OpenFlow: A Security Analysis

Rowan Klöti[1], Vasileios Kotronis[1] and Paul Smith[2]

paul.smith@ait.ac.at

[1]ETH Zürich

[2]AIT Austrian Institute of Technology

MSN 2013, Cosener's House

11th July, 2013

bmvit

Federal Ministry for Transport, Innovation and Technology

50,46%

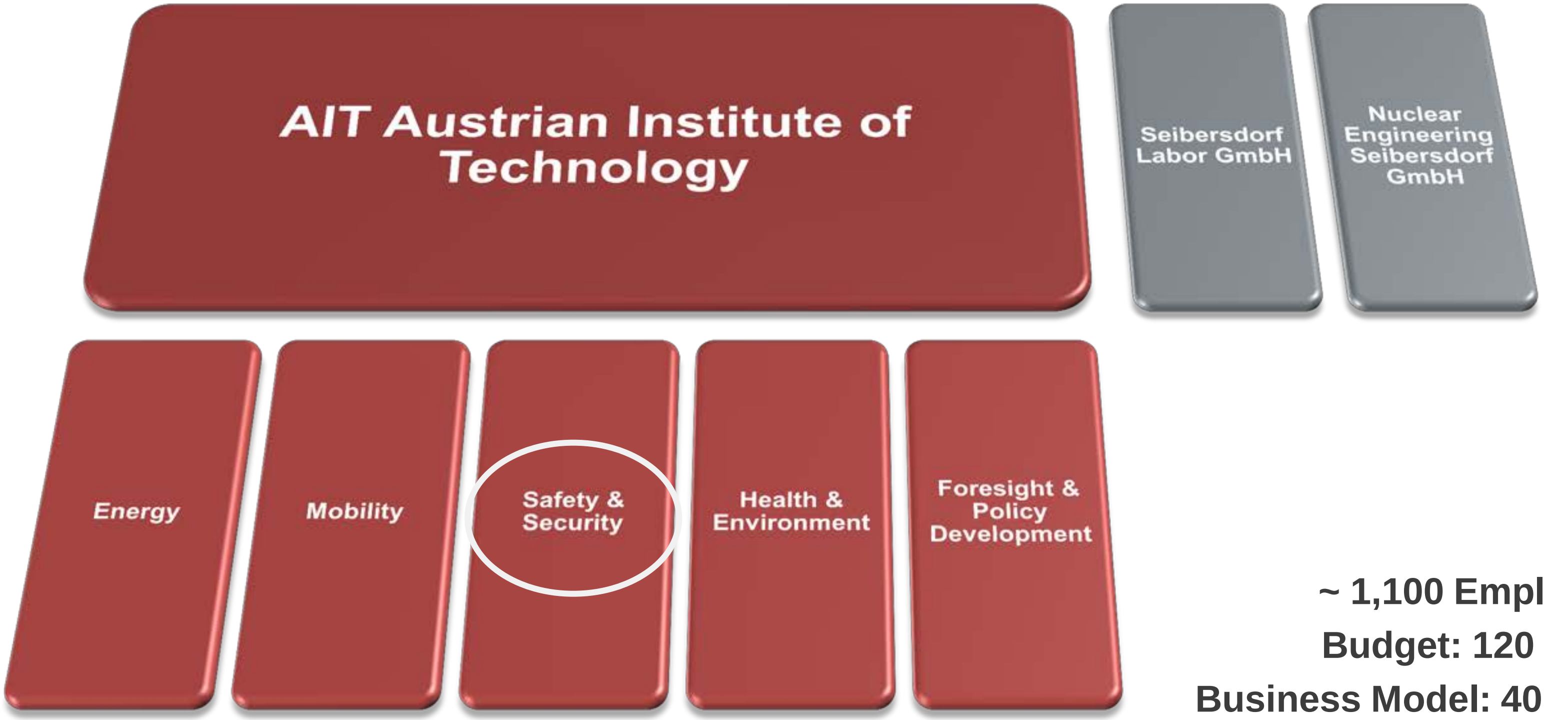
Austrian Industry

Federation of Austrian Industries

49,54%

SEIBERSDORF LABORATORIES

NUCLEAR ENGINEERING SEIBERSDORF



AIT Austrian Institute of Technology

Seibersdorf Labor GmbH

Nuclear Engineering Seibersdorf GmbH

Energy

Mobility

Safety & Security

Health & Environment

Foresight & Policy Development

~ 1,100 Employees

Budget: 120 Mio. €

Business Model: 40:30:30

Reference Projects and Themes

(SG)²
Smart Grid Security Guidance

Smart Grid Security

PRECYSE

Critical ICT Infrastructure Security
https://www.precyse.eu

seccrit
SEcure Cloud computing for CRitical infrastructure IT

Cloud Computing for high-assurance applications
http://www.seccrit.eu

PARIS

Privacy aspects
http://www.paris-project.org/

