Report from Dagstuhl Seminar 16361

Network Attack Detection and Defense – Security Challenges and Opportunities of Software-Defined Networking

Edited by

Marc C. Dacier[1], Sven Dietrich[2], Frank Kargl[3], and Hartmut König[4]

1 **QCRI – Doha, QA,** mdacier@qf.org.qa
2 **City University of New York, US,** spock@ieee.org
3 **Universität Ulm, DE,** frank.kargl@uni-ulm.de
4 **BTU Cottbus, DE,** hartmut.koenig@b-tu.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 16361 "Network Attack Detection and Defense: Security Challenges and Opportunities of Software-Defined Networking".

Software-defined networking (SDN) has attracted a great attention both in industry and academia since the beginning of the decade. This attention keeps undiminished. Security-related aspects of software-defined networking have only been considered more recently. Opinions differ widely. The main objective of the seminar was to discuss the various contrary facets of SDN security. The seminar continued the series of Dagstuhl events Network Attack Detection and Defense held in 2008, 2012, and 2014. The objectives of the seminar were threefold, namely (1) to discuss the security challenges of SDN, (2) to debate strategies to monitor and protect SDN-enabled networks, and (3) to propose methods and strategies to leverage on the flexibility brought by SDN for designing new security mechanisms. At the seminar, which brought together participants from academia and industry, we discussed the advantages and disadvantages of using software-defined networks from the security point of view. We agreed that SDN provides new possibilities to better secure networks, but also offers a number of serious security problems which require further research. The outcome of these discussions and the proposed research directions are presented in this report.

Seminar September 4–9, 2016 – http://www.dagstuhl.de/16361
1998 ACM Subject Classification C.2.1 Network Architecture and Design, C.2.3 Network Operations, K.6.5 Security and Protection
Keywords and phrases attack detection, denial-of-service attack detection and response, intrusion detection, malware assessment, network monitoring, openflow protocol, programmable networks, security, software-defined networking, targeted attacks, vulnerability analysis
Digital Object Identifier 10.4230/DagRep.6.9.1
Edited in cooperation with Radoslaw Cwalinski

Network Attack Detection and Defense – Security Challenges and Opportunities of Software-Defined Networking,
Dagstuhl Reports, Vol. 6, Issue 9, pp. 1–28
Editors: Marc C. Dacier, Sven Dietrich, Frank Kargl, and Hartmut König
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Hartmut König
Marc C. Dacier
Sven Dietrich
Frank Kargl
Radoslaw Cwalinski

From September 4 through 9, 2016, more than 40 researchers from the domains of computer networks and cyber security met at Schloss Dagstuhl to discuss security challenges and opportunities of software-defined networking (SDN).

Software-defined networking has attracted a great attention both in industry and academia since the beginning of the decade. This attention keeps undiminished. In 2014, IDC predicted that the market for SDN network applications would reach $1.1bn. Especially in industry, the vision of "programming computer networks" has electrified many IT managers and decision makers. There are great expectations regarding the promises of SDN. Leading IT companies, such as Alcatel-Lucent, Cisco systems, Dell, Juniper Networks, IBM, and VMware, have developed their own SDN strategies. Major switch vendors already offer SDN-enabled switches.

Software-defined networking provides a way to virtualize the network infrastructure to make it simpler to configure and manage. It separates the control plane in routers and switches, which decides where packets are sent, from the data plane, which forwards traffic to its destination, with the aim to control network flows from a centralized control application, running on a physical or virtual machine. From this controller, admins can write and rewrite rules for how network traffic, data packets, and frames are handled and routed by the network infrastructure. Routers and switches in a sense become "slaves" of this application-driven central server. SDN-enabled networks are capable of supporting user requirements from various business applications (SLAs, QoS, Policy Management, etc.). This is not limited to the network devices of a certain vendor. It can be applied to devices from various vendors if the same protocol is used. Most SDN infrastructure utilizes the widely-used OpenFlow protocol and architecture to provide communication between controllers and networking equipment.

Security-related aspects of software-defined networking have only been considered more recently. Opinions differ widely. Some believe that the security problems introduced by SDN are manageable – that SDN can even bring security benefits; others think that Pandora's Box has been opened where SDN and SDN-enabled networks can never be secured properly.

No doubt, there are a number of serious security problems as the following examples show. SDN controllers represent single points of failures. The controllers as well as the connections between controllers and network devices might be subject to distributed denial of service attacks. Compromising the central control could give an attacker command of the entire network. The SDN controllers are configured by network operators. Configuration errors can have more complex consequences than in traditional settings because they may unpredictably influence the physical network infrastructure. Furthermore, the idea of introducing 'network applications' that interact with the controller to modify network behavior seems like a complexity nightmare in terms of required authentication and authorization schemes. Finally, the SDN paradigm is a major turn around with respect to the basic design rules that have made the Internet successful so far, namely a well-defined layered approach. Whereas in today's world, applications have no say in routing decisions, SDN's promise for highly flexible and application-tailored networking requires a way for applications to optimize networking decisions for their own benefits. However, it is unclear to what extent fairness can be ensured, how conflicting decisions can be resolved, etc. Along the same line, members of the security community worry about the possibility to intentionally design SDN applications that could eventually be turned into attack weapons or simply be misused by malicious attackers. Whether these fears are substantiated or not is something which has not received any scrutiny so far.

On the other hand, SDN is also considered by many researchers as an effective means to improve the security of networks. SDN controllers can be used, for instance, to store rules about the permission of certain requests which cannot be decided at the level of a single switch or router because this requires full overview over network status or additional information and interactions which are not contained in the current protocol versions. Attacks that can be detected this way are ARP spoofing, MAC flooding, rogue DHCP server, and spanning tree attacks. Also, by enabling the creation of virtual networks per application, people speculate that intrusion detection techniques relying on the modeling of the normal behavior of network traffic will become much easier to implement and more reliable in terms of false positive and negatives. Similarly, SDN apps could offer a very simple and effective way to implement quarantine zones for infected machines without cutting them off completely from the network since the quarantine could be customized at the application level (letting DNS and HTTP traffic for a given machine go through but not SMTP, for instance).

These two contrary facets of SDN security were the key ingredients for an extremely lively and very fruitful seminar. The seminar brought together junior and senior experts from both industry and academia, covering different areas of computer networking and IT security. The seminar started with two invited talks by Boris Koldehofe (TU Darmstadt, DE) and Paulo Jorge Esteves-Veríssimo (University of Luxembourg, LU) on the basics and security aspects of software-defined networking. After that we organized six working groups to discuss in two rounds the Good and the Bad of using SDN from the security point of view. Based on the outcome of the working groups and a plenary discussion, we formed another four working groups to discuss required research directions. The first six working groups focus on the following issues: (1) centralization in SDN, (2) standardization and transparency, (3) flexibility and adaptability for attackers and defenders, (4) complexity of SDN, (5) attack surface and defense, and (6) novelty and practicability. The research direction working groups dealt with (1) improving SDN network security, (2) a secure architecture for SDN, (3) secure operation in SDN-based environments, and (4) SDN-based security. The discussion in the working groups was supplemented by short talks of participants to express their positions on the topic or to report about ongoing research activities. Based on the talks, discussions, and working groups, the Dagstuhl seminar was closed with a final plenary discussion which summarized again the results from the working groups and led to a compilation of a list of statements regarding the security challenges and opportunities of software-defined networking. The participants agreed that SDN provides new possibilities to better secure networks, but also offers a number of serious security problems which have to be solved for being SDN a successful technology. The outcome of these discussions and the proposed research directions are presented in the following.

2 Table of Contents

Executive Summary

Invited Talks

Overview of Talks

Working Groups: The Good and the Bad of SDN

Working Groups: Research Directions

Final Plenary Discussion

3 Invited Talks

3.1 An overview on Software-defined Networking

Boris Koldehofe (TU Darmstadt, DE)

Joint work of Frank Dürr, Boris Koldehofe

Software-defined networking is currently a big trend in networking with strong support from both academia and industry. The basic concept of SDN is the separation of network control (control plane) and forwarding functionality (forwarding plane). The control plane is implemented by a controller hosted on a server, which programs the forwarding tables of switches to define communication "flows" in the network. Formerly distributed control logic like distributed routing algorithms are replaced by logically centralized control based on a global view onto the network. This talk discusses the motivation of SDN, offers a basic introduction of the corresponding concepts, and discusses some fundamental challenges.

