Cases for Including a Reference Monitor to SDN

Dimitrios Gkounis, Felix Klaedtke, Roberto Bifulco, Ghassan O. Karame
NEC Laboratories Europe, Germany – <firstname.lastname>@neclab.eu

1. Motivation

Misconfigurations are considered a main source of network failures [1,2]:

- **SDN simplifies** and **automates** network operations, potentially reducing misconfigurations
- But…

Misconfigurations can still happen in SDN:

- Potentially **bigger impact** due to centralized controller access to the **entire network**
- Implementation **bugs** (controller, applications)
- Applications with **competing objectives**
- **Malicious** entities (e.g., compromised application)

Related Work: Data plane verification techniques [3] and checks at controller-application interfaces [4] **do not block the controller** sending configuration rules

2. Solution

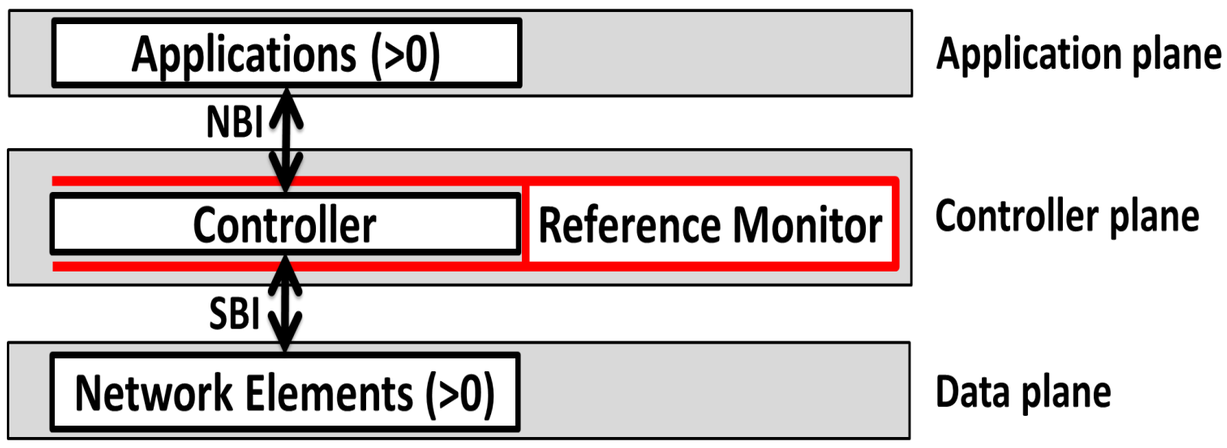
A **reference monitor** for SDN [5]:

- **Trustworthy** component
- **Limits** controller **actions** according to **policies**
- Similar to reference monitors in operating systems

3. Architecture

The **reference monitor** is an entity that runs independently from the controller:

- It hooks into the controller's **SBI** and **NBI**



NBI hook: Retrieves application information

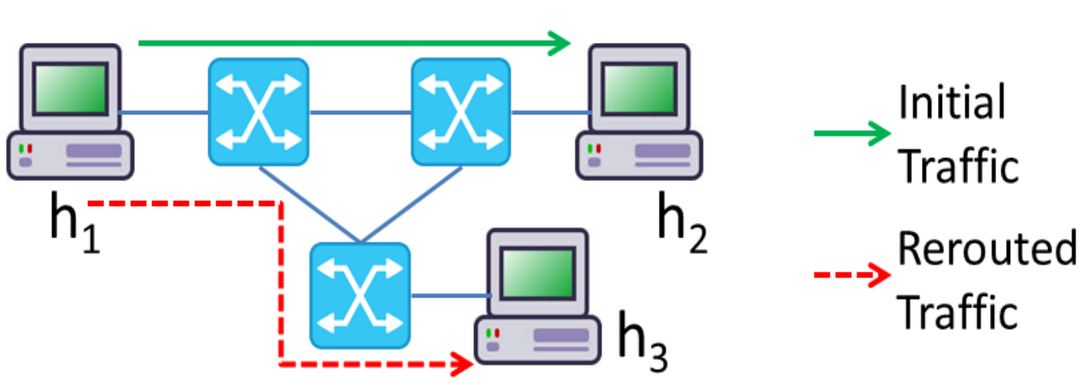
SBI hook: Intercepts SBI calls and blocks them if not policy-compliant

Proof-of-concept implementation for **ONOS**:

- Separate OSGi bundle
- **Minimal modifications** to **ONOS**
- SBI changes:
  - All calls intercepted by the reference monitor
  - Include application information (app ID)

Demo

Competing applications and careless administrators can misconfigure the network



Two forwarding applications on top of **ONOS**:

- Default forwarding application
- Command line interface (CLI) to submit intents:
  - Intents able to connect hosts or redirect traffic

Connectivity requirements:

- Connectivity between h_1-h_2 should be provided by the forwarding application
- The CLI should only be used to connect h_1-h_3

Without the **reference monitor**:

- **CLI can submit intents that "hijack" traffic between h_1 and h_2**

With the **reference monitor**:

- **Provided policy** is enforced
- **OpenFlow messages** violating the policy are **blocked**, e.g., whitelist of applications and network hosts accessing network elements

A second use case about bug prevention is described in the extended abstract

Acknowledgements: This work received funding from the European Union in the context of the H2020 projects VirtuWind and 5G-Ensure (grant agreements 671562 and 671648)

www.virtuwind.eu www.5gensure.eu

References

[1] www-935.ibm.com/services/au/gts/pdf/200249.pdf
[2] J. Sherry et al. Making middleboxes someone else's problem: Network processing as a cloud service. CCR, Oct '12.
[3] P. Kazemian et al. Real time network policy checking using header space analysis. NSDI '13.
[4] S. Shin et al. Rosemary: A robust, secure, and high-performance network operating system. CCS '14.
[5] F. Klaedtke et al. Access control for SDN controllers. HotSDN '14.