CAIS

National Cyber Defence

FastPass

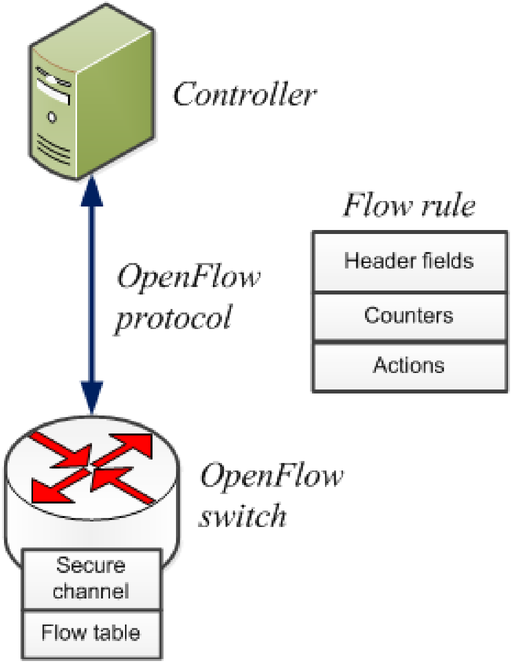
Future Border Control
https://www.fastpass-project.eu/

- Themes: anomaly detection, privacy by design, risk assessment and management, secure and resilient architecture analysis and design, ….

Software Defined Networks and OpenFlow

- Software Defined Networks (SDNs) separate data and control plane

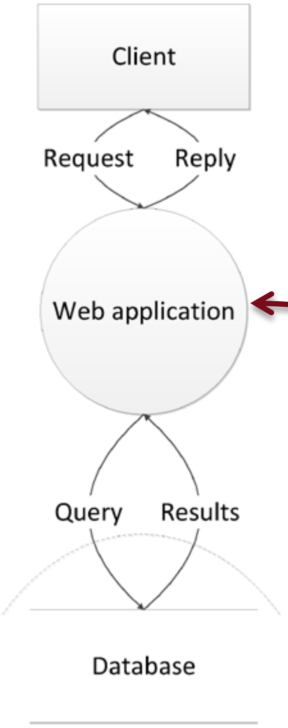
- OpenFlow is the canonical implementation of SDNs
  - Switch implements the *data plane*
  - Controller implements the *control plane*
  - Switch and control connected with a *secure channel*
  - Controller installs *flow rules* on the switch
  - Flow rule *header fields* match packet headers
  - Packets matching flow rules have *actions* performed on them



- No existing security analysis of OpenFlow has been carried out, identifying vulnerabilities and threats

Security Analysis Method

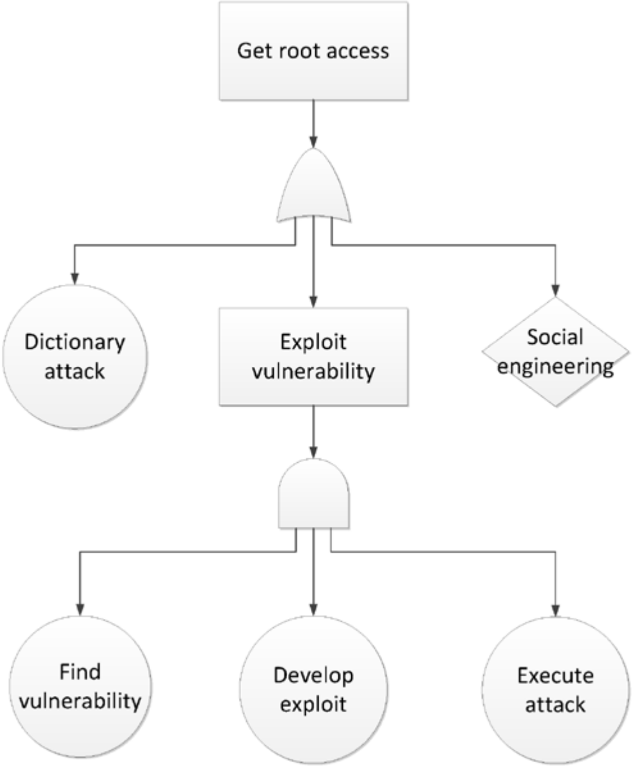
Microsoft
STRIDE Methodology



[component, vulnerability]

