CENTRE FOR SECURE INFORMATION TECHNOLOGIES

Queen's University Belfast

SDN Security

COINS Summer School

Dr. Sandra Scott-Hayward

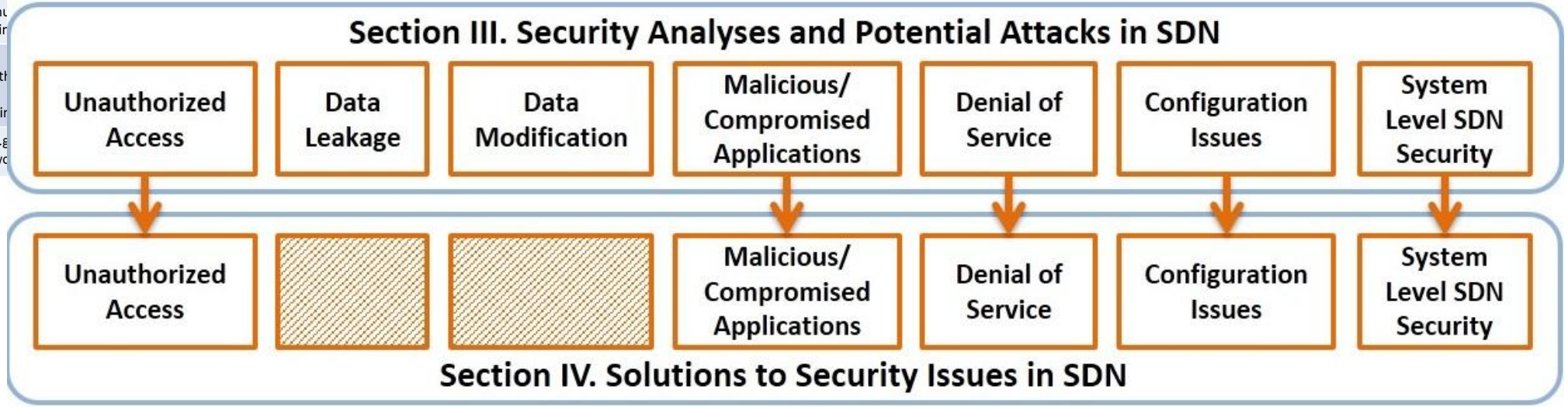
23 August 2015

@CSIT_QUB

Solutions to Security Issues in SDN

Solutions to Security Issues - Analysis

Security Issue/Attack	SDN Layer Affected or Targeted				
	Application Layer	App-Ctl Interface	Control Layer	Ctl-Data Interface	Data Layer
Unauthorized Access e.g.					
• Unauthorized Controller Access/Controller Hijacking			X	X	X
• Unauthorized/Unauthenticated Application	X	X	X		
Data Leakage e.g.					
• Flow Rule Discovery (Side Channel Attack on Input Buffer)					X
• Credential Management (Keys, Certificates for each Logical Network)					X
• Forwarding Policy Discovery (Packet Processing Timing Analysis)			X	X	X
Data Modification e.g.					
• Flow Rule Modification to Modify Packets (Man-in-the-Middle attack)			X	X	X
Malicious/Compromised Applications e.g.					
• Fraudulent Rule Insertion	X	X	X		
Denial of Service e.g.					
• Controller-Switch Commu					
• Switch Flow Table Floodir					
Configuration Issues e.g.					
• Lack of TLS (or other Auth					
• Policy Enforcement					
• Lack of Secure Provisionir					
System Level SDN Security e.g					
• Lack of Visibility of Netwo					



Categorization of Security Solutions

Solution to Security Issue	Research Work	SDN Layer/Interface				
		App	App-Ctl	Ctl	Ctl-Data	Data
Unauthorized Access	Securing Distributed Control [44], Byzantine-Resilient SDN [45]			✓	✓	
	Authentication for Resilience [46]			✓		
	PermOF [47]	✓	✓			
	OperationCheckpoint [48]	✓	✓	✓		
	SE-Floodlight [49], [50]	✓	✓	✓	✓	
	AuthFlow [51]	✓		✓	✓	✓
Data Leakage						
Data Modification						
Malicious Applications	FortNOX [52]	✓	✓	✓	✓	
	ROSEMARY [53]	✓		✓		
	LegoSDN [54]	✓	✓	✓		
Denial of Service	AVANT-GUARD [55], CPRecovery [56]			✓	✓	✓
	VAVE [57]	✓		✓	✓	✓
	Delegating Network Security [58]	✓	✓	✓	✓	✓
Configuration Issues	NICE [59]	✓	✓		✓	
	FlowChecker [60], Flover [61], Anteater [62], VeriFlow [63], NetPlumber [64]	✓	✓	✓	✓	
	Security-Enhanced Firewall [65], FlowGuard [66], [67], LPM [68]	✓		✓	✓	✓
	Frenetic [69], Flow-Based Policy [70], Consistent Updates [71]	✓	✓	✓	✓	
	Shared Data Store [72]	✓		✓	✓	✓
	Splendid Isolation [73]		✓	✓		
	Verificare [74], Machine-Verified SDN [75], VeriCon [76]		✓	✓	✓	
System Level SDN Security	Debugger for SDN [77]	✓			✓	
	OFHIP [78], Secure-SDMN [79]				✓	
	FRESCO [80]	✓	✓	✓	✓	

Policy Chaining (Data Plane Ambiguity)

