Automated Discovery of Cross-Plane Event-Based Vulnerabilities in Software-Defined Networking

Benjamin E. Ujcich¹, Samuel Jero², Richard Skowyra², Steven R. Gomez², Adam Bates¹, William H. Sanders¹, and Hamed Okhravi²

¹ University of Illinois at Urbana-Champaign, ² MIT Lincoln Laboratory
Network “Appification”

Introducing HPE SDN App Store
Select from a range of SDN Applications that allow you to program your network to align with business needs. Deploy directly to the enterprise-ready HPE VAN SDN Controller.

Get Started

1. **Submit your apps**
   Find the resources you need to build a SDN application with documentation and discussions from an active developer community.

2. **Reach your audience**
   After the review process, your SDN application will be available to thousands of users on the industry’s first SDN marketplace.

Security
Apps that extend, or enforce security across the network. If you’d like to talk to us about these solutions, add the products you’re interested in to your shopping cart then look for the Contact HP button to submit to a HP sales professional. HP will then reach out to you outside of the store; or put you in contact with an appropriate HP channel partner, or the ISV.

- All products
- All versions
- All companies
- Clear filters

Sort by: Newest, Downloads, A-Z

- HPE COMMUNITY FREE
- HPE Network Protector: Free Trial
- Hewlett Packard Enterprise

- COMMUNITY FREE
- FortiGate Connector for HP VAN Controller

- COMMUNITY FREE
- Active Honeypot Guardian
Network “Appification”

Do apps work well together?

Introducing HPE SDN App Store
Select from a range of SDN Applications that allow you to program your network to align with business needs. Deploy directly to the enterprise ready HPE VAN SDN Controller.

Get Started

1. Submit your apps
   Find the resources you need to build a SDN applications with documentation and discussions from an active developer community.

2. Reach your audience
   After the review process, your SDN application will be available to thousands of users on the industry’s first SDN marketplace.

Security
Apps that extend, or enforce security across the network. If you’d like to talk to us about these solutions, add the products you’re interested in to your shopping cart then look for the Contact HP button to submit to a HP sales professional. HP will then reach out to you outside of the store; or put you in contact with an appropriate HP channel partner, or the ISV.
Network “Appification”

How can they be exploited?

Get Started

1. Submit your apps
   Find the resources you need to build a SDN applications with documentation and discussions from an active developer community.

2. Reach your audience
   After the review process, your SDN application will be available to thousands of users on the industry’s first SDN marketplace.
Cross-Plane Vulnerabilities

Reactive Event-Based Control Plane

- Packets from hosts
- Flow rule configuration

CONTROL PLANE
DATA PLANE

SDN data plane input

SDN data plane output

OPEN DAYLIGHT

Project Floodlight

ONOS
Open Network Operating System
Cross-Plane Vulnerabilities

SDN data plane input (e.g., packets) → Event dispatch → App event listener → App event listener → SDN data plane output (e.g., flow rules)

Reactive Event-Based Control Plane
Cross-Plane Vulnerabilities

SDN data plane input (e.g., packets) → App event listener → Event dispatch → App event listener → Event dispatch → App event listener → SDN data plane output (e.g., flow rules)

Reactive Event-Based Control Plane
Cross-Plane Vulnerabilities

SDN data plane input (e.g., packets)

Event dispatch

App event listener

Event dispatch

App event listener

App event listener

Reactive Event-Based Control Plane

SDN data plane output (e.g., flow rules)

API call
Cross-Plane Vulnerabilities: **Exploitation**

**Vector:** Spoofed packets via malware

SDN data plane input (e.g., packets)

SDN data plane output (e.g., flow rules)

**Insight:** Missing event handling can be exploited

**Target:** New flow rules; removal of old flow rules

**Malicious information flow**
Data Plane Hosts as Attack Vectors

- Cross-app study led to explore hosts as attackers
- Discovered ONOS data plane firewall vulnerability → arbitrary lateral movement
- Reported to ONOS developers (CVE 2018-12691)
Anatomy of an Exploit

1. The access control app (acl) is activated and registers for any host events (A).

![Diagram of network components and interactions](image-url)
Anatomy of an Exploit

1. The access control app (acl) is activated and registers for any host events (A). The network operator adds access control policies (B).
Anatomy of an Exploit

2. Host 1 sends a syntactically correct but semantically invalid ICMP packet with host 1’s MAC address into the data plane.
Anatomy of an Exploit