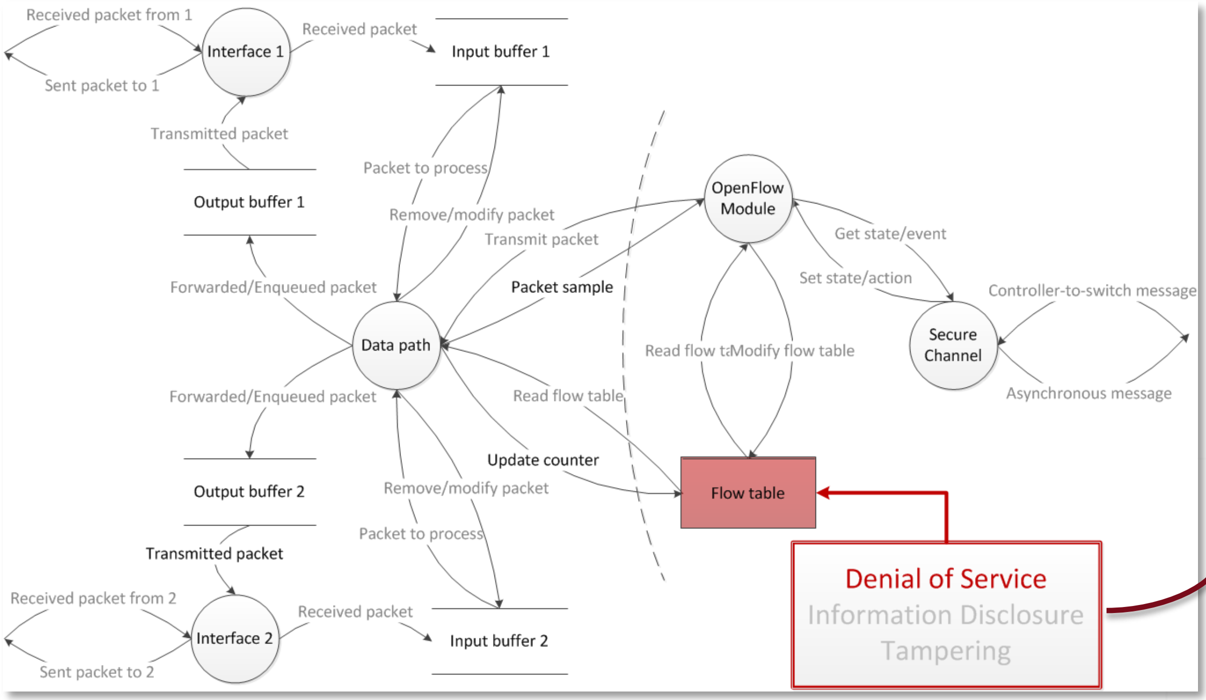
Spoofing
Tampering
Repudiation
Information disclosure
Denial of service
Elevation of privileges

Attack tree analysis

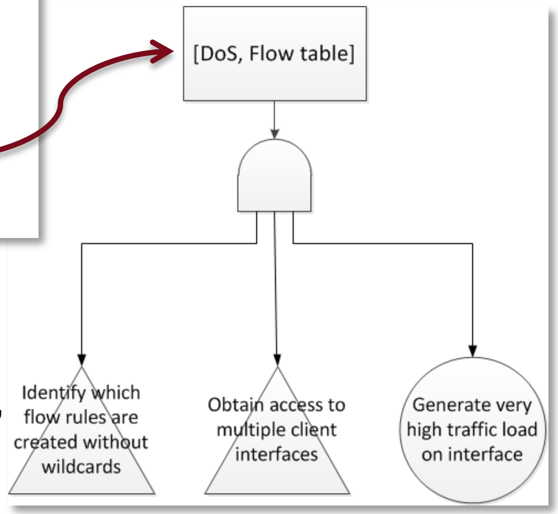


Security Analysis Highlights

OpenFlow switch Data Flow Diagram



Attack tree



[DoS, Flow table]
Identify which flow rules are created without wildcards
Obtain access to multiple client interfaces
Generate very high traffic load on interface

