Poisoning Network Visibility in Software-Defined Networks : New Attacks and Countermeasures

Sungmin Hong, **Lei Xu**, Haopei Wang and Guofei Gu

SUCCESS Lab

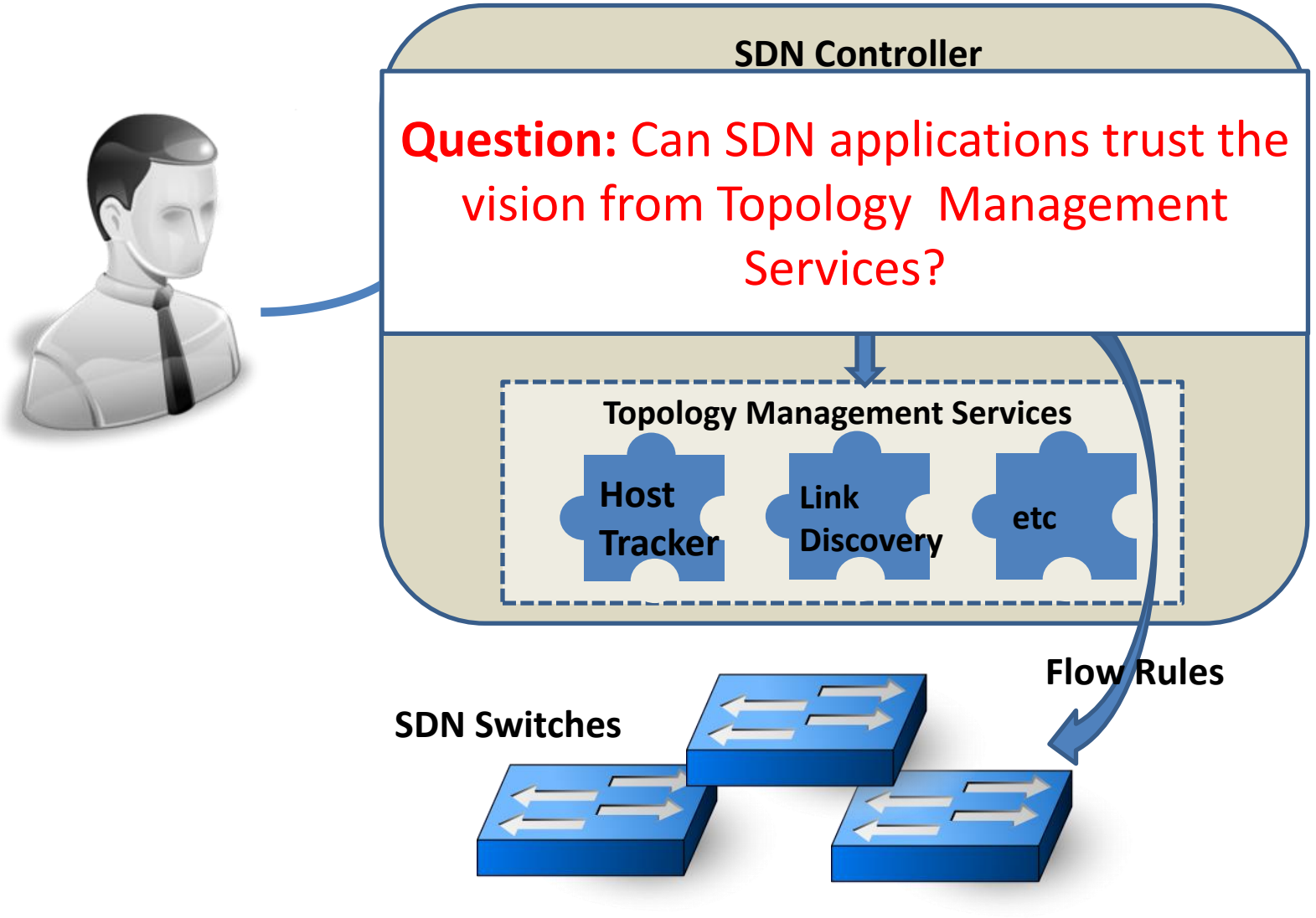
Texas A&M University

What's Software-Defined Network?