Firewall – IDS – Proxy:

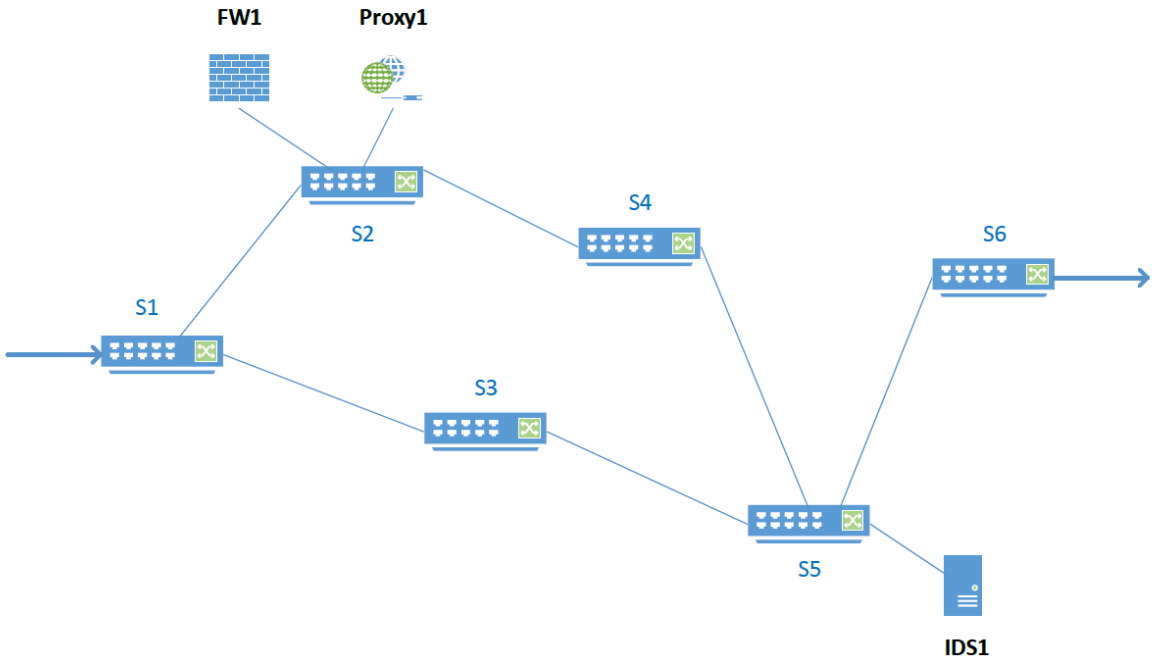
S5 sees the same packet three times and must choose between three actions:

(1) Forward to IDS1
(2) Forward back to S2 for Proxy1
(3) Send to the destination

Proposed Solution:

Tag packet headers to identify the processing state (i.e. location in policy chain) and tunnel packets between switches.

Z.A.Qazi et al.,"SIMPLE-fying Middlebox Policy Enforcement using SDN," ACM SIGCOMM, August 2013.



Policy Chain	Physical Sequence
FW1 – IDS1 – Proxy1	S1 S2 FW1 S2 S4 S5 IDS1 S5 S4 S2 Proxy1 S2 S4 S5 S6

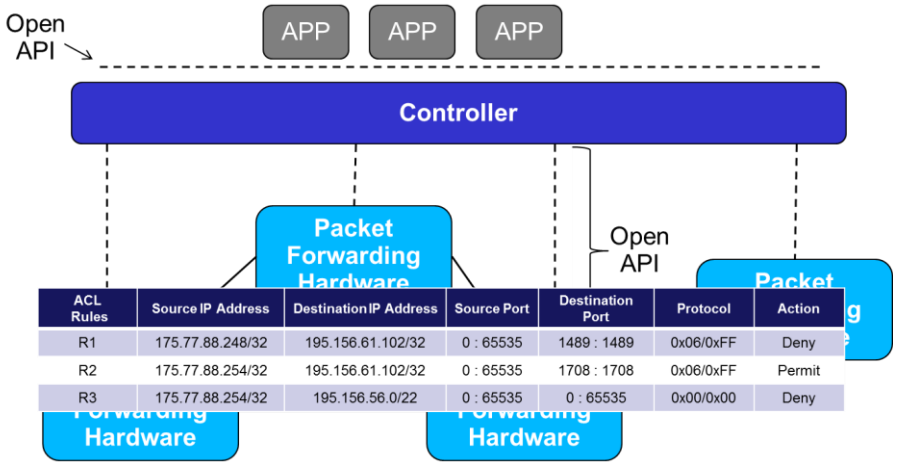
Mitigating SDN Architecture threats using standard technoligies

E.g. SANE Security Analysis (similar OpenFlow Threat Analysis within ONF SecWG)

