both MD5 and SHA-1 are now known to be vulnerable to collision attacks. These two algorithms are therefore not recommended for use in the security solutions proposed by ONF.

REQ 4.3.3: The policies of handling malformed or corrupted packets should be clearly specified.

Non-compliant packets or corrupted control messages should be handled correctly by the entities communicating via ONF protocol.

4.4 Protect the Information Security Triad

REQ 4.4.1: The security solution for an SDN protocol should consider the security issues raised in multiple layers.

For example, the packet headers and signaling messages of underlying transport protocols should be properly protected. BGP running over TLS does not solve the problem of an attacker being able to send a spoofed TCP FIN or TCP RST and causing the BGP session to go down.

REQ 4.4.2: The protocol specification should provide the mechanism to manage and rate control messages initiated by activity in the control/data plane in order to mitigate potential DoS/DDoS threats.

REQ 4.4.3: It is desirable for the SDN control protocol to be extensible to support additional signaling messages/options for dealing with future network attack types.

It is common that security mechanisms/extensions for a protocol are proposed after the publication of the base protocol. Therefore, it is desirable for

extensibility to be considered during the design of the base protocol such that the protocol can be extended for future security purposes.

REQ 4.4.4: A security protocol should be defined in such a way that each protocol message consists of sufficient information to instruct the message recipient(s) to correctly process it, e.g., being able to verify the integrity of the message.

This requirement is defined to avoid the case of a security mechanism being confused or overwhelmed by bogus packets. For example, when a security mechanism uses multiple keys to protect the communications between two network components, a key ID may need to be carried within a packet to indicate which key is used to verify the packet.

REQ 4.4.5: The amplification effect should be considered.

If a device has to generate a response that is much larger than the request, the device may be used by an attacker to perform reflection attacks. This issue can be mitigated by REQs 4.1.1, 4.1.2, and 4.4.2.

REQ 4.4.6: The proposed security mechanism should avoid the introduction of further, knock-on security issues.

For example, if the security solution for an ONF protocol introduces new centralized servers, it is necessary to identify how to protect them from becoming new attack targets (e.g., vulnerable to DDoS).

4.5 Protect Operational Reference Data

REQ 4.5.1: If the loss or improper/uncontrolled modification of certain reference data will result in potential security risks, such information should be securely maintained (e.g., integrity (and optionally confidentiality) protection applied when sensitive information is stored) and only be accessed by authorized entities.

In practice, such information normally includes access control policies, certificates, private keys, service descriptions and policy, etc. Note that sometimes the uncontrolled rollback of some data such as time and counters will result in security issues, e.g., Y2K.

4.6 Make Systems Secure by Default

REQ 4.6.1: The security solution for an ONF protocol may need to specify different default configuration and deployment plans for multiple application scenarios in order to ensure the security of network devices using the SDN protocol across updates, recovery from failures, restarts etc.

Such default configuration information may include default behavior, default algorithms, default key length, types of certificate, pre-defined access control policies, etc.

REQ 4.6.2: Mandatory cryptographic algorithms and security protocols should be specified.

4.7 Provide Accountability and Traceability

REQ 4.7.1: When designing an ONF protocol, critical events or incidents should be notified and logged for auditing purposes as well as reported to the required entities for reliability purposes.

REQ 4.7.2: All logging information from different SDN components should be securely stored (minimally with integrity protection). Confidentiality and integrity protection must be provided when the logs are transported to remote servers for analysis.

REQ 4.7.3: Critical status and counters for different SDN components must be logged for monitoring purposes. Those logs must be regularly monitored in order to detect malicious activities in regards to different SDN components

4.8 Properties of Manageable Security Controls

In addition to the requirements introduced above, a well-designed security mechanism for ONF protocols should also adhere to the following requirements.

REQ 4.8.1: The security mechanism should be able to support various security algorithms so that a user can select their preferred algorithm to secure the system.

This requirement can be used to enable users to select different levels of security protection according to different security objectives.

REQ 4.8.2: The security mechanism should be extensible and support introducing new algorithms or new security functionalities when necessary.

REQ 4.8.3: A security mechanism should be able to support automated key/credential management and consider the issues with generation, distribution, and revocation of security credentials.

Key management is closely related with ID management. See also 4.2.2. This requirement does not preclude the usage of manual key management (though not recommended).

