# Anchors of Trust for Autonomic and Secure Configuration and Assessment in SDN

Diego Kreutz, Fernando Ramos, Paulo Verissimo
*LaSIGE/FCUL, University of Lisbon, Portugal*
*kreutz@lasige.di.fc.ul.pt, fvramos@fc.ul.pt, pjv@di.fc.ul.pt*

## I. Introduction

Contrary to traditional networks, in software defined networks (SDNs) there is a strong separation of the control and data planes. This logical centralization of the control plane functionality and applications in a controller, a.k.a. network operating system (NOS), that runs on commodity servers introduces a great level of flexibility in the network. It also poses new challenges, such as ensuring the reliability and trustworthiness of the control plane. For instance, malicious controllers or forwarding devices can more easily wreak havoc with the network [1], [2], [3]. Therefore, one of the crucial issue of SDN is to establish and maintain trustworthy relationships between controllers and forwarding devices.

To give an idea of the criticality of this scenario, Figure 1 shows the number of vulnerabilities of Cisco's IOS [4], one of the most widely deployed network operating systems. The number of published vulnerabilities is following a trend of nearly continuous growth since 1996. In 2012 and 2013, 58 vulnerabilities were published, representing around 21% of all vulnerabilities since 1992. Even more disturbing is the fact that more than 90% of them score four or more points in the NIST's common vulnerability scoring system, i.e., are of medium to high severity risk. It is worth emphasizing that the IOS is developed by highly skilled network-savvy teams since its inception. It has been thoroughly tested and analyzed. Nevertheless, it ranks among the fifteen operating systems with more vulnerabilities since 1999 [4]. If we think that one of the SDN's selling points is to make it easier for non-network experts to create applications to control the network, things can get even worse if security and dependability are not considered as first class priorities, i.e., crucial design principles for SDNs.

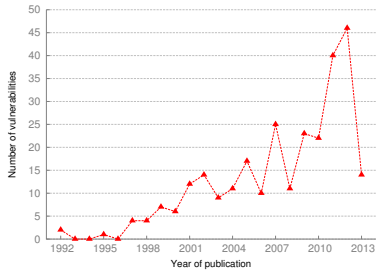


Figure 1. Vulnerabilities of Cisco's IOS from 1992 to September of 2013

We are working on the design of a secure and dependable SDN architecture, based on automatic fault and intrusion handling [5]. As a step of this process, we propose a resilient and trustworthy third party (RTTP) to provide essential mechanisms for allowing secure and automatic configuration (e.g., updated list of reliable controllers) and trustworthiness assessment of control plane elements in SDNs. Our approach is anchored in four requirements. First, it is necessary to ensure the resiliency of the RTTP itself. Second, the RTTP should provide means (e.g., protocols and security mechanisms) to establish and monitor trust relationship between controllers and switches. Third, this external entity should provide trustworthiness assessment mechanisms. For instance, suspected devices should be isolated and/or recovered. Forth, the RTTP should be simple to use and deploy in an SDN environment, making it straightforward to add services that are able to provide reliable configuration and security enhancement features to the network.

## II. Problem Statement

SDNs still are vulnerable to attacks from faked or compromised devices. For instance, with a single device a malicious user can easily launch attacks against controllers and neighboring switches [2], [3], [1]. It is worth mentioning that most of the existing OpenFlow enabled switches are hybrid, i.e., are complex devices that keep backward compatibility and only add OpenFlow as one more protocol. Moreover, by having access to the network an attacker can also introduce its own faked devices, which allows he/she to take control over the network, or part of it.

In previous work we have identified several threat vectors in SDNs [1]. A part of them is related with the lack of trust relationship between forwarding devices and controllers.

Typically, in current OpenFlow enabled networks the IP addresses of the controllers have to be manually configured in forwarding devices. Notwithstanding, forwarding devices can connect to any of the available controllers without strong access control or authentication mechanisms. Additionally, there is no certificate management solution for providing mutual authentications, no trust management between devices and no anchors of trust. Furthermore, most of the currently available OpenFlow devices do not support TLS (i.e., are prone to man-in-the-middle attacks on control plane communications), lack access control mechanisms on controllers, lack strong switch access authentication and authorization, and are prone denial of service attacks on the control plane [2], [3]. One of the few exceptions is the Open

vSwitch (http://openvswitch.org/), running mostly on commodity hardware platforms, which already supports TLS for control plane communications. A few recent industry-driven initiatives, such as `of-config` [6], provide a specification for sending the list of controllers' IPs and port numbers to the forwarding devices. However, this is not enough to ensure reliable and trustworthy relationship between devices and continuous trustworthiness assessment, for instance.