3.2 Towards Secure and Dependable Software-Defined Networks

Paulo Jorge Esteves-Veríssimo (University of Luxembourg, LU)

Joint work of Paulo Jorge Esteves-Veríssimo, Diego Kreutz, Fernando Ramos
Main reference D. Kreutz, F. M. V. Ramos, P. Verissimo, "Towards secure and dependable software-defined networks", in Proc. of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN'13), pp. 55–60, ACM, 2013.
URL http://dx.doi.org/10.1145/2491185.2491199

Software-defined networking empowers network operators with more flexibility to program their networks. With SDN, network management moves from codifying functionality in terms of low-level device configurations to building software that facilitates network management and debugging. By separating the complexity of state distribution from network specification, SDN provides new ways to solve long-standing problems in networking, e.g., routing, while simultaneously allowing the use of security and dependability techniques, such as access control or multi-path. However, the security and dependability of the SDN itself is still an open issue. In this position paper we argue for the need to build secure and dependable SDNs by design. As a first step in this direction, we describe several threat vectors that may enable the exploit of SDN vulnerabilities. We then sketch the design of a secure and dependable SDN control platform as a materialization of the concept advocated here. We hope that this paper will trigger discussions in the SDN community round these issues and serve as a catalyser to join efforts from the networking and security & dependability communities in the ultimate goal of building resilient control planes.

4 Overview of Talks

4.1 Network Monitoring & SDN

Johanna Amann (ICSI – Berkeley, US)

Joint work of Johanna Amann, Robin Sommer
Main reference J. Amann, R. Sommer, "Providing Dynamic Control to Passive Network Security Monitoring", in Proc. of the 18th Int'l Symp. on Research in Attacks, Intrusions, and Defenses (RAID'15), LNCS, Vol. 9404, pp. 133–152, Springer, 2015.
URL http://dx.doi.org/10.1007/978-3-319-26362-5_7

Passive network intrusion detection systems detect a wide range of attacks, yet by themselves lack the capability to actively respond to what they find. Some sites thus provide their IDS with a separate control channel back to the network, e.g., by interacting with SDN capable hardware. In the past, such setups tended to remain narrowly tailored to the site's specifics with little opportunity for reuse elsewhere, as different networks deploy a wide array of hard- and software and differ in their network topologies. To overcome the shortcomings of such ad-hoc approaches we present a network control framework that provides passive network monitoring systems with a flexible, unified interface for active response, hiding the complexity of heterogeneous network equipment behind a simple task-oriented API. We give our experiences deploying our framework in a production network. Furthermore, we sketch future research directions that offload expensive low-level operations from software into network hardware.

4.2 Improving Network Security by SDN – OrchSec and AutoSec Architectures

Kpatcha Mazabalo Bayarou (Fraunhofer SIT – Darmstadt, DE) and Rahamatullah Khondoker

Main reference A. Zaalouk, R. Khondoker, R. Marx, K. M. Bayarou, "OrchSec: An orchestrator-based architecture for enhancing network-security using Network Monitoring and SDN Control functions", in Proc. of the 2014 IEEE Network Operations and Management Symposium (NOMS'14), pp. 1–9, IEEE, 2014.
URL http://dx.doi.org/10.1109/NOMS.2014.6838409
Main reference R. Khondoker, P. Larbig, D. Senf, K. Bayarou, N. Gruschka, "AutoSecSDNDemo: Demonstration of Automated End-to-End Security in Software-Defined Networks", in Proc. of the 2nd IEEE Conf. on Network Softwarization (NetSoft'16), pp. 347–348, IEEE, 2016.
URL http://dx.doi.org/10.1109/NETSOFT.2016.7502404

According to statistics of Deutsche Telekom [1], the number of network attacks per month has increased from 100,000 to 550,000 within 12 months (June 2015 – June 2016). Traditional defense mechanisms that are based on the strategy to automatically detect and manually mitigate attacks are deemed inefficient especially in the context of Industrie 4.0 applications. The concept of Software-Defined Networking (SDN) is based on the separation of the control plane from the data plane of network entities, whereas an SDN controller (representing the control plane) takes decisions based on forwarding rules, routers, switches, etc. (representing the data plane) forward the data accordingly. The planes communicate with each other by an open interface, such as OpenFlow, so that the data plane can directly be programmed. Among others, these centralized monitoring and control features of SDN can be adopted to detect and mitigate network attacks automatically. Towards this, two architectures named OrchSec

[2, 3] and AutoSec [4], have been developed by Fraunhofer SIT. While OrchSec detects and mitigates network attacks, such as DDoS, automatically in a reactive manner, AutoSec takes proactive actions, such as dynamically configuring both the clients connected to a network and the devices forwarding the data, to prevent the networks from being attacked successfully. OrchSec and AutoSec have been integrated and tested in SDN-enabled/SDN-only hardware devices from major switch vendors, such as Huawei, HP, and Cisco.

References

1 DTAG. *Overview of current cyber attacks on Deutsche Telekom AG (DTAG) sensors.* http://www.sicherheitstacho.eu/?lang=en, accessed on 04.08.2016

2 Adel Zaalouk, Rahamatullah Khondoker, Ronald Marx, Kpatcha M. Bayarou. *OrchSec: An orchestrator-based architecture for enhancing network-security using Network Monitoring and SDN Control functions.* NOMS 2014:1–9

3 Adel Zaalouk, Rahamatullah Khondoker, Ronald Marx, Kpatcha M. Bayarou. *OrchSec Demo: Demonstrating the Capability of an Orchestrator-based Architecture for Network Security.* Academic Demo, Open Networking Summit 2014 (ONS 2014), Santa Clara, USA, 3-5 March 2014

4 Rahamatullah Khondoker, Pedro Larbig, Daniel Senf, Kpatcha Bayarou, Nils Gruschka. *AutoSecSDNSemo: Demonstration of Automated End-to-End Security in Software-Defined Networks.* IEEE NetSoft 2016, 6-10 June 2016, Seoul, South Korea

4.3 SDN: A Network Economics Inflection Point

L. Jean Camp (Indiana University – Bloomington, US)

Joint work of Kevin Benton, L. Jean Camp, Martin Swany
Main reference K. Benton, L. J. Camp, "Firewalling Scenic Routes: Preventing Data Exfiltration via Political and Geographic Routing Policies", in Proc. of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense, pp. 31–36, ACM, 2016; pre-print available from author's webpage.
URL http://dx.doi.org/10.1145/2994475.2994477
URL http://www.ljean.com/files/SAFECONFIG2016.pdf
Main reference C. Hall, D. Yu, Z.-L. Zhang, J. Stout, A. M. Odlyzko, A. W. Moore, L. J. Camp, K. Benton, R. J. Anderson, "Collaborating with the Enemy on Network Management", in Proc. of the 22nd Int'l Workshop on Security Protocols, LNCS, Vol. 8809, pp. 163–171, Springer, 2014; pre-print available from author's webpage.
URL http://dx.doi.org/10.1007/978-3-319-12400-1_15
URL https://www.cl.cam.ac.uk/~rja14/Papers/spw14-08-Anderson.pdf

BGP enables as a network of networks, and is also a network of trust. The most clear instantiation of that trust is the updating of router tables based on unsubstantiated announcements. The positive result of this trust is that the network can be extremely responsive to failures, and recover quickly. Yet the very trust that enables resilience creates risks from behavior lacking either technical competence or benevolence. Threats to the control plane have included political interference, misguided network configurations, and other mischief. BGPSEC has been proposed to resolve this, but the economics of path validation are the opposite of incentive aligned.