- Separate network functionality
  - Control Plane (SDN Controller)
  - Data Plane (SDN Switch)

- SDN Controller runs as "Network OS"
  - Network Visibility
  - Programmability

What's Software-Defined Network? (Cont.)



However…

- The Topology Management Services inside SDN controllers are vulnerable to Topology Poisoning Attacks
  - Host Location Hijacking Attack
  - Link Fabrication Attack

Our Contributions

- Perform security analysis on SDN Topology Management Services
- Propose Topology Poisoning Attacks
- Design and implement a new defense solution: TopoGuard

Topology Poisoning Attack

- Threat Model
  - Attacker controls a collection of compromised hosts or VMs (e.g., by malware Infection) in the SDN network
- Target
  - Topology View of SDN controller
    - Vector1: Host Location Hijacking
    - Vector2: Link Fabrication

Vector 1: Host Location Hijacking Attack

Basics of Host Tracking Service

- Host Tracking Service is used to dynamically track location of hosts in the SDN network
  - Seamless handoff among APs
  - Handle frequent host migrations in data center

- HowTo: maintain Host Profile

Host Profile

Controller	Host Profile
NOX	MAC, Location
POX	MAC, IP, Location
Ryu	MAC, IP, Location
Floodlight	MAC, VLAN ID, IP, Location
OpenDayLight	MAC, VLAN ID, IP, Location
Beacon	MAC, VLAN ID, IP, Location
Maestro	MAC, VLAN ID, IP, Location
OpenIRIS	MAC, Location

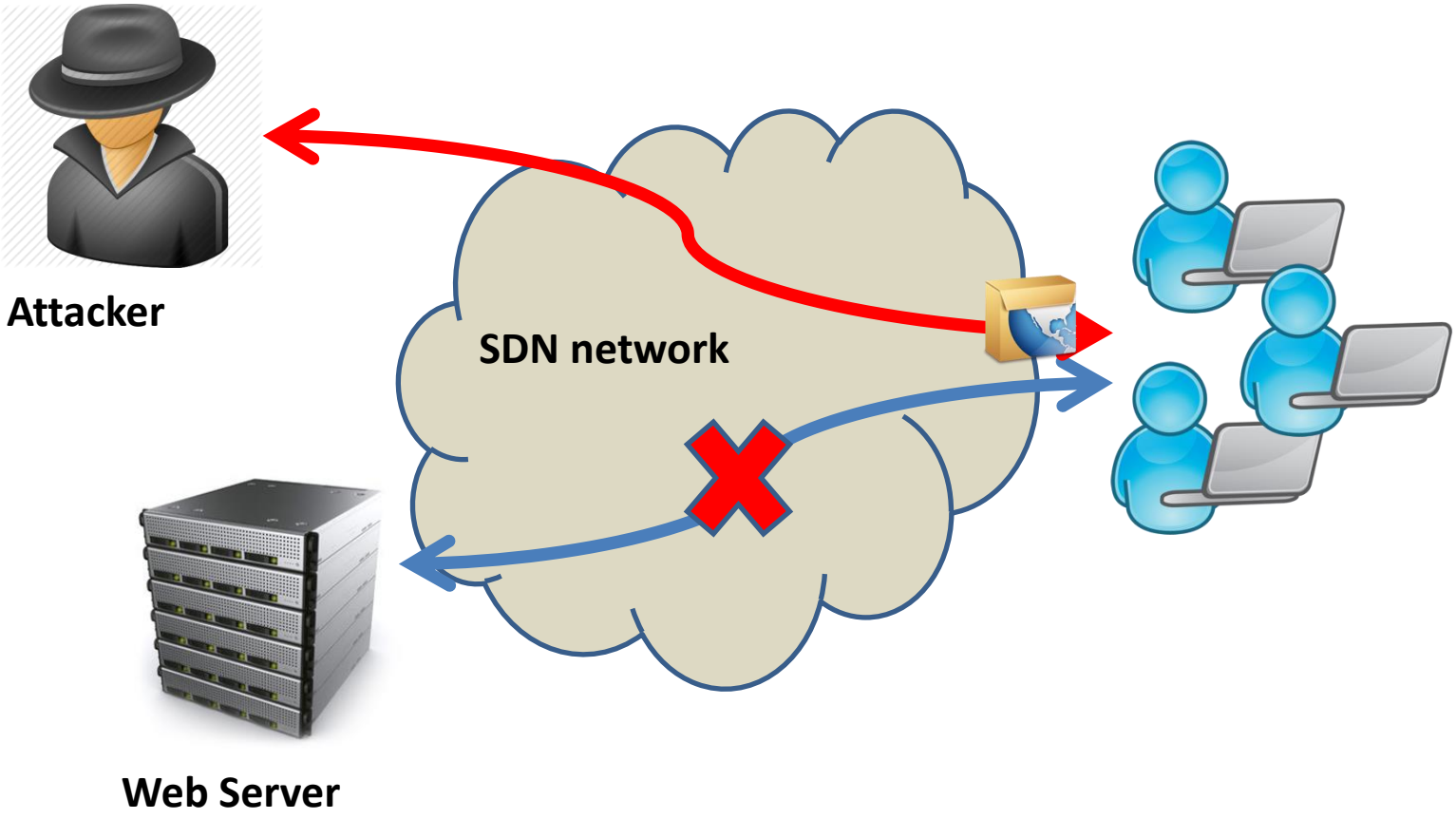
Vector 1: Host Location Hijacking Attack (Cont.)