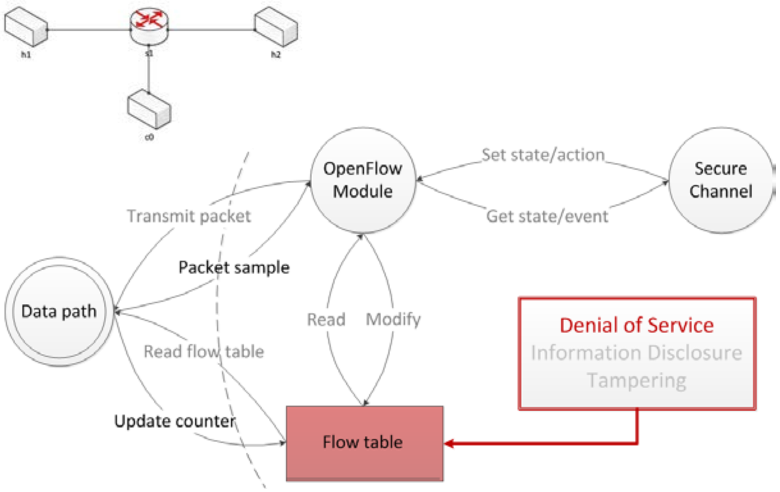
Vulnerability highlights: potential for DoS attacks on the controller and the secure channel, information disclosure via the secure channel, and tampering flow table and controller state, …

Experimental Evaluation

- Mininet-based virtual network using Open vSwitch and a POX-based controller

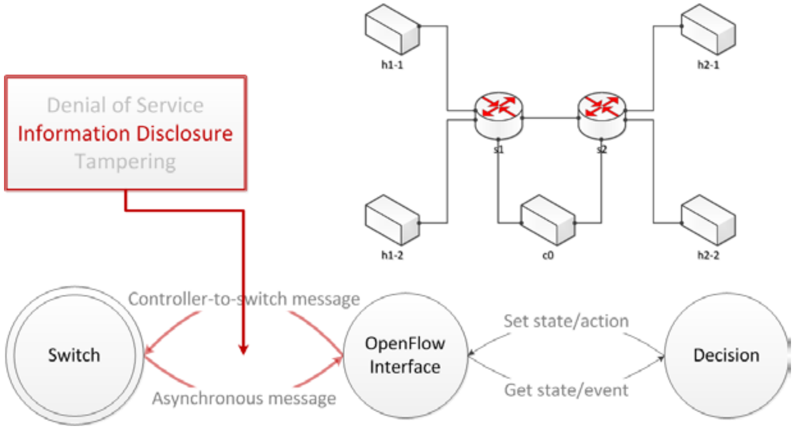
Denial of service on the flow table

- Aim is to overflow the flow table and cause a DoS
- Fixed number of UDP packets sent with permuted source and destination port numbers
- Recorded the number of lost packets corresponding to *All tables full* error with different soft timeout values
- The *forwarding.l2_learning* controller used



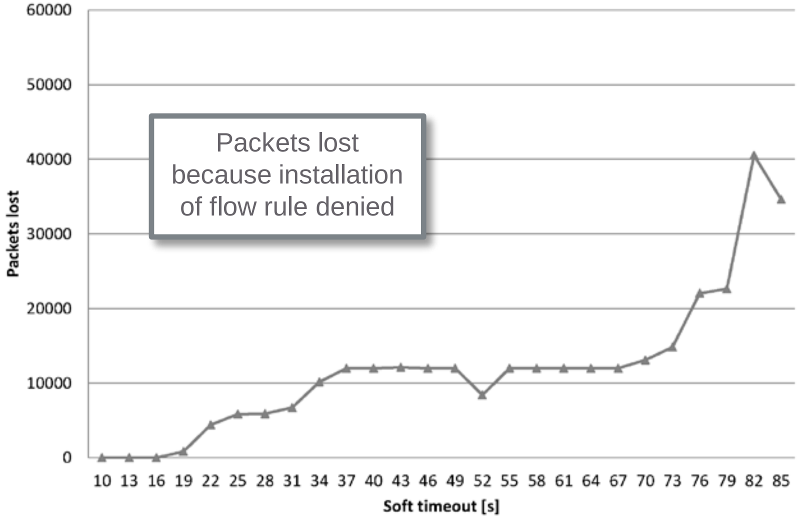
Information disclosure on the secure channel