SDN offers an new approach to economics of networking. To show that this inflection point can improve network-wide security, we constructed a proof-of-concept. This proof of concept translates a series of route updates into a RIB, which is then converted to a flow information base (FLIB). The FLIB then can be subject to arbitrary analysis to defeat different types of attacks. For example, content-leaking misdirection attacks via incorrect routing announcements could become immediately identifiable and individual networks could defend themselves from remote actors.

4.4 Network Security Management for Trustworthy Networked Services

Georg Carle (TU München, DE)

Joint work of Georg Carle, Cornelius Diekmann, Paul Emmerich, Sebastian Gallenmüller, Oliver Gasser, Nadine Herold, Matthias Wachs

When looking back to the previous research area of active and programmable networks 20 years ago, today's architecture of SDN-based networks can be seen as an evolution of these approaches. Our network security management approach combines different methods and components: Tools for automated and reproducible experiments allow automated load and penetration tests of real software and automated mitigation [1], [2]. Internet-wide measurements [3] provide a range of data that can be used in the testbed. Formally verified tools that allow to generate SDN flow tables and firewall rules from high-level specifications [5], and also allow to translate configurations of legacy devices into the same high-level specifications [4].

References

1 Emmerich, Paul and Gallenmüller, Sebastian and Raumer, Daniel and Wohlfart, Florian and Carle, Georg. *MoonGen: A Scriptable High-Speed Packet Generator*. ACM SIGCOMM Internet Measurement Conference (IMC) 2015, Tokyo, Japan, October 2015

2 Wachs, Matthias and Herold, Nadine and Posselt, Stephan-A. and Dold, Florian and Carle, Georg. *GPLMT: A Lightweight Experimentation and Testbed Management Framework*. Passive and Active Measurement: 17th International Conference, PAM 2016, Heraklion, Greece, March 2016

3 Wachs, Matthias and Herold, Nadine and Posselt, Stephan-A. and Dold, Florian and Carle, Georg. *Scanning the IPv6 Internet: Towards a Comprehensive Hitlist*. 8th Int. Workshop on Traffic Monitoring and Analysis TMA 2016, Louvain-la-Neuve, Belgium, April 2016

4 Diekmann, Cornelius and Michaelis, Julius and Haslbeck, Maximilian and Carle, Georg. *Verified iptables Firewall Analysis*. IFIP Networking 2016, Vienna, Austria, May 2016

5 Diekmann, Cornelius and Korsten, Andreas and Carle, Georg. *Demonstrating topoS: Theorem-Prover-Based Synthesis of Secure Network Configurations*. 2nd International Workshop on Management of SDN and NFV Systems, manSDN/NFV 2015, Barcelona, Spain, November 2015

4.5 RADIator – An Approach for Secure and Controllable Wireless Networks

Radoslaw Cwalinski (BTU Cottbus, DE)

Joint work of Radoslaw Cwalinski, Hartmut König
Main reference R. Cwalinski, H. König, "RADIator – An approach for controllable wireless networks", in Proc. of the 2nd IEEE Conf. on Network Softwarization (NetSoft'16), pp. 260–268, IEEE, 2016.
URL http://dx.doi.org/10.1109/NETSOFT.2016.7502421

Wireless local area networks (WLANs) became an essential part of todays enterprise network infrastructures. Due to the use of a shared medium – the electromagnetic waves – for transmitting data, wireless networks are inherently exposed to diverse attacks, such as for example Denial of Service (DoS) attacks at different network layers.

In the talk, we propose a software-defined networking architecture for enterprise wireless local area networks. In our architecture, the access point's (AP) management tasks, including beaconing, client authentication and association, are performed by the central controller instead of by the distributed wireless APs as in traditional networks. The goal is to provide a framework that exposes tools and methods for centralized, fine-grained inspection and processing of 802.11 frames and enable network applications to run in the central controller.

We present our architecture together with examples of controller-based applications that we are currently working on. These applications, such as centralized traffic inspection, anomaly detection, WLAN topology and interference recognition, wireless client geolocalization and client fingerprinting help to optimize and secure the WLAN. We introduce a "trust level"-based access control for wireless clients that uses geolocation information ("where you are"), device fingerprinting ("what you have"), anomaly detection ("what you do") and user credentials ("what you know") to take access decisions, set routing rules or trigger alerts.

4.6 The THD-Sec network security experimental testbed

Hervé Debar (Télécom & Management SudParis – Evry, FR)

The THD-Sec platform is an experimental environment dedicated to network security. It aims at enabling multiple attack and defense scenarios to provide experimental validation of new ideas for network defense. It includes classic IT technologies and interfaces to SCADA protocols. Examples of use of the platform have been published in [1] and [2].

References

1 Sahay, Rishikesh and Blanc, Gregory and Zhang, Zonghua and Debar, Hervé. *Towards Autonomic DDoS Mitigation using Software Defined Networking.* SENT 2015, Feb 2015, San Diego, Ca, United States. Internet society

2 Fabre, Pierre-Edouard and Debar, Hervé and Viinikka, Jouni and Blanc, Gregory. *ML: DDoS Damage Control with MPLS.* NordSec 2016:101–116

4.7 Security in ICS Networks

Tobias Limmer (Siemens AG – München, DE)

Many Industrial Control System solutions have a similar networking topology for which a common deployment practice has developed. As security standards increasingly gain attention, those deployments need to be adapted to new security requirements. This does not only apply to the design of the solution, but also to documentation, implementation, and verification practice. This talk presents an overview of the common deployment practice, security requirements, and open questions.

4.8 Authentication and Authorization in Wired OpenFlow-Based Networks Using 802.1X

Michael Menth (Universität Tübingen, DE)

Joint work of Frederik Hauser, Michael Menth, Mark Schmidt

802.1X is the most widely used authentication and authorization protocol in wired LANs. However, in OpenFlow-based networks, mainly MAC-address-to-identity mapping and web frontend based mechanisms are used which are highly insecure or cumbersome and little flexible, respectively. We propose to integrate the 802.1x authenticator in a network application such that it can support also others than RADIUS-based authentication resources. Further, a network-wide session database is maintained which enables identity-based network control. The authenticator is a network function that can be virtualized and well scaled. Most importantly, the approach is compatible with current infrastructures such as network clients and existing RADIUS-based authentication resources.

4.9 Robust Policy Checking

Christian Röpke (Ruhr-Universität Bochum, DE) and Thomas Lukaseder (Universität Ulm, DE)

The complexity and strategic position of SDN controllers in the network make them a rewarding target for attacks. Taking over an SDN controller means complete control over the network infrastructure. Despite their importance and their value, both for network operators and attackers alike SDN controllers are not secured properly against attacks in their current state. The complex structure of SDN controllers that also offer the possibility of including third party applications makes them hard to secure. Policy checkers are able to verify the compliance of the network set-up against a set of policies and can therefore serve as a warning system whether a controller is compromised. However, current policy checkers are usually placed close to the SDN controller on the same machine. Prior research shows that identifying a compromised SDN controller as such can therefore be circumvented by an attacker. We discuss our ideas on different possible ways to integrate policy checkers in the network independently of SDN controllers. This makes policy checking more robust against a compromised control plane.

4.10 Initial Measurements on Delay Issues within SDN WAN-Scenarios

Thomas Scheffler (Beuth Hochschule für Technik – Berlin, DE)

Joint work of Thomas Scheffler, Awono Ngono, Gabrielle Nelly

Current SDN deployment focuses on data-centers where large content-providers have shown the value of the technology. As the technology matures and equipment becomes more readily available, other deployment areas may become interesting. Our work focuses on the use of SDN technology in Wide Area Networks (WANs). It has been shown before by others [1] that a small number of controllers could serve a large geographic area, such as the Internet2. SDN-WAN deployments would naturally contain certain controller-switch paths that facilitate high propagation delay.

Assuming that such networks use reactive flow instantiation, the following condition holds: whenever traffic reaches the switch, for which no match could be found in the flow table, there exists the need to forward OFP 'packet-in' packets to the controller. These OFP packets will have to be send over a high-delay link and may have a tendency to queue up, if several such events occur in rapid succession. We expect that a high switch-controller delay may alter the behaviour of the network and may have consequences to the end-to-end connections represented by these flows.