5 OpenFlow Switch Specification v1.3.4 Security Analysis

In Section 4, the requirements for achieving a secure baseline in the design and development of ONF protocols are defined. In order to illustrate how the security principles and the requirements defined in Sections 3 and 4 can be applied to enhance the security of an SDN protocol, in this section we present a security analysis of the OpenFlow Switch Specification v1.3.4 [7] from a protocol perspective. Note: The intention is not to replace a threat analysis. This approach is complementary to threat analysis.

We first introduce the attack model related to this analysis.

5.1 Attack Model

5.1.1 Actors

The threats/attacks against the OpenFlow protocol can be classified into two categories: externally initiated or internally initiated. An internal actor has obtained the privileges to modify the OpenFlow protocol implementation or access the related reference data, and then attempts to abuse those privileges to perform the malicious activities from inside the system security boundary. In contrast, the external actor has no such privileges. In this analysis, it is assumed that an external actor is in control of a computing device directly or indirectly attached to the data plane of the SDN devices running OpenFlow, and possesses the tools used to generate valid/invalid types of traffic.

5.1.2 Considered Vectors for Security Breach

In an externally initiated attack, an actor may be able to:

- Passively eavesdrop on the data/control messages. With this type of attack, an actor may be able to gather intelligence useful for subsequent analysis or attacks (e.g., social engineering).
- Perform Man-in-the-Middle (MITM) attacks, DoS/DDoS attacks, or side-channel attacks by replaying data/control messages or inject non-authentic data/control messages into the SDN network.

Besides the above attacks, it is assumed that in an internally initiated attack the actor is also able to use the resources/privilege they hold to perform more sophisticated attacks. For instance, an internal attacker may use the credentials it holds to impersonate another legal controller to communicate with a switch. It is extremely difficult (if possible) for security mechanisms to prevent or detect every type of internally initiated attack, in isolation. However, a well-designed security mechanism with proper authentication, authorization, and logging can effectively identify, confine, and mitigate the potential damage from internal threat actors.

5.1.3 Assets

The assets/properties that the security mechanism for the OpenFlow protocol attempts to protect include:

- Sensitive information transferred within the protocol messages;
- Reference data for OpenFlow instantiation or other reference data for the devices implementing OpenFlow which may be affected by the protocol instantiation (e.g., switch flow table entries);
- SDN network availability and performance information and tenant network information and topology; and
- Resources in the control and data planes (e.g., bandwidth and latency between the involved SDN components).

5.2 Protocol-Specification Analysis:

The analysis is presented according to the flow of the OFv1.3.4 specification [7].
This breakdown is illustrated in Table 1. Only sub-components/scenarios for which an issue has been identified are included in Table 1. In each sub-section, considerations are identified per sub-component/scenario. A table of issues, attacks, countermeasures and relevant requirements is presented per component.

Table 1: OF v1.3.4 Analysis Breakdown

Entity	Components	Sub-components/Scenarios
Switch	Ports	• Physical Ports • Logical Ports • Reserved Ports
	Tables	• Counters
OpenFlow Channel and Control Channel	Channel Connections	• Connection Setup • Connection Interruption • Encryption • Multiple Controllers • Auxiliary Connections

Assumptions:

- While not a strict requirement, it is assumed that a secure version of TLS (e.g., v1.2) (or a TLS equivalent protocol, e.g., DTLS for securing messages over UDP) is implemented between the switch and controller to deal with tampering with message exchanges (insert/delete/modify) and to perform mutual authentication.

- This analysis assumes that each OF switch is connected to one or more controllers within the trust boundary of a single service provider. Issues of inter-boundary trust are outside the scope of this analysis at this time.

5.2.1 OpenFlow Switch

Table 2 details the issues identified when analyzing the OpenFlow Switch components of the OFv1.3.4 specification against the security principles and requirements defined in Sections 3 and 4.

Table 2: Issues, Countermeasures and Principles/Requirements for OpenFlow Switches

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Physical Ports	A physical device can be inserted or changed on the far end in order to perform traffic monitoring perhaps leading to a network attack.	Enable the controller to notice the modifications of far-end MAC Addresses and other link layer states.	Principle 4: Protect the Information Security Triad REQ 4.4.1 Information Disclosure
Logical Ports	Tunnel ID is not provided in Port Statistics messages	Enable the controller to learn the tunnel IDs associated with logical ports	Principle 4: Protect the Information Security Triad REQ 4.4.1 Information Disclosure
Reserved Ports	No way for applications to collect the statistical information of reserved ports (except LOCAL)	Enable the controller to learn such information of reserved ports	Principle 7: Provide Accountability and Traceability REQ 4.7.1 and 4.7.2
Counters	Roll-back of counters is out of control	Discuss how such conditions will not cause inconsistencies	Principle 5: Protect Operational Reference Data REQ 4.5.1
Matching	No specification for handling malformed packets	Any non-compliant incoming packet (IEEE and/or RFC specification) should be dropped by the switch/controller. In addition, a mechanism to check malformed or corrupt OpenFlow control packets should be implemented and strictly enforced in the switch/controller.	Principle 3: Build Security based on Open Standards REQ 4.3.3.
Flow Removal	Inconsistent flow table view at the controller	Any changes to the forwarding state (particularly flow removal initiated by non-master controller) in the switch must be communicated/notified to the controller. This ensures that the controller and switch have a consistent view of the forwarding topology.	Principle 5: Protect Accountability and Traceability REQ 4.7.1