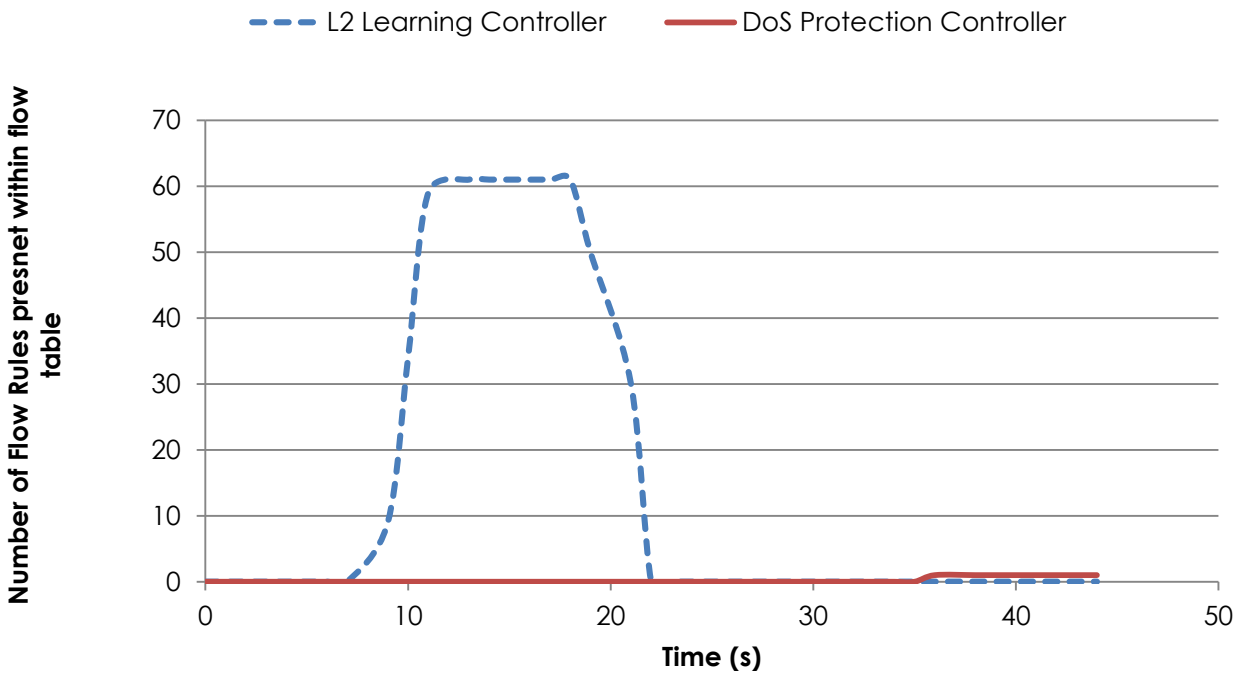
Threat Type	Data Flows	Data Stores	Processes	Interactors
Spoofing				-
Tampering	X[1]	X[2]		
Repudiation			X[4]	X[4]
Information Disclosure	X[1]	X[2,3]		
DoS	-	-	-	
Elevation of Privilege			X[5]	

[1]mitigated with IPSec, [2]mitigated with ACLs, [3]mitigated by not storing secrets, [4]auditing trails in logfile, [5]run with least privileges

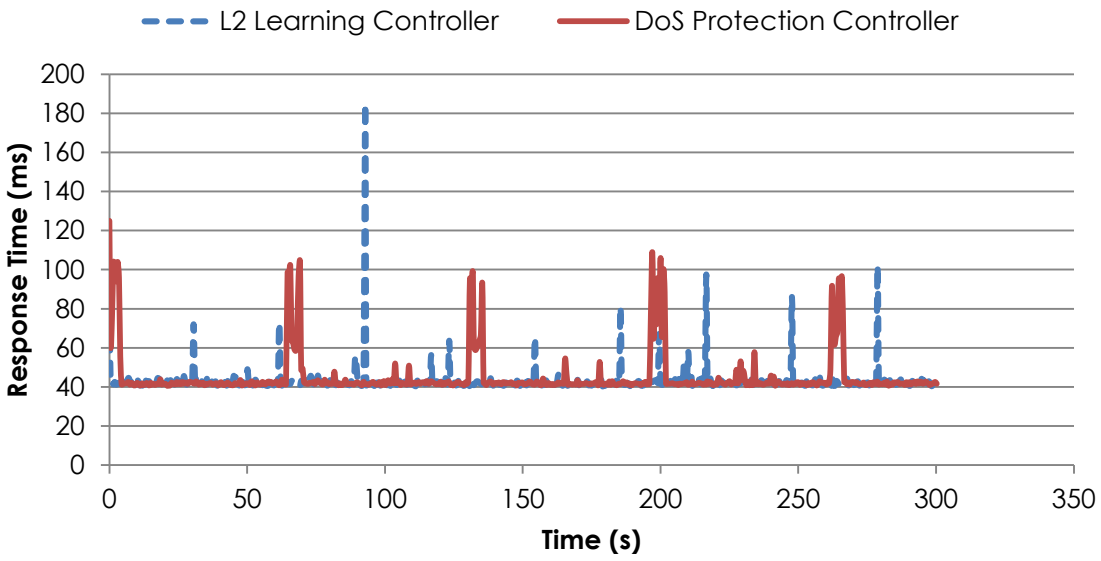
SDN Flow Table Flooding Attack

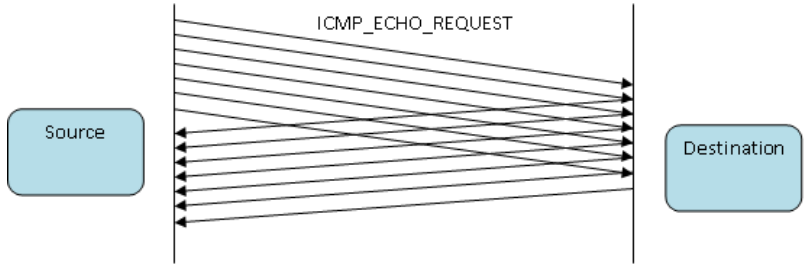


ACL Rules	Source IP Address	Destination IP Address	Source Port	Destination Port	Protocol	Action
R1	175.77.88.248/32	195.156.61.102/32	0 : 65535	1489 : 1489	0x06/0xFF	Deny
R2	175.77.88.254/32	195.156.61.102/32	0 : 65535	1708 : 1708	0x06/0xFF	Permit
R3	175.77.88.254/32	195.156.56.0/22	0 : 65535	0 : 65535	0x00/0x00	Deny



