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.)

**Question:** Can SDN applications trust the vision from Topology Management Services?
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

<table>
<thead>
<tr>
<th>Controller</th>
<th>Host Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOX</td>
<td>MAC, Location</td>
</tr>
<tr>
<td>POX</td>
<td>MAC, IP, Location</td>
</tr>
<tr>
<td>Ryu</td>
<td>MAC, IP, Location</td>
</tr>
<tr>
<td>Floodlight</td>
<td>MAC, VLAN ID, IP, Location</td>
</tr>
<tr>
<td>OpenDayLight</td>
<td>MAC, VLAN ID, IP, Location</td>
</tr>
<tr>
<td>Beacon</td>
<td>MAC, VLAN ID, IP, Location</td>
</tr>
<tr>
<td>Maestro</td>
<td>MAC, VLAN ID, IP, Location</td>
</tr>
<tr>
<td>OpenIRIS</td>
<td>MAC, Location</td>
</tr>
</tbody>
</table>
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.)

SDN network

Attacker

Web Server
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

<table>
<thead>
<tr>
<th></th>
<th>SW1_DPID</th>
<th>SW1_PORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packet-Out</td>
<td>SW1_DPID</td>
<td>Output: SW1_PORT</td>
</tr>
<tr>
<td>Packet-In</td>
<td>SW2_DPID</td>
<td>SW2_PORT</td>
</tr>
</tbody>
</table>
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.)

Fake Switch Connection!
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

<table>
<thead>
<tr>
<th>Controller</th>
<th>Host Tracking Service</th>
<th>Link Discovery Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOX</td>
<td>hosttracker.cc</td>
<td>discovery.py</td>
</tr>
<tr>
<td>POX</td>
<td>host_tracker.py</td>
<td>discovery.py</td>
</tr>
<tr>
<td>Ryu</td>
<td>host_tracker.py</td>
<td>switches.py</td>
</tr>
<tr>
<td>Floodlight</td>
<td>DeviceManagerImpl.java</td>
<td>LinkDiscoveryManager.java</td>
</tr>
<tr>
<td>OpenDayLight</td>
<td>DeviceManagerImpl.java</td>
<td>DiscoveryService.java</td>
</tr>
<tr>
<td>Beacon</td>
<td>DeviceManagerImpl.java</td>
<td>TopologyImpl.java</td>
</tr>
<tr>
<td>Maestro</td>
<td>LocationManagementApp.java</td>
<td>DiscoveryApp.java</td>
</tr>
<tr>
<td>OpenIRIS</td>
<td>OFMDeviceManager.java</td>
<td>OFMLinkDiscovery.java</td>
</tr>
</tbody>
</table>
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
Evaluations: Effectiveness

Pre-condition and Post-Condition violations

Switch port role violation
Evaluations: Overhead

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

<table>
<thead>
<tr>
<th>LLDP Processing Overhead</th>
<th>Normal Packet processing Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02ms</td>
<td>0.032ms</td>
</tr>
</tbody>
</table>
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

LLDP Construction time (ms)

- FL without HMAC: 0.25
- FL with HMAC: 0.28

LLDP Verification time (ms)

- FL without HMAC: 0.30
- FL with HMAC: 0.32

Overhead:
- Without HMAC: 2.92%
- With HMAC: 1.64%