5.2.2 OpenFlow Channel and Control Channel

Table 3 lists the issues identified when analyzing the OpenFlow Channel and Control Channel of the OFv1.3.4 specification against the security principles and requirements defined in Sections 3 and 4.

Table 3: Issues, Countermeasures and Principles/Requirements for OpenFlow Channels

Section	Potential Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Connection Setup	No information provided on TLS usage	Clarification on TLS usage should be provided or a pointer to specification in a companion protocol.	Principle 6: Make Systems Secure by Default REQ 4.6.1.
	TLS does not provide protection of TCP headers.	Security mechanisms such as TCP-AO that provide protection to TCP headers could be considered.	Principle 3: Build Security based on Open Standards REQ 4.3.1.
	No information on managing credential details (keys, certificates)	Credentials should be configured and managed by a switch management protocol like OF-Config. A pointer in OF protocol is required	Principle 5: Protect Operational Reference Data REQ 4.5.1. Principle 8: Properties of Manageable Security Controls REQ 4.8.3.
Connection Interruption	Potential for reduced security level following connection interruption.	Same level of security should be maintained before and after the connection interruption. The controller should be notified of the switches current state after reconnection. In this case, a message should be generated to the controller following any transition in mode of operation (from “fail-standalone mode” to “fail-secure mode”).	Principle 6: Make Systems Secure by Default REQ 4.6.1.
Encryption	Only authentication using certificates is discussed, this implies the exclusion of message authentication based on pre-shared key	Add statements regarding support for multiple types of authentication mechanism	Principle 8: Properties of Manageable Security Controls REQ 4.8.3

	Fail to discuss the cases where only integrity protection is provided.	Message integrity protection should be supported when the information transported over the OpenFlow messages is not sensitive.	Principle 4:Protect the Information Security Triad REQ 4.4.4
Multiple Controllers	Potential for conflict between multiple controllers with Equal role.	Employ policy conflict resolution mechanisms at the controller or add additional flags in the specification to detect conflict flows like CHECK_OVERLAP	Principle 8: Properties of Manageable Security Controls REQ 4.8.2.
	Fingerprinting is possible by Asynchronous messages being sent to all attached controllers.	Mutual authentication between controllers and switches is required regardless of controller role.	Principle 1: Clearly define Security Dependencies and Trust Boundaries REQ 4.1.1.
	Malicious controller requests role change to Master, demoting the legitimate controller to Slave.	A message should be sent to the Master controller to identify a role change. A message should be sent to all controllers upon new controller connection.	Principle 4: Protect the Information Security Triad REQ 4.4.3.
	Unauthorized access or manipulation of controller connection role.	Secure switch storage of controller connection information.	Principle 5: Protect Operational Reference Data REQ 4.5.1.
	Ambiguous role status event notification	Role Status Event - Reason should identify which controller requested the role change and report that in the reason to other controllers whose role is changed; "Another controller asked to be master" is ambiguous.	Principle 7: Protect Accountability and Traceability REQ 4.7.1
Auxiliary Connections	Lack of notification when receiving an invalid DPID	An error message should be generated for an incoming packet with an invalid DPID.	Principle 7: Protect Accountability and Traceability REQ 4.7.1

	When a key is used to protect different channels, the compromise of one channel may result in the compromise of others.	Different keys should be used for each connection (main and auxiliary).	Principle 4: Protect the Information Security Triad REQ 4.1.3

5.2.3 Additional Issues

There are two assumptions for the analysis in Sections 5.2.1 and 5.2.2. However, the assumptions are not mandated in the OpenFlow specification. If an assumption is not satisfied, additional issues may need to be considered. Table 4 lists some of issues arising when no security protection is provided for OpenFlow connections.