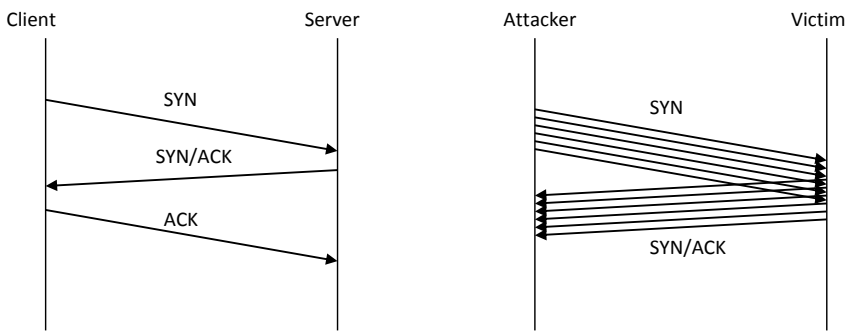
ICMP Flood Attack





Controller	Average Response Time (ms)	Total traffic received on port of victim during ICMP flood (packets)	Total traffic transmitted by all hosts during ICMP flood (packets)
L2 Learning	43.172	99608	99608
DoS Protection	45.282	1252	14523

SYN Flood Attack



Controller	Total traffic transmitted by victim during TCP SYN flood (packets)	Total traffic received by victim during TCP SYN flood (packets)
L2 Learning	29911	59821
DoS Protection	30	10

Controller Throughput

Controller	*Average throughput with 1000 unique MAC addresses*	*Average throughput with 10000 unique MAC addresses*
L2 Learning	764.32 responses/s	688.32 responses/s
DoS Protection	294.34 responses/s	90.54 responses/s

Operation Checkpoint: SDN Application Control

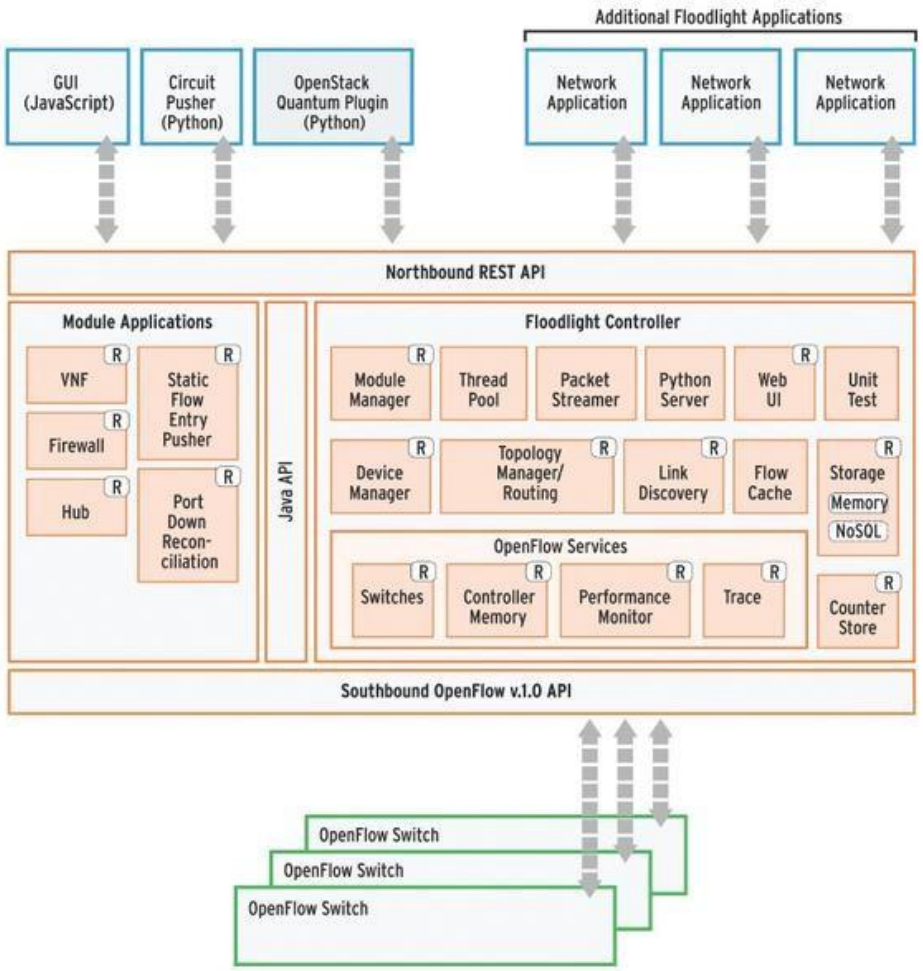
Fundamental security challenge is the ability for a malicious application to access network state information and manipulate network traffic for nefarious purposes.