3. ONOS sees the packet (A)
Anatomy of an Exploit

3. ONOS sees the packet (A) and registers a new host with its MAC address but not IP address (B).
3. ONOS sees the packet (A) and registers a new host with its MAC address but not IP address (B). It generates a HOST_ADDED event (C).
Anatomy of an Exploit

4. The acl app sees the HOST_ADDED event (A)
Anatomy of an Exploit

4. The acl app sees the **HOST_ADDED** event (A) and host (B), but since the host doesn’t have an IP, the app does not insert flow deny rules.
Anatomy of an Exploit

4. The acl app sees the HOST_ADDED event (A) and host (B), but since the host doesn’t have an IP, the app does not insert flow deny rules.
Anatomy of an Exploit

5. Host 1 attempts to send regular traffic to its desired victim destination (host 2). Since no matching flows exist, ONOS handles the packet.
Anatomy of an Exploit

6. ONOS registers host 1’s new IP address (A)
6. ONOS registers host 1’s new IP address (A) as a HOST_UPDATED event (B). acl does not handle HOST_UPDATED events, so it does nothing.
Anatomy of an Exploit

6. ONOS registers host 1’s new IP address (A) as a HOST_UPDATED event (B). acl does not handle HOST_UPDATED events, so it does nothing.

AclManager.java
handling new host events
HOST_UPDATED events not handled

AclManager.java processing host events
× Never gets called

private void processHostAddedEvent(HostEvent event, AclRule rule) {
  DeviceId deviceId = event.subject().location().deviceId();
  for (IpAddress address : event.subject().ipAddresses()) {
    if ((rule.srcIp() != null) ?
      (checkIpInCidr(address.getIp4Address(), rule.srcIp())
      (checkIpInCidr(address.getIp4Address(), rule.dstIp())
        if (!aclStore.checkRuleWorksInDevice(rule.id(), deviceId)
        if (aclStore.getAllowingRuleByDenyingRule(rule.id())){
          List<RuleId> allowingRuleList = aclStore
            .getAllowingRuleByDenyingRule(rule.id());
            if (allowingRuleList != null) {
              for (RuleId allowingRuleId : allowingRuleList) {
                generateAclFlow(aclStore.getAclRule(allowing
              }
            }
            }
        }
    }
}

Never gets called
Anatomy of an Exploit

7. The packet gets sent to a second app (A)
Anatomy of an Exploit

7. The packet gets sent to a second app (A), which instantiates the flow (allow) rule (B)
Anatomy of an Exploit

7. The packet gets sent to a second app (A), which instantiates the flow (allow) rule (B) and allows host 1 to communicate with host 2 (C).
What Makes This Challenging?

- No ground truth about what events ought to be handled
- Multiple entry points for code analysis
- Not all event handling can affect the data plane
What Makes This Challenging?

- No ground truth about what events ought to be handled
- Multiple entry points for code analysis
- Not all event handling can affect the data plane

EVENTSCOPE
Automated Discovery of Cross-Plane Event-Based Vulnerabilities in SDN
What Makes This Challenging?

- No ground truth about what events ought to be handled
- Multiple entry points for code analysis
- Not all event handling can affect the data plane

EVENTSCOPE
Automated Discovery of Cross-Plane Event-Based Vulnerabilities in SDN

Found 14 new vulnerabilities
EventScope Solution

No ground truth about what events ought to be handled

Cluster apps according to similar functionality

Multiple entry points for code analysis

Not all event handling can affect the data plane
**EVENTSCOPE**

**App Event Use**

- **SDN app code**
- **Candidate Vulnerability Generator**
  - Missing Event Types
    - 1
    - 2
    - 3

- **SDN controller code**

- **API definition**

---

**Host event kind**
- HOST_ADDED
- HOST_REMOVED
- HOST_UPDATED
- HOST_MOVED

**Event types of Host event kind**

**Link event kind**
- LINK_ADDED
- LINK_REMOVED
- LINK_UPDATED

**Event types of Link event kind**

---

**Category 1**
- L2 forwarding app
- L3 routing app

**Category 2**
- Firewall app

**Firewall**

---

32
EventScope App Event Use

Candidate Vulnerability Generator

SDN app code

SDN controller code

API definition

Missing Event Types

Apps

0.0 0.2 0.4 0.6 0.8 1.0

Similarity
EVENTSCOPE Solution

- No ground truth about what events ought to be handled
- Multiple entry points for code analysis
- Not all event handling can affect the data plane