Table 4: Issues, Countermeasures and Principles/Requirements for no TLS on D-CPI

Section	Issue	Potential/Candidate Countermeasure	Security Principle/Requirement
Multiple Controllers	Fingerprinting is possible by Asynchronous messages being sent to all attached controllers.	Mutual authentication between controllers and switches is required regardless of controller role.	Principle 1: Clearly define Security Dependencies and Trust Boundaries REQ 4.1.1.
	Integrity of role request messages.	Use secure channel communication.	Principle 1: Clearly define Security Dependencies and Trust Boundaries REQ 4.1.4.
Auxiliary Connections	Manipulation of Controller role information across an insecure auxiliary connection.	All controller-switch connections (auxiliary and main) should use secure channel communication.	Principle 1: Clearly define Security Dependencies and Trust Boundaries REQ 4.1.4.
	If the Datapath ID and auxiliary ID are not sufficiently random, an attacker may perform offline attacks on the auxiliary connections over UDP.	Extend the ID to 96 bits. The lower 48 bits are the switch MAC address, while the top 48 bits are randomly generated.	Principle 2: Assure Robust Identity REQ 4.2.1

5.3 Summary of Recommendations

Tables 2, 3, and 4 present the countermeasures or recommendations generated based on our analysis of the OpenFlow Switch Specification v1.3.4.

In this section, the recommendations are separated into two strands; (1) securing the OpenFlow protocol itself, and (2) securing the data plane. For (1), we present our recommendations as OpenFlow bugs to be fixed. For (2), we propose additional features which do not directly benefit the security of OpenFlow communications but could be used to enhance the capability of Network components to deal with attacks on the Data Plane.

5.3.1 Securing the OF Protocol:

5.3.1.1 Use and specification of TLS

Issue: The use of TLS is currently under-specified in the document.

- No information on TLS version or usage information
- Need clear specification on credential management

Recommendations:

- The specification should recommend/state the use of a secure version of TLS (i.e., 1.2 or greater) or a TLS equivalent protocol (i.e., DTLS for securing messages over UDP) for auxiliary connections.
- While the use of plain TCP is understandable, the specification should explicitly callout and recommend the use of TLS for all connections
- Include the recommended mandatory cipher suite to be supported by OpenFlow switches: **TLS_RSA_WITH_AES_256_CBC_SHA256**.
- Ability to configure different cipher settings
- Key management requirements: For instance, Different keys should be used for each connection (main and auxiliary).
- In consideration of the above points, we recommend the specification to provide a pointer to configuration protocols ***"It is recommended to configure and manage security credentials (cipher settings and certificates) using a switch management protocol like the OF-Configuration protocol"***

5.3.1.2 Connection Interruption Issues

Issue:

- No notification for a transition in mode of operation (from "fail-standalone mode" to "fail-secure mode")
- Potential for reduced security level following connection interruption.

Recommendations:

Same level of security should be maintained before and after the connection interruption.

- A message should be generated to the controller following any transition in the switch mode of operation. This also helps in deciding if the controller should read all flow entries after re-connection.
- The mode of operation can be sent as part of switch feature reply or switch configuration message in OpenFlow.

5.3.1.3 Multiple Controllers: Role Change and Status

Note: This analysis is based on Specification v1.4 as a result of several updates to that section.

Issue: Section 6.3.5

- *"When a controller changes its role to OFPCR_ROLE_MASTER, the switch changes **all other controllers with the role OFPCR_ROLE_MASTER** to have the role OFPCR_ROLE_SLAVE, but does not affect controllers with role OFPCR_ROLE_EQUAL".* There can be only 1 Master Controller. The text should be changed to reflect this.
- *"When the switch performs such role changes, if a controller role is changed from OFPCR_ROLE_MASTER to **OFPCR_ROLE_SLAVE**, the switch must generate a controller role status event for this controller informing it of its new state".* The switch must notify role status event when a controller role is changed to either SLAVE or EQUAL

Recommendations:

- A message should be sent to all controllers upon new controller connection.
- 7.4.4: Role Status Event message: Reason should include some form of information to indicate which controller initiated the request rather than sending an ambiguous reason e.g., "Another controller asked to be master

5.3.1.4 Additional Recommendations for Securing OpenFlow:

- Counter updates to the controller should be set at pre-defined intervals and with acknowledgment. The rollover of counters that may cause potential inconsistency needs to be controlled.
- An error message should be generated for an incoming packet with an invalid DPID.
- A mechanism to check malformed or corrupted OpenFlow control packets should be implemented and strictly enforced in the switch and all controllers.
- For stronger security guarantees: Consider the possibility of using a security protocol which could protect the TCP headers (e.g., TCP-AO)
- The controller should acknowledge flow removal messages from the switch