In the talk we present our testbed that allows us to introduce a variable, controlled delay between the SDN switch and controller. Our experiments show that in certain circumstances a high switch-controller delay leads to a large number of OFP packets forwarded to the controller. Current SDN switches simply forward all incoming packets for an unknown flow to the controller. One or several high-bandwidth flows thus flood the switch-controller link with many unnecessary OFP packets that still need to be forwarded to and processed by the controller. Since these packets are forwarded via a high-delay link, a large number of packets are already in flight, before a control message can reach the switch. This could potentially lead to an increased work-load on the controller, saturation of the switch-controller link, increased packet-forwarding delay, and the introduction of novel Denial-of-Service scenarios. We also found that delay values higher than 150ms affect TCP connections, represented by the flows, causing additional retransmission of packets to reach the network.

References

1 Brandon Heller, Rob Sherwood and Nick McKeown. *The Controller Placement Problem.* Proceedings of the First Workshop on Hot Topics in Software Defined Network (HotSDN'12), Helsinki, Finland, 2012

4.11 Party's Over – Why we are not only late to the SDN party

Alexander von Gernler (genua GmbH – Kirchheim bei München, DE)

Discussions about SDN are nice, but what if our insights will later on not be needed by the real world, because they have found better alternatives or doing it on their own no matter what we recommend? In this talk, I analyse the needs of several potential SDN users, namely data centers, company networks, and university networks. Data centers will mostly undergo a market consolidation, leaving out barely more players other than the cloud services of the Big Five companies, among them Amazon AWS, Google, and Microsoft Azure. They most likely will not be in dire need of our insights generated at Dagstuhl, as they have enough manpower and resources to just do it on their own.

Company networks, on the other hand, will undergo a transformation getting much leaner, following ideas like Google's BeyondCorp. Thus, SDN will not be of great importance here as well. What is left are university networks. They are often open-minded and will adapt or at least try out new ideas conceived by science. But then again, they are a really small market, so the impact of our ideas will be limited if only used in a university context.

5 Working Groups: The Good and the Bad of SDN

5.1 What benefits more? Attack Surface or Opportunity for Defense?

Kpatcha Mazabalo Bayarou (Fraunhofer SIT – Darmstadt, DE)

Joint work of Kpatcha Mazabalo Bayarou, Georg Carle, Sven Dietrich, Paulo Jorge Esteves-Veríssimo, Mattijs Jonker, Thomas Lukaseder, Michael Meier, Christian Röpke, Thomas Scheffler, Han Xu

SDN definitely increases the attack surface and the standards notoriously lack security mechanisms, e.g., for authorization which are BAD. On the other hand, SDN provides means to implement new security features faster and introduce them into the system in cases that were not possible earlier which are GOOD. Detecting attacks may therefore become a lot easier and reliable.

So which of the two aspects is more relevant and how will the final balance be? The working group discusses the two aspects by considering what is bad or good for the attackers' perspectives. The same consideration is made with regard to the defenders' perspectives. For this discussion the members of the group come up with the consideration of the limitations that may face both sides depending on which aspect/case is under consideration i.e. the discussion on bad or on good.

The discussion on the BAD relates to the advantage that the attacker gets from the SDN technology. The centralized architecture of SDN, lack of defenders expertise, and immature technology could benefit the attackers. For example, the introduction of malicious controller apps may allow for wider impact of the attack.

The discussion on the GOOD relates to the advantage of SDN for defenders and the limitation SDN poses to attackers. The centralized architecture of SDN which brings global view of networks, open hardware interfaces, and central control might benefit the defenders.

For example, open hardware interface empowers developers and network operators to create tailored security solutions.

What is the final balance? Finding attack surfaces that the SDN brings, is the precondition to defend against them. When usable, affordable and standard solutions will be provided against the attack surfaces, then opportunities for defense will be increased as the defender will be able to create innovative protection mechanisms using SDN by shifting the focus from protecting the SDN itself.

5.2 Standardisation & Transparency

Radoslaw Cwalinski (BTU Cottbus, DE) and Hartmut König (BTU Cottbus, DE)

Joint work of Johanna Amann, L. Jean Camp, Georg Carle, Radoslaw Cwalinski, Marc C. Dacier, Jan Kohlrausch, Hartmut König, Thomas Scheffler

The goal of the working group was to discuss the benefits and disadvantages of standardization and transparency in Software-Defined Networks. On the one hand, with SDN/OF networks may converge to one standard and a few (open) implementations that are easier to secure or fix than the myriads of diverging solutions. On the other hand, monoculture is bad if successfully attacked.

Starting with the positive side of standardization the members of the working group identified the following aspects. First, standardization of protocols for controlling network devices mitigates the risks of erroneous configurations. Ideally, network devices operate with open interfaces, avoiding vendor lock-in and reducing costs. Standardization also brings more players into the game thus allows for competition whereas the current non-standardization create vendor lock-ins and software solutions that are not future-proof. Standardized interfaces allow network monitoring to use networking systems in an unprecedented way, i.e. to filter information that they do not need.

The group members recognized also the advantages of transparency which is particularly critical for routing and security applications. Transparency helps with testing, including penetration testing and fuzzing. It also allows conformance testing by different organizations with open test suites and open, public test results. The point is that although vendors claim to be standard compliant, it tends to be a false promise which cannot be easily verified without public test suites and public test results.

On the bad side, the group participants agreed that standardization is subject for manipulation for organizations with high resources. Complexity of standardization is a proven way to decrease the interoperability in practice thus increase opportunities for a vendor lock-in. Additionally, complex standard interfaces are hard to set up and to manage. They also can come with “standard vulnerabilities”. These vulnerabilities might therefore affect an even larger number of standardized systems. Network monoculture of such standardized systems may make it easier for attackers to compromise the system’s security. Further, current standards are often not suited for SDN, e.g., the standards of PKI for SDN are inappropriate. They can offer a false feeling of authentication and an illusion of security.

SDN will always need to interact with the legacy world. This interaction sets limitations to the security benefits of SDN. The challenges of BGP will not disappear with SDN – important threats like BGP prefix hijacking remain difficult to deal with. In addition, the presence of legacy middleboxes can also break many SDN-based security mechanisms. Debugging methods from legacy networks may be affected by SDN too e.g., ping may not follow the same path as http. Generally speaking, SDN programming may be influencing traffic in a complex way. The conclusion is that SDN promises network transparency but also challenges it.

The participants agreed that standards are often battles for finite resources. Increasingly, the standards become more complex and burden developers which leads to increased complexity at the software level. The sad truth is: security is traditionally sacrificed for interoperability.

Finally, the separation of organizations served on an airport has been presented as an use case to demonstrate the benefits of SDN. Today the separation is mostly done with MPLS which is limited and cumbersome to configure. Using SDN the isolation can be done in a convincing and straightforward way. Another example presented was the isolation of flows within an aircraft and between an aircraft and ground data centers involving different organizations: aircraft manufacturer, engine manufacturer, airline, maintenance organization, airport.

5.3 Flexibility and Adaptability for Attackers and Defenders

Boris Koldehofe (TU Darmstadt, DE)

Joint work of Boris Koldehofe, Simin Nadjm-Tehrani, Rene Rietz, Robin Sommer, Jens Tölle, Emmanuele Zambon

Preface: Some of the given statements are not exclusively valid for SDN. The advantages and disadvantages can occur with other advanced network management technologies as well. Standardized and widely-used approaches will intensify opportunities and risks.

Starting with the potentially problematic aspects of SDN usage the members of the working group identified the following challenges, most of them a cause of increased complexity:

- Code for managing and configuring SDN capable switches may come from various sources, and some of them may contain malicious contents.
- Networking devices may have technical capabilities which are not used by most of the users. So it is not transparent to hosts what the actual network configuration is.
- If more than one user is allowed to configure the system, even with good intentions there will be unknown side effects taking the system to places the service provider did not imagine.
- The flexible updates creates a need for much more complex access control systems that are hard to manage, and add to the complexity of the overall system.
- The notion of normality is harder to define in an SDN that is programmable simply due to larger degrees of freedom, and hence detection of abnormal events gets harder. Attackers can use this “confusion” of conception to hide the attack steps. The need for flexibility will mandate for more extensive interpretation of network data (i.e., looking at/parsing the application layer). This will increase the attack surface in both SDN switches and controllers.
- Attackers may get the same capabilities as the operators once they breach the trust management system – and they will exploit it.