Vulnerability Analysis

- Few security restrictions on host location update!

- Attacker can impersonate any network identity with its index of Host Profile, e.g., MAC address

Vector 1: Host Location Hijacking Attack (Cont.)



Countermeasure: Host Location Hijacking Attack

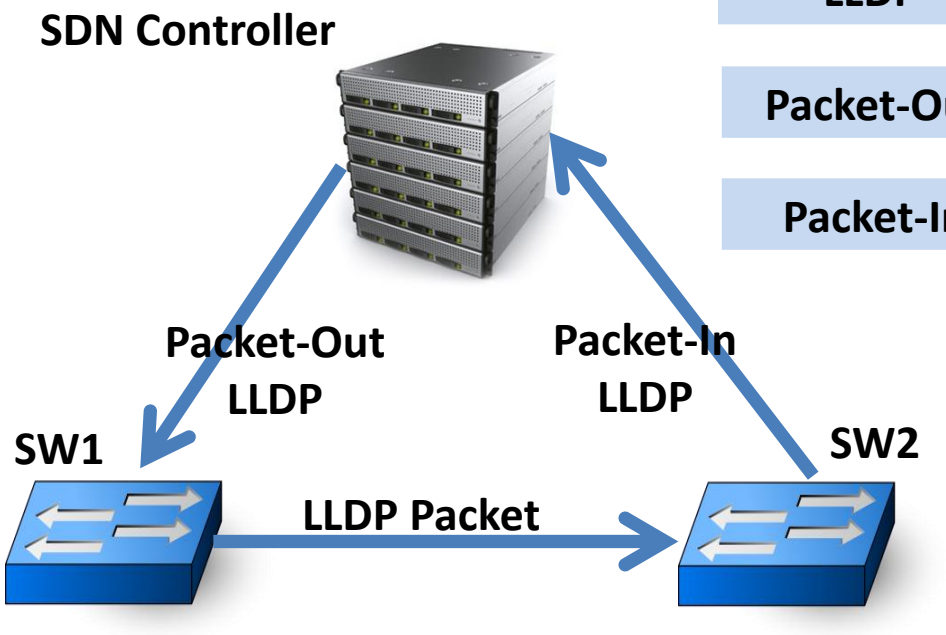
Verify the legitimacy of Host Migration

- Pre-Condition Check
  - Invariant: Port-Down Signal

- Post-Condition Check
  - Invariant: Non-Reachability in previous location

Vector2: Link Fabrication Attack

- Basics of Link Discovery Service
  - SDN controller discovers switch connections by LLDP packets

