Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 19, 2013

M. Wasserman
S. Hartman
Painless Security

April 17, 2013

Security Analysis of the Open Networking Foundation (ONF) OpenFlow Switch Specification

draft-mrw-sdnsec-openflow-analysis-02

Abstract

This document discusses the security properties of the OpenFlow Switch Specification version 1.3.0 (OpenFlow), a Software-Defined Network (SDN) solution produced by the Open Networking Foundation (ONF). It analyzes the suitability of OpenFlow for use in "the cloud" or on the open Internet. It also makes some suggestions about how OpenFlow could be made more secure for use in those environments.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. OpenFlow Security Features 2
3. Specification Issues with OpenFlow Security 3
3.1. Optional Security, Failure Path Unspecified 3
3.2. No Certificate Details 3
3.3. Importance of Checking Certificates on Both Sides 4
3.4. TLS 1.0 Vulnerabilities 4
3.5. Failure Handling Solutions 4
3.6. Solutions to Keep the Consistancy of Flow Policies . . . 4
4. Applicability of OpenFlow Security Mechanisms 5
4.1. One Controller per Switch Scenario 5
4.2. Security in Other Scenarios 5
4.2.1. Multiple Controllers per Switch 5
4.2.2. Multiple Apps Talking to a Central Controller 6
5. Suggestions for Future Work 6
6. Security Considerations 6
7. Acknowledgements 6
8. Informative References 6
Authors' Addresses 7

1. Introduction

This document discusses the security properties of the OpenFlow Switch Specificaiton (OpenFlow) version 1.3.0 [ref], a Software-Defined Network (SDN) solution produced by the Open Networking Foundation (ONF). It analyzes the suitability of OpenFlow for use in "the cloud" or on the open Internet. It also makes some suggestions about how OpenFlow could be made more secure for use in those environments.

TBD: Add a short overview of OpenFlow.

2. OpenFlow Security Features

To quote the OpenFlow specification, "The switch and controller may communicate through a TLS connection. The TLS connection is initiated by the switch on startup to the controller, which is located by default on TCP port 6633 . The switch and controller mutually authenticate by exchanging certificates signed by a site-specific private key. Each switch must be user-configurable with one certificate for authenticating the controller (controller

certificate) and the other for authenticating to the controller (switch certificate)."

In other words, OpenFlow includes an optional security feature that allows the use of TLS on an OpenFlow control channel. This mechanism provides for authentication of the switch and the controller (if certificates are properly checked in both directions) to prevent attackers from impersonating a switch or a controller. It also allows for encryption of the control channel to prevent eavesdropping.

The OpenFlow specification mentions that auxiliary connections can use UDP with DTLS, but there is no further discussion of how DTLS will be used. Presumably it would use the same certificate-based authentication as is described for connections using TCP and TLS.

3. Specification Issues with OpenFlow Security

OpenFlow security is minimally specified, to the point where the differences between multiple OpenFlow implementations could cause operational complexity, interoperability issues or unexpected security vulnerabilities. This section outlines some of the issues in the OpenFlow specification that it might be useful to address in a later version of the specification.

3.1. Optional Security, Failure Path Unspecified

OpenFlow security is optional, requiring that implementations include support for non-secure control connections to ensure interoperability. Furthermore, there is no indication about whether implementations should fall back to insecure operation if authentication fails, or whether the connection should be closed after an authentication failure.

Also, as the OpenFlow specification states, "when using plain TCP, it is recommended to use alternative security measures to prevent eavesdropping, controller impersonation or other attacks on the OpenFlow channel." However, the specification gives no indication of what sort of alternative security measures are needed to prevent those attacks.

3.2. No Certificate Details

The OpenFlow document does not specify what certificate format will be used for the certificates that are exchanged between the switch and the controller, nor does it indicate what field(s) in the certificate will be used for naming. This could lead to operational complexity (if different names need to be used in different

implementations to indicate the same device), or to a lack of interoperability.

3.3. Importance of Checking Certificates on Both Sides

Although the OpenFlow specification indicates that certificates will be exchanged in both directions, it does not explicitly state that the certificates must be checked on both ends of the connection. There are many TLS implementations that do not currently check client certificates, and if a similar approach was used in OpenFlow, it could result in vulnerability to man-in-the-middle attacks.

3.4. TLS 1.0 Vulnerabilities

The security text in the OpenFlow specification refers to "TLS", without any reference or version number. The failure to specify a TLS version number could results in non-interoperable implementations if some OpenFlow implementations include TLS 1.0 and others include TLS 1.1. Also, there are security vulnerabilities in TLS 1.0 that have been fixed in TLS 1.1. So, OpenFlow implementations that use TLS 1.0 may be subject to man-in-the-middle attacks, as well as other attacks against TLS 1.0. It would be advisable to mandate the use of TLS 1.1.