All in all, attackers can actually control the operations in arbitrary ways, they can confuse or blind the defenders, or create inconsistencies. They are able to gather a global view of the network (and a more fine-grained too) from a single location. They will be able to exploit the additional complexity brought in by the flexibility (e.g., code exploitation on switch-side and controller-side).

The flexibility makes it harder for the defender of SDNs. Because of dynamic configurations, it is more difficult for a human to tell if the current/past configuration is intended/correct. The more user-friendly tools get, the less humans are able to do the job themselves and have a deep understanding of the underlying technology and protocols. The flexibility makes it hard to define meaningful policies for SDNs, e.g. which flows are affected by a specific network application and modified in a specific way. The flexibility provided by SDNs may exacerbate the conflicts between the objectives of networking teams vs. security monitoring teams.

The working group discussed also the positive aspects of application of SDN technologies. From the point of view of defenders, it gets easier to:

- do static and dynamic network isolation
- do fine granular authentication/authorization of clients
- enable active response (blocking, restricting), including deep inside the local network
- gain network overview, creating awareness on current security situation
- do adaptive monitoring (e.g., tell the switch that we don't want to see this particular flow (file transfer) anymore)
- do efficient network monitoring using in-network processing
- creating resilience: enable rate limiting or rerouting of traffic when under attack.

From an attackers point of view, the following attack-related activities get harder:

- Network reconnaissance
- Analysis of a properly separated network environments
- Man-in-the-middle attacks using spoofing (if there is a proper SDN concept, e.g., address configuration/resolution using SDN services)
- Takedown of a complete system (e.g., by limiting the attack to certain services)

Conclusion: All in all, what we see as the real added value of SDN to security is the ability to interact with switches and routers by means of APIs. These APIs can be leveraged for a number of security-related tasks, independently from the complete adoption of the SDN paradigm.

5.4 Too novel to be applied or the way out of security ossification?

Tobias Limmer (Siemens AG – München, DE)

Joint work of Marc Eisenbarth, Felix Erlacher, Frank Kargl, Thomas Kemmerich, Tobias Limmer, Ramin Sadre, Sebastian Schmerl, Bettina Schnor, Radu State, Alexander von Gernler

SDN is a novel technology and may solve several problems that are surfacing in current network topologies. Increasing heterogeneity, caused by new initiatives such as Bring Your Own Device (BYOD) or developments in the area of Internet of Things (IoT), or highly dynamic network changes required by virtualization are just a few examples.

To control the effects of those new developments, more fine-grained control is necessary as is currently supported by legacy networking equipment. For example, the augmentation of traditional firewalls that allows them to examine and filter intra-subnet traffic may help to protect potentially untrusted endpoints from each other. SDN supports this use case by introducing a common transparent interface to networking devices for network security mechanisms. Using this standard interface, software and devices from different vendors may become interoperable and may be managed within one environment. However, the current state of available standards, such as OpenFlow, is not promising here. It can be easily seen that those standards and related regulations are still immature, as important parts are not defined yet. In the case of OpenFlow, northbound interfaces are not standardized yet, and available network apps typically disregard security completely.

The new possibilities in the security area are based on the flexible architecture of SDNs. This fact results in configurations and network topologies that may become very complex. From a technical point of view, a diverse set of problems arises here: SDNs usually should distribute components within the network to ensure reliability. What happens if multiple controllers issue conflicting instructions to network devices? In what way should controllers prevent problematic situations caused by multiple interacting networking apps that have been downloaded from a central app store? What happens if a network is segmented in multiple parts, and newly appearing devices need to be boot-strapped to be integrated into the network? Many security applications within SDNs also rely on packet forwarding to centralized components which may analyze those packets. On the one hand, SDNs are supposed to make a network more efficient, but on the other hand, new features may lead to uncontrolled network link congestions which may require even higher data rates compared to traditional networks. The complexity of SDNs may also impact compliance certifications in the banking sector or safety regulations in the area of Operational Technology (OT). These questions are still largely unresolved and need to be addressed before SDNs are deployed in this flexible operation mode.

Still, many large Internet companies and ISPs show much interest in SDN deployments, and several of those make already use of SDNs. In the current state, much expert know-how and many customizations are necessary to successfully deploy SDN and benefit from its features. Facebook, as an example, already has an SDN-based deployment method for big data centers. ISPs may benefit from a common framework of all network devices which supports a common language to express network policies and rules. This would allow providers to simplify policy compliance and configuration, and may even open new business opportunities such as customers who could upload apps to their provider's infrastructure for customized network features, such as DDoS protection, QoS, or packet filtering. Due to open standards and one common environment, those network apps can be sandboxed by the underlying controller, allowing to separate network logic and security.

Instead, we may also continue to rely on proven and well-established security technologies like firewalls or intrusion detection systems that we know how to handle. If network topology and devices are chosen carefully, most of the features that can be realized with SDNs are also available within traditional networking environments. Furthermore, SDNs will only be able to fully automatically control and manage the simplest networks – customization and management by network experts will still be necessary in many cases. But what about networks that constantly face changing requirements from the business side, technical problems caused by evolved network topologies with devices from different vendors in different versions? Here, SDN may provide a solution due to its capabilities to standardize interfaces and features across vendors and network devices.

5.5 Is SDN more complex or simpler?

Claas Lorenz (genua GmbH – Kirchheim bei München, DE)

Joint work of Dieter Gollmann, Peter Herrmann, Hartmut König, Claas Lorenz, Michael Menth, Björn Scheuermann

The concept of SDN promises a reduction in complexity by splitting networks into a dedicated data plane and a logically centralized control plane. When explaining concepts like routing, the software approach in SDN seems much more simple than the distributed algorithms and protocols in classical networks, since it can just be represented as a simple graph problem. This narrative is stressed by two aspects that are hidden in the simplicistic model of SDN regarding the controller as a single entity rather than a distributed system. The need for scalability and operational requirements, e.g., concerning fault tolerance, enforce a distributed approach. Additionally, the realization of the control plane completely in software raises issues about its algorithmic complicateness. This is due to the additional requirements that were not imposed on classical networks, but are now thinkable in SDN. While this is a unique selling point in terms of possible features, it raises serious concerns for security, as it opposes simplicity which is a key design principle for building secure systems.

State-of-the-Art controller implementations suffer a tremendous feature bloat which is most likely buggy and rather untested. The same problem occurs with switches, which are often legacy equipment,are enriched with an OpenFlow interface. The simplicity, as intended by the SDN paradigm, is not very common in practice which might be a result of the consortial standardization model leading to hard fights between financially potent parties and feature rich compromises in standards and implementations. Nevertheless, there exist industry-grade whitebox switches as well as simple, lightweight controller implementations. For the price of providing less features, the realization of SDN using simple and possibly less attackable components is possible. Nevertheless, if an advanced feature set is required the controller must be designed as a distributed network operating system with security enforcement mechanisms in place analogous to traditional operating systems. An example trait would be the distinction between a kernel and a user land with well-defined interfaces and access control.

Flows as data model in a switched network are much simpler notions than layered packets in traditional routed networks. This may help to define a general structural core while providing powerful functionality. If this core could then be standardized and implemented very narrowly it is likely to be well designed, broadly tested, and hardened properly. On the other hand, the separation of data and control plane creates different views and, with emphasis on their consistency, makes the creation of a wholistic security solution a tough challenge. Even though, this distinction makes the decomposition of components easier and therefore better testable. Also, security patches for the control plane become more feasible.

Besides the defense of the SDN itself, it can be used to simplify mitigation of attacks that are commonly seen in classical networks. Attacks like ARP flooding or DHCP spoofing can be tackled in a simple and effective manner with SDN. In addition, every switch may provide firewalling functionality helping to achieve a defense in depth.