- Determine the existence of aggregated flow rules…and services
- Measure distribution of TCP connection setup response times
- Flows that have no rules installed will incur extra controller propagation and processing delay
- Two POX modules used:
  - *forwarding.l2_learning* controller (control)
  - *forwarding.l2_aggregator_simple* controller, which uses wildcards for source fields



Experimental Results

Denial of service on the flow table

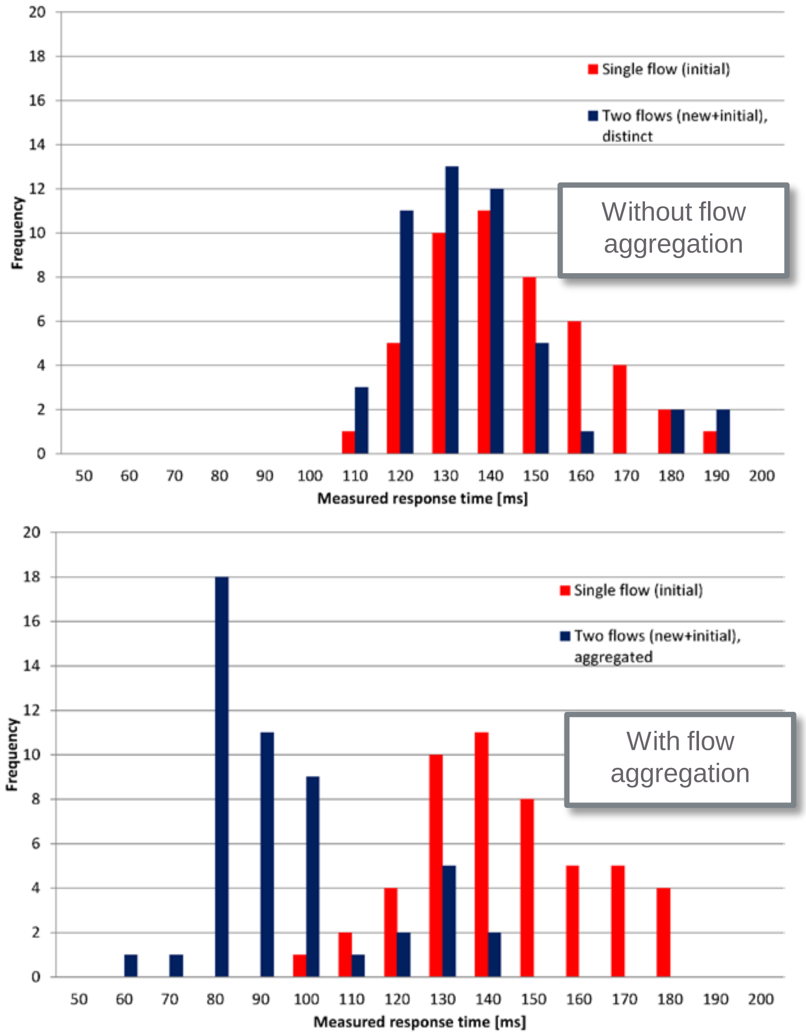


Packets lost because installation of flow rule denied

Take home message

- We can gain insights into controller behaviour based on delay characteristics
- How effective this is depends on the ratio of data path versus control delay
- It is possible create a DoS attack against the Flow table of an OpenFlow switch
- Can these two attacks be combined?

Information disclosure on the secure channel



Without flow aggregation

With flow aggregation

Conclusion and Recommendations

- Identified vulnerabilities in OpenFlow 1.0 and demonstrated they can feasibly be exploited
  - Some vulnerabilities are addressed in later versions of OpenFlow

- The security analysis method can be used to identify vulnerabilities and how they could be exploited
  - This is useful to understand where to focus efforts on security → potentially influence design decisions
  - A challenge is creating DFDs at the right level of abstraction

- Future work could include demonstrating attacks on more realistic infrastructures, e.g., available to the Ofelia project

- More details can be found here:

Rowan Klöti. OpenFlow: A Security Analysis. MSc thesis, D-ITET, ETH Zurich. ftp://ftp.tik.ee.ethz.ch/pub/students/2012-HS/MA-2012-20.pdf, 2013.

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

AIT Austrian Institute of Technology

Dr Paul Smith
Senior Scientist
Safety & Security Department

paul.smith@ait.ac.at | +43 664 883 90031 | www.ait.ac.at/it-security