Towards Secure and Dependable Software-Defined Networks

Diego Kreutz, Fernando M. V. Ramos, Paulo Veríssimo
LaSIGE/FCUL, University of Lisbon
1. **Decoupling** control and data plane

2. **Logical centralization** of network control

3. **Programming** the network
Excellent, now we can program the network!
Wait, now **others** can program the network!
Outline

Main threat vectors in SDNs

Security & Dependability by design

Final remarks
Outline

Main threat vectors in SDNs

Security & Dependability by design

Final remarks
Not specific to SDNs, but can be a door for augmented DoS attacks.

Possible solutions: IDS + rate bounds for control plane requests
Not specific to SDNs, but now the impact is potentially augmented.

Possible solutions: software attestation with autonomic trust management
Specific to SDNs: communication with logically centralized controllers can be explored.

Possible solutions: threshold cryptography across controller replicas
Specific to SDNs, controlling the controller may compromise the entire network.

Possible solutions: replication + diversity + recovery
Specific to SDNs, malicious applications can now be easily developed and deployed on controllers.

Possible solutions: software attestation, security domains
Threat vector 6 exploiting vulnerabilities in admin stations

Not specific to SDNs, but now the impact is potentially augmented.

Possible solutions: double credential verification
Threat vector 7
lack of trusted resources for forensics and remediation

Not specific to SDNs, but it is still critical to assure fast recovery and diagnosis when faults happen.

Possible solutions: immutable and secure logging, secure and reliable snapshots
Seven main threat vectors

- 1 and 3: communications
- 2, 4, 5, 6: elements
- 7: communications and elements
Outline

Main threat vectors in SDNs

Security & Dependability by design

Final remarks
Sec&Dep tools to consider

• Replication
  – Dynamic device association
  – Self-healing mechanisms for *perpetual* operation

• Diversity
  – Avoid shared, common vulnerabilities

• (Autonomic) trust
  – between controllers and devices
  – between applications and controller software
Sec&Dep tools to consider

• Security domains
  – kernel mode vs user mode (e.g., FortNOX)

• Fast and reliable software update and patching (e.g., HotSwap)
DESIGN OF A SEC&DEP SDN CONTROL PLATFORM

v0.1
One single centralized controller
Multiple instances of a centralized controller
Master-slave controllers

Controller A

Controller B

Controller C

App A

App A

App A

nox

nox

nox
Master-slave controllers (what if B fails?)
Master-slave controllers (adding a consistency layer)
Multiple active controllers

Controller A

Controller B

Controller C

East/Westbound API (data distribution service)
One single app instance (App B) can now configure the whole network.
One **single app** instance (App B) can now configure the whole network.
Increasing the robustness of the system by adding diversity.
Diversity of controllers requires a **common Northbound API**.
Outline

Main threat vectors in SDNs

Security & Dependability by design

Final remarks
Our main message

• SDN: a fascinating dilemma
  – evolution of networking
  – increasing the threat surface

Security & Dependability should be built by design.