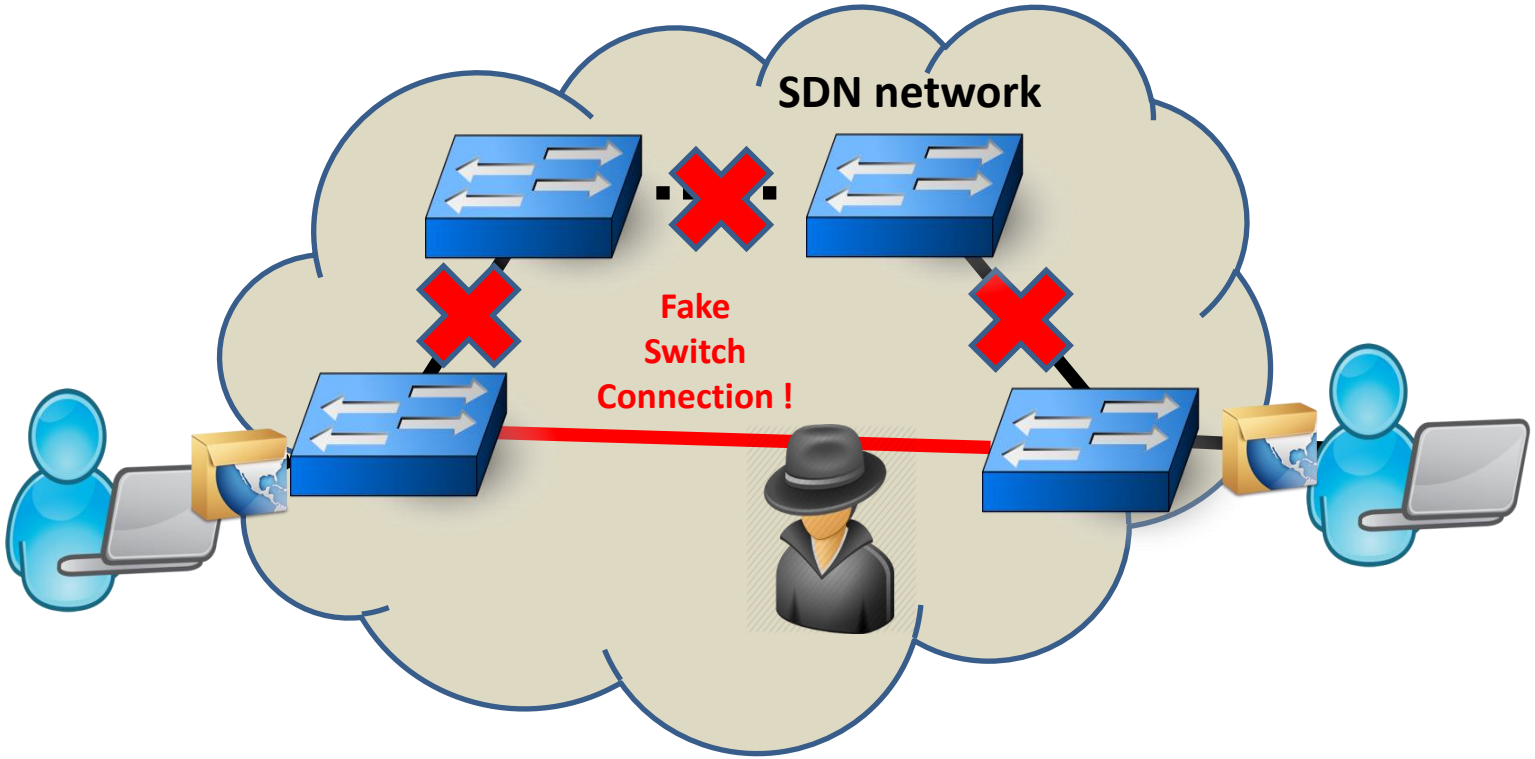


LLDP	SW1_DPID	SW1_PORT
Packet-Out	SW1_DPID	Output: SW1_PORT
Packet-In	SW2_DPID	SW2_PORT

Vector2: Link Fabrication Attack (Cont.)

- Vulnerability Analysis
  - **Security Omission1** : The integrity of LLDP packets can be violated
  - **Security Omission2** : A host can be involved in LLDP propagation
- Fake LLDP Injection
- LLDP Relay

Vector2: Link Fabrication Attack (Cont.)