All in all, SDN introduces numerous challenges regarding complexity and simplicity of the system. It has the potential to be simple but making it simple is quite complex. The decomposition of components is easy, but their secure reassembly remains challenging. Therefore, a self-limitation regarding the necessity of features must be taken into consideration to allow a simple and secure design, implementation, and operation of a Software-Defined Network.

5.6 The Good and the Bad of Centralization in SDN

Christian Rossow (Universität des Saarlandes, DE)

Joint work of José Jair C. de Santanna, Issa Khalil, Evangelos Markatos, Michael Menth, Christian Rossow

By design, SDN centralizes many networking aspects that traditionally might have been decentralized. For example, SDN-driven networks may offer a centralized location to access or steer the data, control, and management plane. Furthermore, SDN-driven networking algorithms can assume a centralized data model, which was not possible in traditional networking. Since this is a radical change in the way we think of networks, we have investigated in our working group pros and cons implied by the centralization aspects of SDN.

First, a centralized architecture and network management creates a single point of failure which downgrades the resilience given by a distributed system. It is debatable whether traditional networks do not already offer single point of failures, but SDN adds some additional centralization points that might be exploited by an (i) internal attacker that suddenly has a central place to monitor and manipulate the network or (ii) by an external adversary that compromises vulnerable SDN components. This requires further thoughts on how SDN can be protected against such attacks.

Second, it may happen that the centralized decision engine of SDN adds a new type of denial-of-service (DoS) vector. For example, an attacker might be able to overload the controller with unknown flows that require constant decision makings. On the other hand, the centralization of SDN allows to more effectively tackle existing types of DoS attacks, as it has a global view of the network topology and can correlate this information with the traffic analysis for more reliable attack detection results. The two areas bear interesting research questions that should be investigated further.

Third, an important aspect is how SDN signaling is organized, in-band or out-of-band. If both the data and the control plane share the same (physical or logical) network segment (in-band signaling), the control plane may also become corrupted if the data plane breaks. As a consequence, out-of-band signaling schemes should be explored further to allow an easier recovery.

Fourth, scalability is a key feature of centralized systems. SDN involves a few critical parts that may become bottlenecks, however. For example, the flow tables may fill, so that the hierarchy of the networks requires careful thinking. In addition, if a layer of redundancy or load balancing is added (e.g., in terms of multiple controllers), suddenly there is the need for communication to avoid any possible state or decision inconsistencies. These aspects motivate further research how the centralized parts should be designed in a scalable fashion.

Fifth, although SDN increases the network complexity and the plentitude of intertwining algorithms may emit possibly contradicting policies, we are convinced that it is especially the centralization that plays in our hands to resolve such inconsistencies. Reacting to and removing such policy inconsistencies is much easier in a centralized network, such as SDN. This has positive implications on many types of policies, such as centralized routing algorithms, firewalls, or network monitoring methodologies.

To sum up, the centralization imposed by SDN indeed creates new challenges, but the benefits are clearly predominant. However, it is important to address the open research questions in this regard to ensure security and resiliency of the centralized SDN aspects.

6 Working Groups: Research Directions

6.1 Research Directions: Methods, Policy, and Attacker Model – Assessing and Improving the Security of SDN Networks

Georg Carle (TU München, DE)

Joint work of Georg Carle, Sven Dietrich, Dieter Gollmann, Bettina Schnoor, Peter Herrmann, Christian Röpke

When assessing suitable approaches for specifying security goals for SDN, it was identified that existing methods include natural language approaches, such as the ones used in ISO 27000, Common Criteria, BSI Base Protection Catalogue, and also formal approaches, as part of Linux iptables, Unified Modeling Language (UML), Security Policy Languages, and BAN logic from the protocol analysis field. It was identified that an important goal is to automatically derive secure SDN configurations. That requires extensions of the state-of-the-art methods, by providing additional information elements for the full range of components of SDNs, representing all states of SDN network elements. There is also a need for new tools that are capable with dealing with this additional information.

Methods to assess the security of SDN networks range from penetration testing to formal analysis, such as using policy checkers. Penetration testing has several limitations, such as the limited coverage of the system. It may also be difficult to identify the problems that tests do not find. In particular, the outcome of various tests may depend on the state a specific SDN component is in, which may depend on past input via different network interfaces. Policy checkers allow one to identify a case where a set of policy rules violates a set of security policies. However, if the policy set is incomplete, it is possible that certain violations would not detected. On the other hand, with penetration testing such violations that are not detected by formal methods may indeed be detected.

When assessing what current policy checkers cannot detect in SDN networks, it was identified that concurrency violations are an important problem in SDN, as this may lead to policy or invariant violations, such as blackholes, forwarding loops, or non-deterministic forwarding [1].

Methods to provide a trust base for SDN include providing a security kernel inside the SDN controller [2], which are able to distinguish between various types of SDN controller applications. For example, in the case of coexistence of a firewall and a load balancer application on the controller, the firewall application would have priority over the load balancer application.

Concerning relevant attacker models, it was identified that related work, such as [3], provides a highly useful taxonomy of attacker models. In order to prevent that possible attacks may be successful, one can consider an approach in which the different states a network may be distinguished. That means identifying good states in which the known attack cannot be successful while avoiding bad states. For the latter bad states, it is known that attacks can be successful.

Overall, it was identified that the differences of SDN to conventional networks make it very hard to ensure the security of SDNs. This is a consequence of the additional complexity of SDN, in which controllers change the configuration of the switches, allowing for a variety of automated reconfigurations. This makes attacks possible in which an attacker causes a reconfiguration to occur that leads to the desired outcome. For example, an attacker may create legitimate but dubious traffic, thereby causing the controller to regularly reconfigure the switches.

All approaches that allow one to handle the increased complexity are considered to be highly useful. They ensure that certain SDN applications can only influence certain flows. By applying the concept of network isolation, SDN enables network slicing and Virtual Network Operators.

References

1 Ahmed El-Hassany, Jeremie Miserez, Pavol Bielik, Laurent Vanbever, and Martin Vechev. *SDNRacer: Concurrency Analysis for Software-Defined Networks*.PLDI'16, Santa Barbara, CA, USA June 2016

2 Phillip Porras, Seungwon Shin, Vinod Yegneswaran, Martin Fong, Mabry Tyson, Guofei Gu. *A security enforcement kernel for OpenFlow networks.* First workshop on Hot topics in software defined networks HotSDN 2012, Helsinki, Finland, August 2012

3 Diego Kreutz, Fernando M. V. Ramos, Paulo Veríssimo. *Towards secure and dependable software-defined networks.* Workshop on Hot Topics in Software Defined Networking HotSDN 2013, Hong Kong, August 2013

6.2 Research Directions: Secure Operations in SDN-based Environments

Marc C. Dacier (QCRI – Doha, QA)

Joint work of L. Jean Camp, Marc C. Dacier, Jan Kohlrausch, Tobias Limmer, Michael Meier, Simin Nadjm-Therani, Ramin Sadre, Thomas Scheffler, Sebastian Schmerl, Radu State

On Thursday, September 8, 2016, one of the themes debated by the participants in a parallel session was oriented towards the issues on how to securely operate an SDN-based environment. It led to a very lively discussion for several hours, the gist of it is summarised here below.

Before thinking of operating an SDN environment, a key question discussed by the team was related to the rolling out of SDN in an existing environment. There was a consensus to say that it was unlikely that (i) SDN would completely replace an existing, non SDN based, environment and that (ii) any deployment would have to take place in an incremental way. In both situations, namely transient phase of deployment and ongoing operation of a mixed environment (SDN and non SDN), it was felt that specific security concerns would have to be addressed since the promises of an homogeneous, well defined, centrally controlled SDN environment would not be present. There was the feeling within the group that such operational concerns were not properly addressed by existing solutions yet and that it would deserve some further research to lead to practical solutions.

The group generally agreed that SDN would not replace but instead complement the networking toolbox at the disposal of operators. Two specific use cases were discussed where SDN was seen as a, possibly, useful paradigm to use. The first one was related to the emerging "Bring Your Own Device" paradigm (BYOD) in which potentially compromised devices were dynamically added to the networking infrastructure. The need for a simple and clear mechanisms to enforce well defined policies for such devices was an argument in favour of an SDN environment. Indeed, if well done, SDN could be used to automatically implement concepts such as the quarantine of misbehaving devices, degraded – or fail safe – modes for the network in case of worm propagations, adaptive scrutiny of network flows to look for data exfiltration, etc. . .