Abstract event flow with graphical model
EVENTSCOPE
Event Flow Graph

SDN app code

SDN controller code

API definition

Candidate Vulnerability Generator
Missing Event Types
1 2 3

Event Flow Graph Generator
Data Plane In
Data Plane Out

Component 1
Packet event listener

App 1
Host event listener

App 2
Host event listener

Component 2
Host event listener

Use event listeners of components and apps as entry points
**EVENTSCOPE**

Event Flow Graph

- **SDN app code**
- **SDN controller code**
- **API definition**

---

**Candidate Vulnerability Generator**

- Missing Event Types: 1 2 3

**Event Flow Graph Generator**

- Data Plane In
- Data Plane Out

---

**Component 1 Packet event listener**

**Dispatching Host event to all Host listeners**

- App 1 Host event listener
- App 2 Host event listener
- Component 2 Host event listener

---

**HOST_ADDED**

**HOST_REMOVED**

**HOST_UPDATED**

etc...

**Link event dispatchers and event listeners**
**EVENTSCOPE**

Event Flow Graph

- SDN app code
- SDN controller code
- API definition

### Candidate Vulnerability Generator

Missing Event Types

1 2 3

### Event Flow Graph Generator

- Data Plane In
- Data Plane Out

### Event Flow Graph

- **Host service**
  - `getHosts()` READ
  - `addFlow()` WRITE

- **FlowRule service**

### Component 1 Packet event listener

- `HOST_ADDED`
- `HOST_REMOVED`
- `HOST_UPDATED`
  etc...

### Component 2 Host event listener

### App 2 Host event listener

### App 1 Host event listener

Dispatching Host event to all Host listeners

Add **API calls** to relevant control plane objects
**EVENTSCOPE**

Event Flow Graph

- **SDN app code**
- **SDN controller code**
- **API definition**

---

Candidate Vulnerability Generator

- Missing Event Types
- 1 2 3

Event Flow Graph Generator

- Data Plane In
- Data Plane Out

---

App 1 **Host** event listener

- `getHosts()` READ

App 2 **Host** event listener

- `addFlow()` WRITE

Component 2 **Host** event listener

---

**DATA PLANE INPUTS** (e.g., packets from hosts)

Component 1 **Packet** event listener

- `HOST_ADDED`
- `HOST_REMOVED`
- `HOST_UPDATED`
- etc...

---

**DATA PLANE OUTPUTS** (e.g., switch flow rules)

**Identify API calls where data plane inputs and outputs occur**
EVENTSCOPE Solution

- No ground truth about what events ought to be handled
- Multiple entry points for code analysis
- Not all event handling can affect the data plane
- Trace viable control paths in event flow graph
EVENTSCOPE
Vulnerability Validation

Candidate Vulnerability Generator
- SDN app code
- SDN controller code
- API definition

Event Flow Graph Generator
- Data Plane In
- Data Plane Out

Vulnerability Validator
- Data Plane In
- Data Plane Out

Candidate Vulnerability Generator:
- Missing Event Types
  1. Event 1
  2. Event 2
  3. Event 3

Event Flow Graph Generator:
- Data Plane In
- Data Plane Out

Vulnerability Validator:
- Data Plane In
- Data Plane Out

DATA PLANE INPUTS
- Host Manager Packet event listener
- Access control app Host event listener
- App Host event listener
- Component Host event listener

DATA PLANE OUTPUTS
- Event listener with relevant missing event
- Event listener (other)
EVENTSCOPE Evaluation

Candidate Vulnerability Generator
- Missing Event Types
  - 1
  - 2
  - 3

Event Flow Graph Generator
- Data Plane In
- Data Plane Out

Vulnerability Validator
- Data Plane In
- Data Plane Out

Vulnerabilities

Reported 14 vulnerabilities to ONOS Security Team and requested CVE identifiers

ECE ILLINOIS
CVE-2019-11189

Firewall app installs flow deny rule to prevent host from communicating
CVE-2019-11189

Access control app never gets called!

Malicious host spoofs ARP message, which tricks controller into thinking host has moved

Host mobility app removes access control’s flow rules!
CVE-2019-11189

Malicious host can now force flow allow rule to be installed and send packets into the network.
Conclusions

- Considered the **cross-plane event-based vulnerability** problem in SDN

- Design takeaways
  - **Hosts** have **outsized effect** on SDN operation
  - Security analysis must consider **all apps working together**
  - Developers must **design defensively**

- Discovered and validated **14 new vulnerabilities** in ONOS SDN controller
Thank you for your time!

Benjamin E. Ujcich
E-mail: ujcich2@illinois.edu
Web: http://ujcich2.web.engr.illinois.edu

This work was supported in part by NSF Grant Nos. CNS-1657534 and CNS-1750024. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.