Countermeasure: Link Fabrication Attack

Verification

- LLDP propagation path invariant
  - Solution: switch port role check

- LLDP integrity Invariant
  - Solution: HMAC

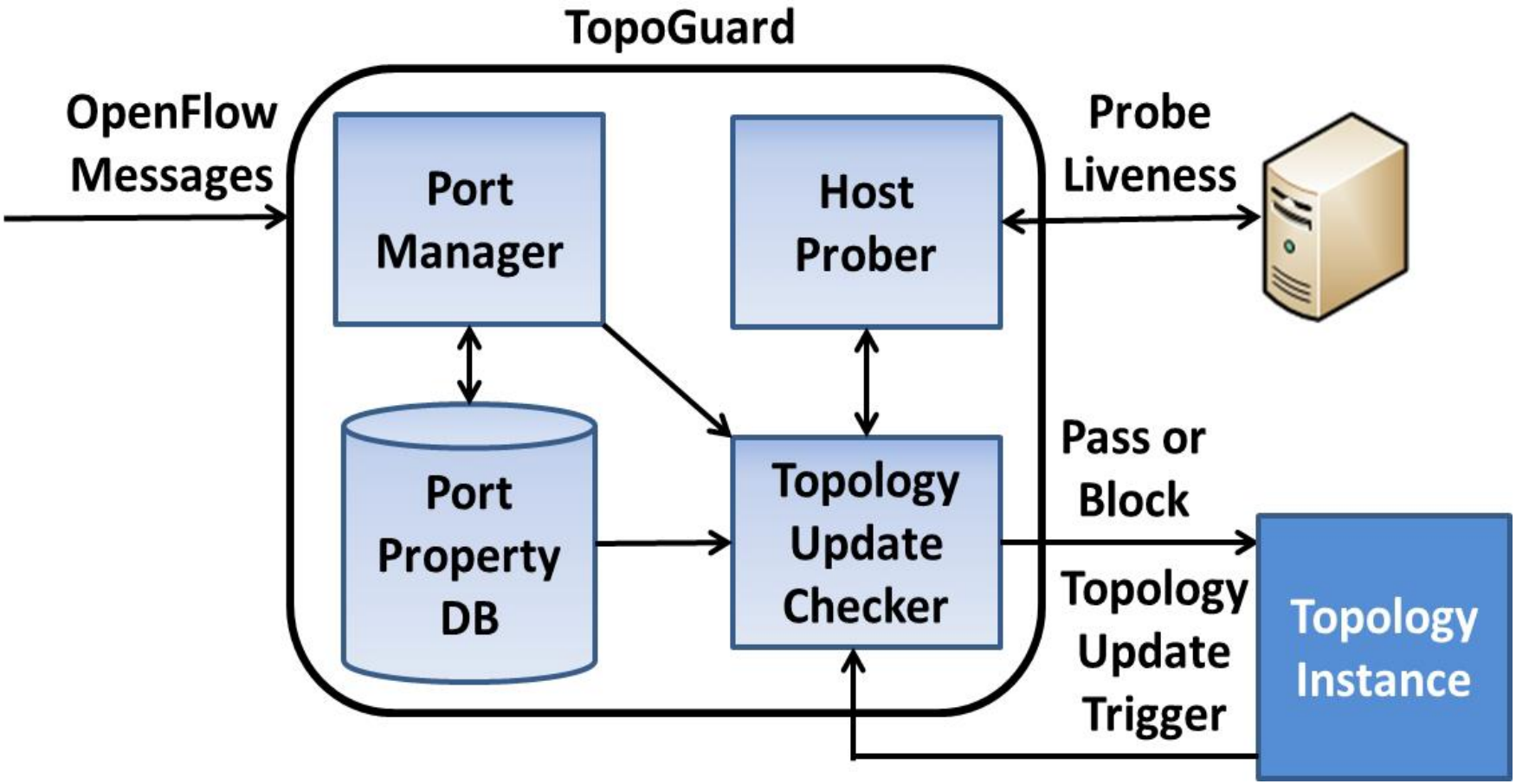
Vulnerable SDN Controllers in the market