The second use case discussed by the group was related to critical infrastructures or, more generally, so called "Operational Technology" (OT) environment, as opposed to "Information Technology" (IT). It was noted that, nowadays, whereas the OT department was in charge of running the OT infrastructure, its security still usually felt under the responsibility of the IT department. It was observed that in such deployment, SDN could help the IT department in improving the limited visibility they currently have and would make it easier for them to enforce, at the networking level, the needed security policies. A contrario, it was also acknowledged that OT environments are quite resistant to changes and a convincing argument had to be brought forward to implement such radical change which would, quite likely, require to replace most, if not all, routers and switches in these environments.

More generally, it was felt that, whereas SDN clearly has some claimed benefits, there was a need for a thorough economic study of the pros and cons which would take into consideration the possible negative effects on security and the supplemental costs associated with a reinforcement of the needed security tools.

The human dimension of the SDN impact on security was also discussed. Not only in the way its deployment could be a bridge between the IT and OT worlds, as discussed before, but also with the increased risks created by giving a lot of powers to the few (or sole?) administrators of the SDN controller. As we see more attacks due to insider, it was agreed that the risk of having a malicious administrator was not to be neglected and to be dealt with but more research was required to come up with a satisfactory solution. Along the same line, there was some fear expressed that the possibility of having various kinds of applications running in the controller to serve different purposes could lead to some serious organisational disputes if not properly anticipated. For instance, if two distinct departments (e.g. marketing and IT security) want each to have their own application in the controller, built on distinct requirements (e.g. quality of service vs. security), who would (i) detect possible inconsistencies between decisions made by these applications and (ii) decide which one to favour?

The problem of various applications, designed and developed by independent teams, running in the same controller is a very large problem that has been discussed at length by the team. It came out that there is a clear need for more research to be done in order to help the people running SDN platforms to decide not only if (i) a given application is secure in the first place (i.e. without any vulnerability, and not malicious) but, more importantly, if (ii) the addition of a new application to a controller where other applications are already running would not create security issues due to the composition of the decisions made by each application independently. Is it possible to prove, by construction, that, assuming each application is "secure", the software resulting from the composition of all these applications remains secure? This was seen as an important open research area.

Finally, it was expected that most of the problems that the domain of network operational security has been dealing with in the past would, could or should be revisited in the sense that the introduction of SDN was changing the attack surface that people had been used to consider when looking at distributed systems. For instance, the existence of a common controller used for two networks separated by a firewall could open the door for new techniques to circumvent the firewall (if SDN was not correctly configured). More generally, the presence of such common controller could be seen as a new way to implement well known covert channels. Also, an SDN environment, if not very securely configured, would offer lots of opportunities for new ways to launch denial of service attacks, to avoid detection by deep packets inspection devices etc.

All in all, it was felt that SDN could certainly help in improving the operational security of a network environment but that many problems remain unsolved (i) to ensure that a given SDN environment would be secure by construction, (ii) to prevent malicious users (especially administrators) or applications from misusing such environment and (iii) to detect when such misuse would occur.

6.3 Research Directions: SDN-based Security

Frank Kargl (Universität Ulm, DE)

Joint work of Johanna Amann, Kpatcha Mazabalo Bayarou, José Jair C. de Santanna, Radoslaw Cwalinski, Marc Eisenbarth, Felix Erlacher, Marko Jahnke, Mattijs Jonker, Frank Kargl, Evangelos Markatos, Rene Rietz, Christian Rossow, Robin Sommer, Jens Tölle

The working group discussed how SDN would enable new forms of network security mechanisms to be envisioned, designed, and implemented, or how SDN would allow existing mechanisms to be implemented in a more flexible or interoperable way. For this, we first identified typical attacks where we assumed a potential for SDN-based mitigation mechanisms. Attacks we discussed included DDoS, reconnaissance, Man-In-the-Middle, malicious modifications of the network including any accidental misconfigurations, and malware-related attacks that we spread into initial infection, internal spread, Command & Control (C&C) communication and data exfiltration.

We then created a table where all these attacks were listed in relation to the common categorization of security mechanisms in prevention, detection, reaction, and forensics. For each of the resulting cells, we discussed how SDN would support or hinder the design of such security mechanisms.

The discussion results are depicted in Table 1. For the purpose of this text, we will only address what participants considered the most interesting ideas. In general, we identified that SDN enables mostly two types of capabilities that security mechanisms may make use of.

First, SDN and OpenFlow allow holistic control of network devices throughout all active network components. With this, mechanisms that inspect or filter traffic anywhere in the network become possible. Second, SDN offers a standardized interface for interacting with the network which would allow cross-platform security mechanisms that are not tight to a specific vendor.

For DDoS attacks, it should probably be investigated further how fine-grained filtering throughout the own network can help to either prevent such attacks or react to such attacks and filter out attack traffic and how this may be more effective than central filtering. However, this is probably mostly effective for egress filtering and therefore mitigating attacks that originate from your own network. Beyond, if we foresee the notion of "network apps", these may also be used to implement mitigation logic for a specific attack on your network. This mitigation logic could then be deployed in the network of your ISP in order to have a highly specific, fine-granular and customized filtering being created by the network operator executed within all ISP's devices.

Reconnaissance attacks may also be easier to detect with SDN. The assumption is that there is often no fixed central place to detect such attacks. Particularly if they stem from internal nodes, applying an IDS on your Internet gateway will not be effective and you would therefore deploy your IDS on many places inside your network. This may require substantial resources. We came up with the notion of Network Function Virtualization (NFV) of network security mechanisms like firewalls or IDSs/IPSs that would run on a central cloud server or on cloud servers distributed in the network. You would then use the SDN functionalities to pre-filter traffic and forward the resulting streams or packets to the IDS for inspection. If there are suspicious activities being detected, you may even reduce filtering to inspect the traffic more intensively. Beyond, NFV of network security mechanisms in cloud datacenters would allow migrating the IDS or firewalls that monitor a certain critical virtual machine together with that virtual machines.

Regarding Man-In-the-Middle attacks, we discussed that SDN would allowing to quickly react to such attacks once they are detected. Hosts running such an attack could be quickly isolated and then investigated by forensic mechanisms. Regarding malicious or accidental modifications in the network, we think that SDN could help by having a central point where network configuration (including open flow tables) is accessible. Then, detecting inconsistencies and applying plausibility checks to this network state would allow detection of malicious modifications to routing, identification of unauthorized hosts, changes to network topology and many more such attacks.

At the same time, we also acknowledged that applying SDN in your network will, in general, make the configuration and the state of your network much more complex and thus detecting such attacks in the first place will become much harder. This is a general problem for network security in SDN-enabled networks: due to the high volatility and fine granularity of network configuration, it may be substantially harder to detect attacks. This applies also to other parts of this discussion, like the Man-In-the-Middle detection.

Regarding malware, we again identified a potential for applying NFV to have mechanisms like malware scanning or IDS being applied flexibly and scalable in the network. So if there is a malware outbreak and spread in one part of the network, resources can be allocated on your cloud servers to inspect particularly that traffic in that network segment. Likewise, if you have critical resources that get relocated to different parts of the network as part of cloud operations, the network security mechanisms may migrate together with them.

Next, SDN may also support easier containment of malware infections and spread. You may easily segment your network, e.g., triggered by the IDS or virus scanner having detected infections on some host. One idea was to even simulate the possible spread of a malware based on known SDN state. So if a malware is known to spread via a certain protocol, one could simulate which other hosts are reachable in a transitive way and then apply more stringent filtering and isolation to those hosts that are potentially infected.

Finally, malware may also be addressed by using SDN mechanisms to redirect the communication of infected machines with their C&C servers. This so called sinkholing would allow to redirect traffic to C&C's IP addresses to a security host where that traffic can be forensically analyzed, filtered, or even modified, e.g., to issue instructions to infected hosts. This will also allow to gather detailed statistics on infected machines.