Another critical concern is related to configuration errors, which are one of the major causes of today's system failures. As an example, misconfigurations are the second major cause of service-level failures at one of Google's main services [7]. In fact, it has already been identified that misconfigurations of commercial OpenFlow switches can easily lead to attacks on the control plane [2], [3].

Therefore, one of our objectives is to investigate issues regarding the security and reliability of control plane device relationship, communications and trustworthiness assessment of devices. For instance, how can manual configuration (e.g., IPs of controllers) be avoided in an effective and safe way? How to establish trustworthy relationship between controllers and forwarding devices? How to allow dynamic instantiation and placement of controllers without compromising the infrastructure's security? How can the trustworthiness of networking devices be measured? How to identity and isolate (or replace) suspicious devices?

## III. Proposed Approach

Considering the posed questions for the development and deployment of future SDNs, one of our main research goals is to identify the requirements and building blocks for developing and deploying an autonomous and reliable trusted third party for providing some of the assurances required to build more reliable and trustworthy software defined networks. We briefly discuss our proposal for providing reliable configuration automation and trustworthiness assessment in controllers and forwarding devices, as illustrated in Figure 2. To achieve this goal, a RTTP is used to provide essential functions such as allow administrators to easily assign unique certificates to new devices, provide dynamic and periodical certificate and key update protocols, keep track of available controllers and forwarding devices, provide lookup services for identifying the available controllers and forwarding devices, recommend controllers or forwarding devices based on their history, reputation and trustworthiness assessment of the devices, and provide periodical trustworthiness assessment of each network element.

Some of the major challenges for achieving the proposed solution are: *design a reliable and trustworthy RTTP* to ensure a correct and autonomic configuration and trustworthiness assessment of controllers and forwarding devices; *use roots of trust to augment the trustworthiness of the RTTP itself*, i.e., combine remote attestation solutions, such as those provided by tamper resistant FPGAs [8] and TPMs

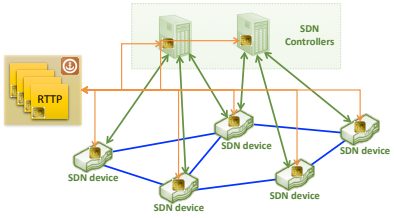


Figure 2. A RTTP for automatic security and trustworthiness assessment

similar to those that Cisco Systems is introducing as roots of trust in network devices [9], with traditional state machine replication protocols, diversity and proactive-reactive recovery techniques [10], to ensure continued correct operation; *use roots of trust to assess the trustworthiness of SDN devices*, which are essential components to enable remote attestation; and *integrate models and tools for assessing the reputation of devices*, making it feasible to recommend controllers for OpenFlow enabled forwarding devices.

A trust management model specifies, analyses, establishes, monitors and finishes trust relationship. The indirect trust relationship are given by recommendations from the RTTP. Thus, the trust management model should: (a) establish trust relationship in an autonomous way, reducing the complexity of the interactions between SDN devices; (b) protect the critical data (e.g. certificates, shared keys) from malicious entities; (c) do periodical trustworthiness assessment of all devices; (d) minimize the human intervention, automating the certificate renewal process, trustworthiness assessment and device recommendations; and (e) make suitable recommendations (e.g. isolate a suspected device) despite eventual uncertainty.

## Acknowledgment

This work is supported by EU's FP7, through SecFuNet (FP7-ICT-STREP-288349), and by CNPq, through grant 202104/2012-5.

## References

[1] D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *SIGCOMM HotSDN*, 2013.
[2] K. Benton, L. J. Camp, and C. Small, "Openflow vulnerability assessment," in *SIGCOMM HotSDN*, 2013.
[3] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in *IEEE SDN4FNS*, 2013.
[4] "Cisco IOS: Security vulnerabilities," 2013, http://goo.gl/9msfvu.
[5] P. Verissimo, N. Neves, C. Cachin, J. Poritz, D. Powell, Y. Deswarte, R. Stroud, and I. Welch, "Intrusion-tolerant middleware: the road to automatic security," *IEEE Security & Privacy*, vol. 4, no. 4, 2006.
[6] ONF, "OpenFlow Management and Configuration Protocol," 2013, http://goo.gl/uSr9gE.
[7] U. Hoelzle and L. A. Barroso, *The Datacenter As a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 1st ed. Morgan and Claypool Publishers, 2009.
[8] E. Peterson, "Developing Tamper Resistant Designs with Xilinx Virtex-6 and 7 Series FPGAs," 2013, http://goo.gl/tvTphR.
[9] R. M. Montalvo, "Trusted systems in networking infrastructure," University Park, PA, 2013, http://goo.gl/jkxjqr.
[10] P. E. Verissimo, N. F. Neves, and M. P. Correia, "Intrusion-tolerant architectures: Concepts and design," in *Architecting Dependable Systems*, ser. Lecture Notes. Springer, 2003, vol. 2677.