Controller	Host Tracking Service	Link Discovery Service
NOX	hosttracker.cc	discovery.py
POX	host_tracker.py	discovery.py
Ryu	host_tracker.py	switches.py
Floodlight	DeviceManagerImpl.java	LinkDiscoveryManager.java
OpenDayLight	DeviceManagerImpl.java	DiscoveryService.java
Beacon	DeviceManagerImpl.java	TopologyImpl.java
Maestro	LocationManagementApp.java	DiscoveryApp.java
OpenIRIS	OFMDeviceManager.java	OFMLinkDiscovery.java

Defense System

- We propose , **TopoGuard**, currently as a new security extension in Floodlight controller
  - Pre-Condition check and Post-Condition check
  - Switch port role check
  - HMAC
- The source code is online:
  - https://github.com/xuraylei/floodlight_with_topoguard.git
- In the future, we will realize our mitigations to other controllers

System Architecture



Evaluations: Effectiveness

Pre-condition and Post-Condition violations

```
lew I/O server worker #2-1] Link added: Link [src=00:00:00:00:00:00:00:01 outPort=3, dst=00:00:00:00:00:00:
:rver-main] Starting DebugServer on :6655
'$PortManager:New I/O server worker #2-1] Device:7a:f1:0d:d6:31:fd is added on:
'$PortManager:New I/O server worker #2-1] sw:1,port:1
'$PortManager:New I/O server worker #2-1] Device:96:2a:7e:28:2f:54 is added on:
'$PortManager:New I/O server worker #2-1] sw:1,port:2
'$PortManager:New I/O server worker #2-2] Device:ca:81:f9:df:0f:b1 is added on:
'$PortManager:New I/O server worker #2-2] sw:2,port:2
'$PortManager:New I/O server worker #2-1] Device:2a:45:f6:50:b9:cf is added on:
'$PortManager:New I/O server worker #2-1] sw:3,port:3
'$PortManager:New I/O server worker #2-2] Violation: Host Move from switch 1 port 1 without Port ShutDown
'$PortManager:New I/O server worker #2-1] Violation: Host Move from switch 1 port 1 is still reachable
```

Switch port role violation

```
lew I/O server worker #2-2] Link added: Link [src=00:00:00:00:00:00:00:03 outPort=1, dst=00:00:00:00:00:00:
ler:New I/O server worker #2-1] Inter-switch link detected: Link [src=00:00:00:00:00:00:00:02 outPort=3, ds
lew I/O server worker #2-1] Link added: Link [src=00:00:00:00:00:00:00:02 outPort=3, dst=00:00:00:00:00:00:
ler:New I/O server worker #2-1] Inter-switch link detected: Link [src=00:00:00:00:00:00:00:01 outPort=3, ds
lew I/O server worker #2-1] Link added: Link [src=00:00:00:00:00:00:00:01 outPort=3, dst=00:00:00:00:00:00:
lew I/O server worker #2-1] Link added: Link [src=00:00:00:00:00:00:00:03 outPort=2, dst=00:00:00:00:00:00:
ler:New I/O server worker #2-1] Inter-switch link updated: Link [src=00:00:00:00:00:00:00:03 outPort=2, dst
lew I/O server worker #2-1] Link updated: Link [src=00:00:00:00:00:00:00:03 outPort=2, dst=00:00:00:00:00:0
:rver-main] Starting DebugServer on :6655
'$PortManager:New I/O server worker #2-2] Violation: Receive LLDP packets from HOST port: SW 1 port 2
'$PortManager:New I/O server worker #2-2] Violation: Receive LLDP packets from HOST port: SW 1 port 2
'$PortManager:New I/O server worker #2-2] Violation: Receive LLDP packets from HOST port: SW 1 port 2
```

Evaluations: Overhead

- TopoGuard introduces two-fold overhead
  - Delay for processing LLDP and other Packet-Ins
  - Additional time overhead to verify HMAC TLV

LLDP Processing Overhead	Normal Packet processing Overhead
0.02ms	0.032ms

Conclusion

- The topology management services in SDN controller is facing security challenges
- Two Topology Poisoning Attacks can poison the topology view of SDN controller
- New security extensions to SDN controller as mitigations to the threats

Thanks You !

Backup: HMAC Overhead