Independent of those attacks, we also came up with the idea to use the isolation capabilities of SDN to create islands of personal devices within a network. Thus, all devices belonging to the same user – smartphones, tablets, smartwatches, laptops, etc. – would sit within the same island and could freely communicate with each other, including broad- and multicast discovery protocols, while external communication could be subject to a consistent security policy for that specific user. Overall, we considered SDN to be an interesting enabler for security mechanisms and could come up with a whole series of concrete ideas that we think would merit further investigations in future research projects.

Table 1 SDN-enabled security mechanisms.

	Prevent	**Detect**	**React**	**Forensics**
DDoS	Fine-grained filtering	Offloading certain filtering/detection operation at the switch level to be able to operate at line rate while extending inspection at more than netflow information	Using the whole network to react	Statistics, logging and packet inspection for better understanding how the DDoS works
Reconnaisence	(1) Using the whole network for filtering (2) hiding the network structure	Network Function Virtualization (NFV) for IDS, honeypot on-demand	NFV for IDS, honeypot on-demand (e.g., virtual deployment of a honeypot)	Statistics, logging and packet inspection
MITM (not at the application layer)	Fine-grained traffic control	(1) Detecting routing anomalies (may be harder in the presence of SDN, due to increased complexity) (2) Detecting forwarding correlations (also possible before SDN)	Quick isolation	(1) Negative: increased number of more complex states (2) Implement MITM for inspection
Misconfigurations & malicious modifications	Global policy with SDN	Consistency and plausibility checking on flow tables becomes more difficult due to increased complexity	Probing of network behaviour of dedicated resources (e.g., isolation of errors)	(1) Statistics, logging and packet inspection (2) Checking network invariants
Malware (initial) infection	NFV for virus scanner	(1) Using whole network for detection (2) NFV for IDS	IDS/quarantining potentially infected hosts	Logging, network-wide view to identify where the attack came from
Malware spread	(1) Pervasive possibility for isolation/segmentation (2) Segmentation may disrupt some services (e.g., NetBIOS)	(1) Using the whole network for detection (2) NFV for IDS	IDS/quarantining potentially infected hosts	Simulation of malware spread (feedbacks to better prevention and reaction)
Malware C&C	Sinkholing C&C	Netflow-like analysis	Sinkholing C&C	Redirecting C&C traffic for analysis
Malware data exfiltration		Detecting “NSA style” keyword exfiltration based on SDN logs	Modification/marking exfiltrated data	

6.4 Research Directions: Secure Architecture for SDN

Alexander von Gernler (genua GmbH – Kirchheim bei München, DE)

Joint work of Paulo Jorge Esteves-Veríssimo, Alexander von Gernler, Thomas Kemmerich, Issa Khalil, Boris Koldehofe, Hartmut König, Claas Lorenz, Björn Scheuermann, Han Xu

The working group dealt with the topic to find a secured architecture for SDN. Based on an exemplary diagram of an SDN setting we tried to identify security issues concerning single components, links, or functional elements of the SDN setting. We discussed whether there are applicable architectural patters and best practice experience. All participants agreed that there is a need for such architecture, but the time was too short to find a conclusive proposal. A solution of this problem requires deeper and long-term research. In our discussion, a number of questions have been raised which require further research activities. Among these were:

1. How to securely implement and deploy "network apps"? How to design the northbound interface so it is secure and expressive?
2. Complexity is an important issue in SDN. How can SDN solutions be simplified? How can SDNs scaled securely?
3. How to implement access control and authorization in SDN networks?
4. How can we protect the controller itself?
5. How can we secure the communication between controller & switches?
6. How can we perform intrusion detection and anomaly detection in SDNs?
7. How can we perform intrusion detection / resp. achieve SIEM functionality in the SDN context?
8. How differently do we have to deal with misbehaving/malicious clients?
9. How can we deal with misbehaving/rogue applications?
10. How to mitigate attacks?
11. What is the role of trusted hardware in switches? Is it needed for strong security?
12. How can you operate SDN in presence of untrusted HW components?
13. How do we ensure the software quality of the SDN infrastructure (controller, HW, ...)?

7 Final Plenary Discussion

7.1 Theses on SDN security

Hartmut König (BTU Cottbus, DE) and Radoslaw Cwalinski (BTU Cottbus, DE)

Joint work of all seminar participants

In the final plenary session of our seminar, the participants formulated the following theses regarding the security of SDN.

1. SDN is hard to define, one needs to be clear about assumptions and goals. SDN feature consolidation will come, but is not yet foreseeable.
2. The main advantage for SDN deployment will not be security. However, SDN creates a lot of security problems, many of which do not have a clear solution.

3. On the other hand, SDN enables new creative forms of security mechanisms – without being mandatory for them. Reaction possibilities to security incidents can be enhanced. One can use SDN for security even without full deployment of SDN in the network.
4. SDN security solutions demand a holistic approach including trusted computing base in network component. Secure software engineering will become more relevant for networks with SDN. Securing SDN, in particular network apps, requires substantial progress in software security and other fields, such as access control and policy definition.
5. Simple SDN solutions foster SDN security, but keeping SDN simple is complex!
6. Centralized controllers create many internal security challenges, e.g., “Packet INs” are considered harmful. More static uses of SDN are better for security.
7. There is no clear SDN/OpenFlow security roadmap.
8. Without security, SDN will not succeed!

Participants

- Johanna Amann
ICSI – Berkeley, US
- Kpatcha Mazabalo Bayarou
Fraunhofer SIT – Darmstadt, DE
- José Jair C. de Santanna
University of Twente, NL
- L. Jean Camp
Indiana University – Bloomington, US
- Georg Carle
TU München, DE
- Radoslaw Cwalinski
BTU Cottbus, DE
- Marc C. Dacier
QCRI – Doha, QA
- Hervé Debar
Télécom & Management SudParis – Evry, FR
- Sven Dietrich
City University of New York, US
- Falko Dressler
Universität Paderborn, DE
- Marc Eisenbarth
Arbor Networks – Waco, US
- Felix Erlacher
Universität Innsbruck, AT
- Paulo Jorge Esteves-Veríssimo
University of Luxembourg, LU
- Dieter Gollmann
TU Hamburg-Harburg, DE
- Peter Herrmann
NTNU – Trondheim, NO
- Marko Jahnke
CERT-BPOL – Swisttal, DE
- Mattijs Jonker
University of Twente, NL
- Frank Kargl
Universität Ulm, DE
- Thomas Kemmerich
Norwegian University of Science & Technology, NO
- Issa Khalil
QCRI – Doha, QA
- Hartmut König
BTU Cottbus, DE
- Jan Kohlrausch
DFN-CERT Services GmbH, DE
- Boris Koldehofe
TU Darmstadt, DE
- Tobias Limmer
Siemens AG – München, DE
- Claas Lorenz
genua GmbH – Kirchheim bei München, DE
- Thomas Lukaseder
Universität Ulm, DE
- Evangelos Markatos
FORTH – Heraklion, GR
- Michael Meier
Universität Bonn, DE
- Michael Menth
Universität Tübingen, DE
- Simin Nadjm-Tehrani
Linköping University, SE
- Rene Rietz
BTU Cottbus, DE
- Christian Röpke
Ruhr-Universität Bochum, DE
- Christian Rossow
Universität des Saarlandes, DE
- Ramin Sadre
University of Louvain, BE
- Thomas Scheffler
Beuth Hochschule für Technik – Berlin, DE
- Björn Scheuermann
HU Berlin, DE
- Sebastian Schmerl
Computacenter – Erfurt, DE
- Bettina Schnor
Universität Potsdam, DE
- Robin Sommer
ICSI – Berkeley, US
- Radu State
University of Luxembourg, LU
- Jens Tölle
Fraunhofer FKIE – Wachtberg, DE
- Alexander von Gernler
genua GmbH – Kirchheim bei München, DE
- Han Xu
Huawei Technologies – München, DE
- Emmanuele Zambon
SecurityMatters B.V., NL