3.5. Failure Handling Solutions

After a switch loses contact with all controllers, a switch will enter the either "fail secure mode" in which the switch transfer packets according to the existing records in the flow table or "fail standalone mode" in which the switch acts as a normal switch or router, depending upon the switch implementation and configuration. The OpenFlow specification does not support specifying such fail modes at run time, and thus after a controller lost connections with its switches it will not know how the packets will be processed by the switches. This problem will introduce difficulties in damage confinement or bring potential security issues.

In addition, in most scenarios, after generating a flow, the controller needs to distribute the policies of the flow to all the associated switches. During this procedure, if a switch meets any problem in deploying the policies, all the other related switches must not use these policies either. However, the current specification does not discuss how to support this atomic requirement.

3.6. Solutions to Keep the Consistancy of Flow Policies

The OpenFlow specification has not yet well analyzed the multiple headed scenarios where multiple applications try to modify the policies of the same flow concurrently. If multiple applications does not update the policies of the same flow in an well organized way, errors may be raised.

4. Applicability of OpenFlow Security Mechanisms

4.1. One Controller per Switch Scenario

The OpenFlow specification was originally written with the idea that there would be one controller (or a small set of tightly-coordinated controllers in a redundant deployment) controlling a set of switches. The OpenFlow specification correctly identifies two of the primary security threats in this scenario as eavesdropping and controller impersonation. The security mechanisms described in the OpenFlow specification are well-suited to protect against those threats. With some clarifications (as described above), the same mechanisms could be effective against man-in-the-middle attacks, which are also a signficant concern in this scenario.

4.2. Security in Other Scenarios

Recent SDN discussions have raised the idea that multiple, non-tightly-coordinated processes might want to control the switching behavior of a network. There are two high-level scenarios under discussion to accomplish this goal, one where multiple controllers are used to control the same set of switches, and another where the needs of multiple control processes are mediated by a centralized controller that controls a set of switches. This section explores the applicability of the security mechanisms in the OpenFlow protocol, as currently specified, to those scenarios.

4.2.1. Multiple Controllers per Switch

In this scenario, multiple control processes talk directly to a single switch. OpenFlow security is poorly-suited to use in this scenario for two reasons: lack of support for authorization, and a lack of granular access/control.

OpenFlow authentication is, essentially, a binary process. An authenticated controller has access to view or change the full configuration of the switch. It also has access to all of the traffic flowing through the switch. This is not desirable in a case where mutliple control processes are being used to control different portions of the network traffic.

4.2.2. Multiple Apps Talking to a Central Controller

This scenario effectively combines the two scenarios described above.

At the top-level, multiple control processes are talking to a single centralized controller, each communicating its own needs. This is akin to the "Multiple Controllers per Switch" scenario described in the previous section. OpenFlow, as currently specified, is not well-suited for use at this level, due to its lack of support for authorization and fine-grained access control.

At the lower-level, a single, centralized controller is used to control a group of switches. Ignoring the specification issues raised earlier in this document, the security mechanisms defined in the OpenFlow specification are well-suited for communication between the centralized controller and the switches in this scenario.

5. Suggestions for Future Work

We would recommend that the IETF publish a document advising the ONF about the current weaknesses in the OpenFlow security specification. The document should also make specific suggestions for updates to the OpenFlow specification that would address those weaknesses. This document should focus on clarifications needed to ensure interoperability, as well as changes needed to eliminate vulnerabilities. The ONF could then decide when or if to include the suggested changes in the OpenFlow specification.

We would also recommend that the IETF publish a document outlining the security requirements for a protocol to run between applications and an SDN controller, or between multiple SDN controllers and a set of switches. This document would be useful for operators who are deciding what protocols to use for their SDN deployments, or for protocol designers (in the IETF, in the ONF or elsewhere) to use as the basis for designing security mechanisms for protocols intended for that purpose.

6. Security Considerations

TBD

7. Acknowledgements

This document was written using the xml2rfc tool described in RFC 2629 [RFC2629].

8. Informative References

[RFC2629] Rose, M.T., "Writing I-Ds and RFCs using XML", RFC 2629,
June 1999.

Authors' Addresses

Margaret Wasserman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Phone: +1 781 405 7464
Email: mrw@painless-security.com
URI: http://www.painless-secuirty.com

Sam Hartman
Painless Security

Email: hartmans-ietf@mit.edu