Northbound Interface (NBI) Communication involves:

- Reading Network State
- Writing Network Policies

Objective: Protect against unauthorized control function access attempts

Floodlight Architecture



OpenFlow Controller Article, Floodlight Architecture and Relationships, http://www.admin-magazine.com/

Problem Description

Weaknesses in current approach:

- No authentication of RESTful API commands
- No scheme to ensure rules installed do not overlap or interfere with one another
- Applications do not have to provide identity information
- No application regulation or behaviour inspection after installation

Potential Solutions:

- Rule conflict detection and correction
- Application identification and priority enforcement
- Malicious activity detection and mitigation

System Attributes:

1. Define a complete set of permissions
2. Provide a secure storage structure for saving unique application IDs mapped to the set of permissions granted to that application
3. Provide a means for the network administrator/operator to add/remove application permissions (by its unique ID)
4. Provide a REST call for applications to query the controller and discover their assigned permissions
5. Secure the methods, in the Floodlight controller, that carry out the functions described by each of the permissions in the permission set
6. Log all unauthorized operation attempts to a log file for auditing purposes

Permissions Categorization

Category	Permission	Screening method(s)
Read	read_topology	**getAllSwitchMap:** Controller.java **getLinks:** LinkDiscoverManager.java
	read_all_flow	**getFlows:** StaticFlowEntryPusher.java
	read_statistics	**getSwitchStatistics:** SwitchResourceBase.java **getCounterValue:** SimpleCounter.java
	read_pkt_in_payload	**get:** FloodlightContextStore.java
	read_controller_info	**retrieve:** ControllerMemoryResource.java
Notification	pkt_in_event	**addToMessageListeners:** Controller.java **addListener:** ListenerDispatcher.java
	flow_removed_event	
	error_event	
Write	flow_mod_route	**insertRow:** AbstractStorageSource.java
	flow_mod_drop	**deleteRow:** AbstractStorageSource.java
	set_flow_priority	**insertRow:** AbstractStorageSource.java
	set_device_config	**setAttribute:** OFSwitchBase.java
	send_pkt_out	**write:** IOFSwitch.java **writeThrottled:** IOFSwitch.java
	flow_mod_modify_hdr	**parseActionsString:** StaticFlowEntries.java
	modify_all_flows	**setCommand:** OFFlowMod.java

Application Permissions Management:

Unique ID is key to access LinkedHashMap structure storing application permissions (encrypted and serialized)

Application Permissions Interrogation:

```
ckane@ckane-VirtualBox:~/floodlight$ java -cp target/floodlight.jar security.PermissionsCLI -help

User requires help using PermissionsCLI

usage: permissionsCLI
 -help                 Display help information
 -id <arg>             Application ID
 -permissions <arg>    List of permissions
 -set                  Set application permissions
 -unset                Unset application permissions

Valid Permissions: read_topology, read_all_flow, read_statistics, read_pkt_in_payload, read_controller_info,
pkt_in_event, flow_removed_event, error_event, topology_event, flow_mod_route, flow_mod_drop, flow_mod_modify
_hdr, modify_all_flows, send_pkt_out, set_device_config, set_flow_priority, "ALL" (grants all permissions to
application)

Set Example: permissionCLI -set -id <application-id> -permissions <list of permissions>
Unset Example: permissionCLI -unset -id <application-id>
```

Application Permissions Querying:

REST URI: */wm/security/<id>/permissions/json*

Operation Checkpoint:

Floodlight Method *getAllSwitchMap* has been modified to incorporate the new security mechanism