5.3.2 Securing the Data Plane:

- MAC Address modification may be reported.
- The controller should be able to learn the tunnel IDs associated with logical ports.
- The controller should periodically collect the statistical information of ports.
- State transition of all SDN components should be logged.
- All logged information should be protected.
- Design flow-control mechanism to assure reliable updates and communications between controllers and switches (more research needed)
- Enforce message validation and integrity to avoid unintended consequences of misconfiguration of instantiation of corrupt table entries
- Implement a PKI CA to manage trust, authenticity, revocation and repudiation
- Ensure authenticity of communications endpoints within the OF SDN fabric (802.1x)
- Employ policy conflict resolution mechanisms at the controller.
- Secure switch storage of controller connection information.

6 Summary

Programmability can provide opportunities to enhance the security posture of networks. For example, it may be possible to use SDN techniques to construct a security solution that is able to coordinate both network and security devices to detect and react to attacks in a more flexible manner. However, the implementation of new network security functionality should not be achieved at the expense of overall system integrity and security

The objective of this document is to present a set of high-level security principles that should be applied to ensure that products based on ONF-developed standards and architectures can be implemented in a consistent, fundamentally secure manner. This is a foundational work of ONF security. In order to illustrate the implementation of these principles in design and development, this document presents a set of security requirements associated with the individual security principles but specifically applied to securing SDN protocols. Finally, a set of recommended corrective measures for the OpenFlow v1.3.4 protocol has been identified based on the detailed security requirements.

This work is the first document in the work plan of the security project. In the future, it is proposed to detail security requirements for further elements of the SDN architecture.

7 References

[1] R. Kloeti, "OpenFlow: A Security Analysis," April 2013. [Online]. Available: ftp://yosemite.ee.ethz.ch/pub/students/2012-HS/MA-2012-20 signed.pdf

[2] K. Benton, L. J. Camp, and C. Small, "OpenFlow Vulnerability Assessment," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013, pp. 151–152.

[3] D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013, pp. 55–60.

[4] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN Security: A Survey," in IEEE SDN for Future Networks and Services (SDN4FNS), 2013, pp. 1–7.

[5] V. T. Costa and L. H. M. K. Costa, "Vulnerability Study of FlowVisor-based Virtualized Network Environments," in 2nd Workshop on Network Virtualization and Intelligence for the Future Internet, 2013

[6] D. Li, X. Hong, and J. Bowman, "Evaluation of Security Vulnerabilities by Using ProtoGENI as a Launchpad," in Global Telecommunications Conference (GLOBECOM 2011). IEEE, 2011, pp. 1–6.

[7] "OpenFlow Switch Specification Version 1.3.4," Open Networking Foundation. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.3.4.pdf

[8] "SDN Architecture (Issue 1)," June, 2014. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR SDN ARCH 1.0 06062014.pdf

[9] Ferraiolo, David F., and D. Richard Kuhn. "Role-based access controls." arXiv preprint arXiv:0903.2171 (2009)

[10] The TCP Authentication Option (RFC 5925); Improving TCP's Robustness to Blind In-Window Attacks (RFC 5961); Recommendations for Transport-Protocol Port Randomization (RFC 6056); TLS (RFC 4346).

[11] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, "Frenetic: A network programming language," ACM SIGPLAN Notices, vol. 46, no. 9, pp. 279–291, 2011.

[12] A. Khurshid, W. Zhou, M. Caesar, and P. Godfrey, "VeriFlow: Verifying network-wide invariants in real time," ACM SIGCOMM Computer Communication Review, vol. 42, no. 4, pp. 467–472, 2012.

[13] P. Kazemian, M. Chang, H. Zeng, G. Varghese, N. McKeown, and S. Whyte, "Real time network policy checking using header space analysis," in USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2013.

LIST OF CONTRIBUTORS

Todd Aven (Goldman Sachs),

Paul Barret (NetScout Systems),

Craig Callahan (Goldman Sachs),

Danping He (Huawei),

Dave Hood (Ericsson)

Aubrey Merchant-Dest (Blue Coat),

Sriram Natarajan (Deutsche Telekom),

Makan Pourzandi (Ericsson),

Sandra Scott-Hayward (Queen's University Belfast),

Guo Shu (China Mobile)

Zhou Sujing (ZTE),

Marc Woolward (Goldman Sachs),

Dacheng Zhang （Alibaba）