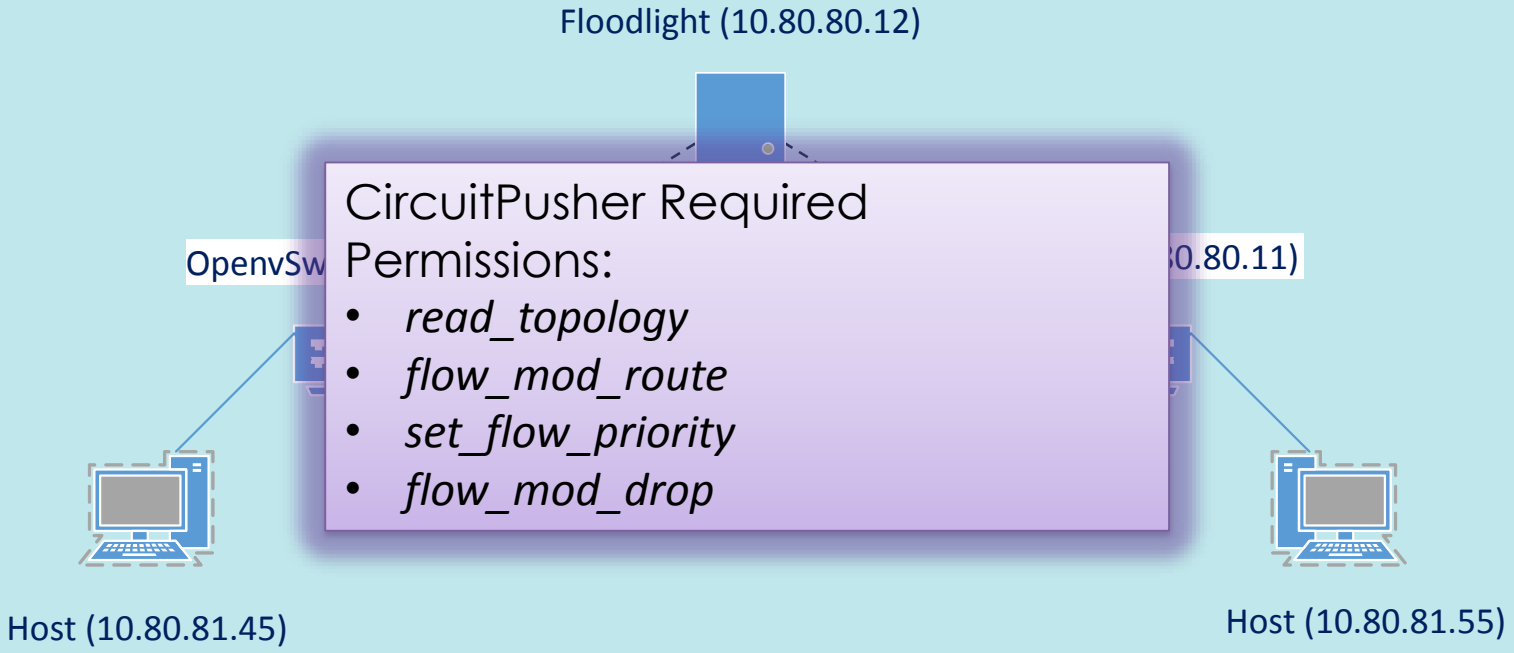
```
1391        public Map<Long,IOFSwitch> getAllSwitchMap(String appId) {
1392            Map<Long,IOFSwitch> switches =
1393                    new HashMap<Long, IOFSwitch>(this.syncedSwitches);
1394            OperationCheckpoint opChkpt = new OperationCheckpoint();
1395            if (opChkpt.isOperationPermitted("read_topology", appId)) {
1396                if (this.role != Role.SLAVE) {
1397                    switches.putAll(this.activeSwitches);
1398                }
1399            }
1400            return switches;
1401        }
```

Unauthorized Operations Log:

<date><time><applicationID><deniedpermission>

CircuitPusher Example (1/5)

***CircuitPusher** …"utilizes Floodlight REST APIs to create a bidirectional circuit, i.e. permanent flow entry, on all switches in route between two devices based on IP addresses with specified priority"*



Floodlight Controller, User Documentation -> REST Applications -> Circuit Pusher, 19 Nov 2012, bit.ly/1qZ1Rjk/

With no permissions granted to *circuitpusher*, the attempt to add a bidirectional circuit fails in an attempt to retrieve switch details:

```
admin2@sdn02:~/floodlight$ ./apps/circuitpusher/circuitpusher.py --controller=10.80.80.12:8080 --type ip --src 10.80.8
1.45 --dst 10.80.81.55 --add --name testCircuit
Namespace(action='add', circuitName='testCircuit', controllerRestIp='10.80.80.12:8080', dstAddress='10.80.81.55', srcA
ddress='10.80.81.45', type='ip')
curl -s http://10.80.80.12:8080/wm/device/circuitpusher/?ipv4=10.80.81.45

Traceback (most recent call last):
  File "./apps/circuitpusher/circuitpusher.py", line 99, in <module>
    sourceSwitch = parsedResult[0]['attachmentPoint'][0]['switchDPID']
IndexError: list index out of range
```

After the *read_topology* permission is added, the initial commands of the application complete successfully:

```
admin2@sdn02:~/floodlight$ java -cp target/floodlight.jar security.PermissionsCLI -set -id circuitpusher -permissions
read_topology

 Application ID: circuitpusher
 Operation: Set
 Permissions:
   read_topology

admin2@sdn02:~/floodlight$ ./apps/circuitpusher/circuitpusher.py --controller=10.80.80.12:8080 --type ip --src 10.80.8
1.45 --dst 10.80.81.55 --add --name testCircuit
Namespace(action='add', circuitName='testCircuit', controllerRestIp='10.80.80.12:8080', dstAddress='10.80.81.55', srcA
ddress='10.80.81.45', type='ip')
curl -s http://10.80.80.12:8080/wm/device/circuitpusher/?ipv4=10.80.81.45

curl -s http://10.80.80.12:8080/wm/device/circuitpusher/?ipv4=10.80.81.55
```

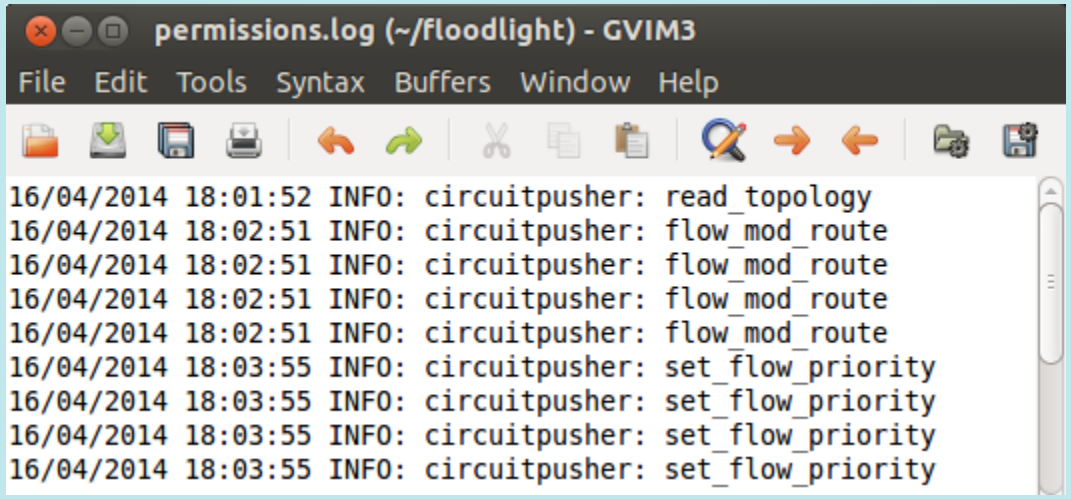
However, *ovs-ofctl dump-flows <dpid>* shows switch flow table empty

Once the remaining permissions are added (*flow_mod_route* and *set_flow_priority*), the circuit is installed correctly with flow rules installed at the switches:

```
admin2@sdn02:~/floodlight$ sudo ovs-ofctl dump-flows br2
NXST_FLOW reply (xid=0x4):
 cookie=0xa0000000000000, duration=28.544s, table=0, n_packets=0, n_bytes=0, ip,in_port=3,nw_src=10.80.81.55,nw_dst=10
.80.81.45 actions=output:1
 cookie=0xa0000000000000, duration=28.589s, table=0, n_packets=0, n_bytes=0, ip,in_port=1,nw_src=10.80.81.45,nw_dst=10
.80.81.55 actions=output:3
 cookie=0xa0000000000000, duration=28.567s, table=0, n_packets=0, n_bytes=0, arp,in_port=1 actions=output:3
 cookie=0xa0000000000000, duration=28.52s, table=0, n_packets=0, n_bytes=0, arp,in_port=3 actions=output:1
admin2@sdn02:~/floodlight$
```

CircuitPusher Example (5/5)

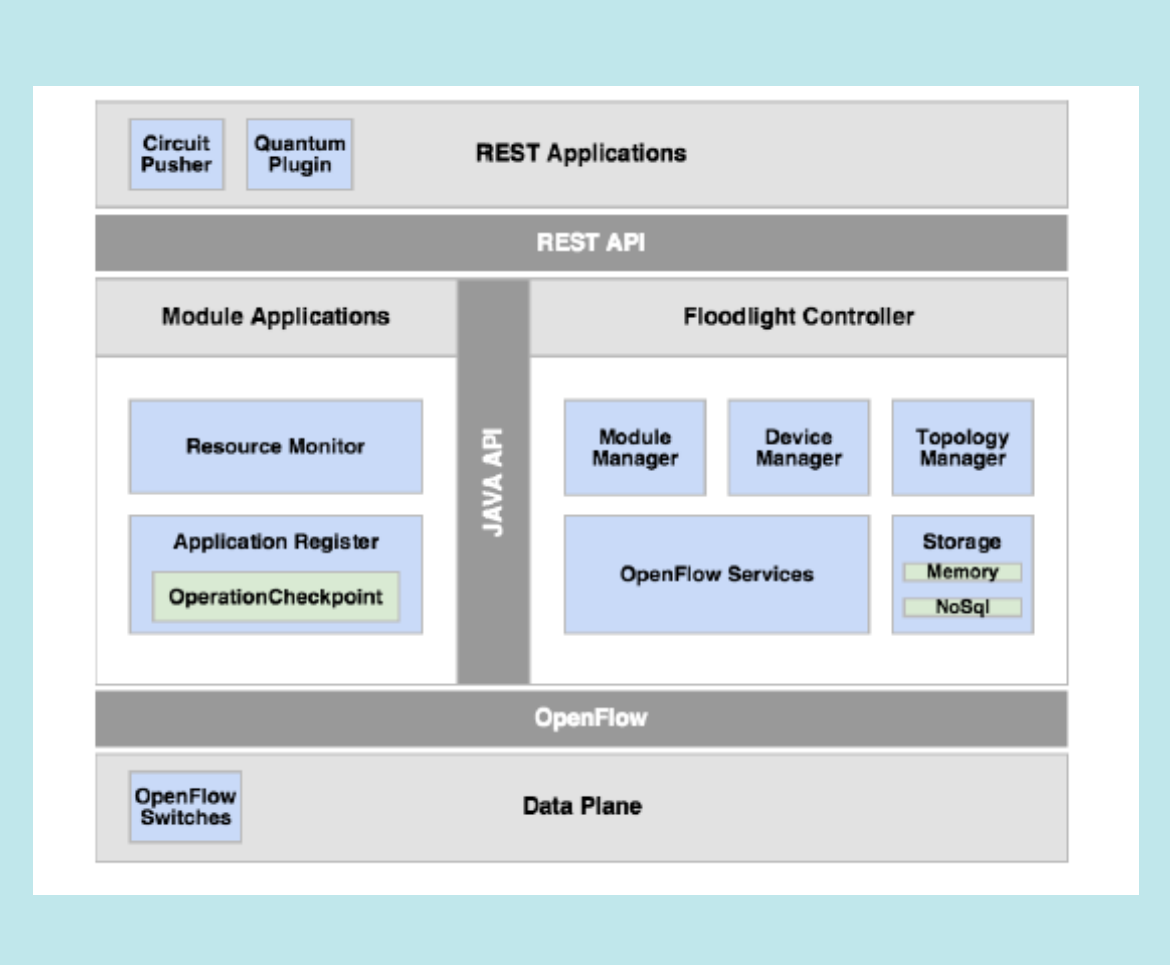
The log file holds the record of the unauthorized *circuitpusher* access attempts:



OperationCheckpoint introduces limited latency to the Floodlight Controller:

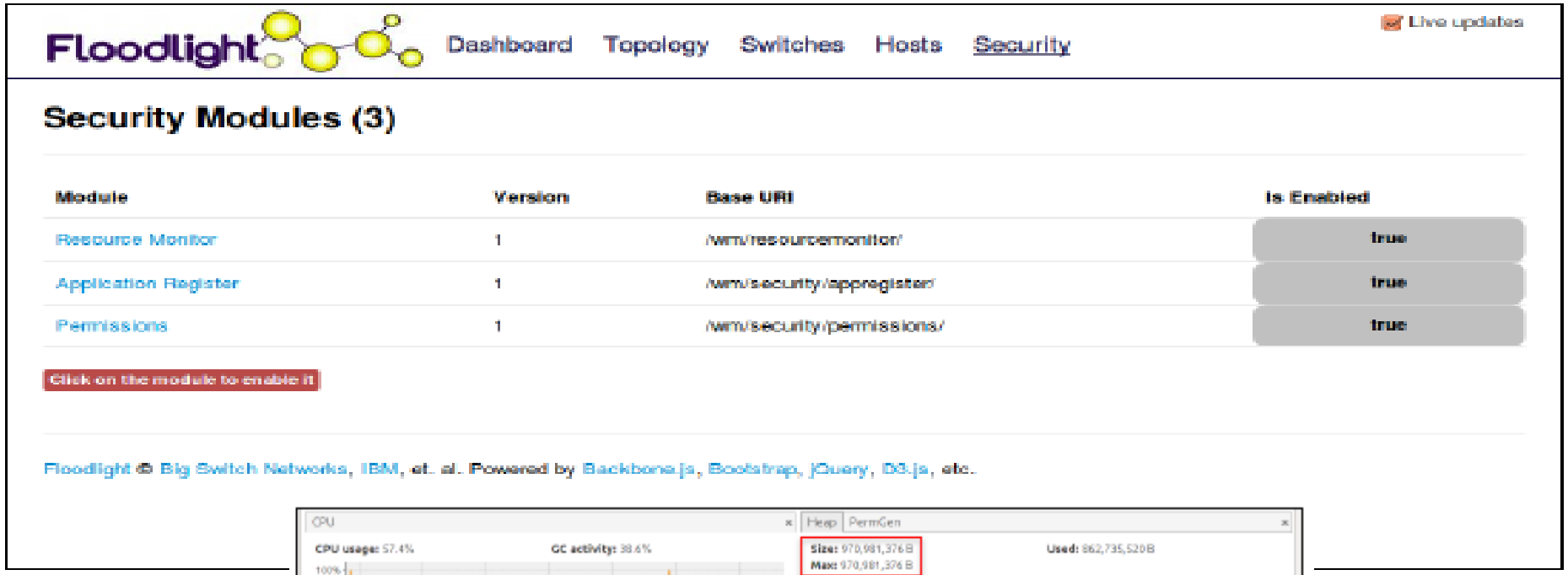
	Avg.	Std. Dev.
Execution Time *(μs)* without *OperationCheckpoint*	5.625	2.955
Execution Time *(μs)* with *OperationCheckpoint*	372.750	103.191
Latency *(μs)*	367.125	102.437

App Register/Resource Monitor




```
 ____                  _________                  __                  __
/  _  \ _____ _____   \_____   \ ____   ____ |__| _____/  |_ _________
/  /_\  \\____ \\____ \   |      _// __ \ / ___\|  |/  ___/\   __\/ __ \_  __ \
/    |    \  |_> >  |_> >  |    |   \  ___// /_/  >  |\___ \  |  | \  ___/|  | \/
\____|__  /   __/|   __/   |____|_  /\___  >___  /|__/____  > |__|  \___  >__|
        \/|__|   |__|             \/     \/_____/         \/           \/
                    Application Register for Floodlight

<Main> (R)egister, (U)nregister, (L)auncher, (P)ermissions, (C)heck, (E)xit. Enter an option: c

<Check>
Currently registered applications [circuitpusherID, test], instances [cp2, cp1, test_app]
Enter application/instance ID: circuitpusherID
Application [circuitpusherID] attributes:
registered  true
arguments   true
permissions true
path        /home/rmg6/floodlight-0.91/apps/circuitpusherID/circuitpusherID.py
hash        998867cbd3f9e8a32d20270a6e9c7ae556008d5caff9381a92656fb31dbe0db3
instances   [cp2, cp1]

<Main> (R)egister, (U)nregister, (L)auncher, (P)ermissions, (C)heck, (E)xit. Enter an option: c

<Check>
Currently registered applications [circuitpusherID, test], instances [cp2, cp1, test_app]
Enter application/instance ID: test_app
Instance [test_app] attributes:
permissions false
launched    false
app_id      test

<Main> (R)egister, (U)nregister, (L)auncher, (P)ermissions, (C)heck, (E)xit. Enter an option: p

<Permissions> (S)et, (U)nset, (C)heck, (B)ack to main menu. Enter an option: s
Currently registered applications [circuitpusherID, test]
Enter Application ID: test
Current permissions of [test] application:
read_topology  false
read_all_flow  false
read_statistics  false
read_pkt_in_payload  false
```

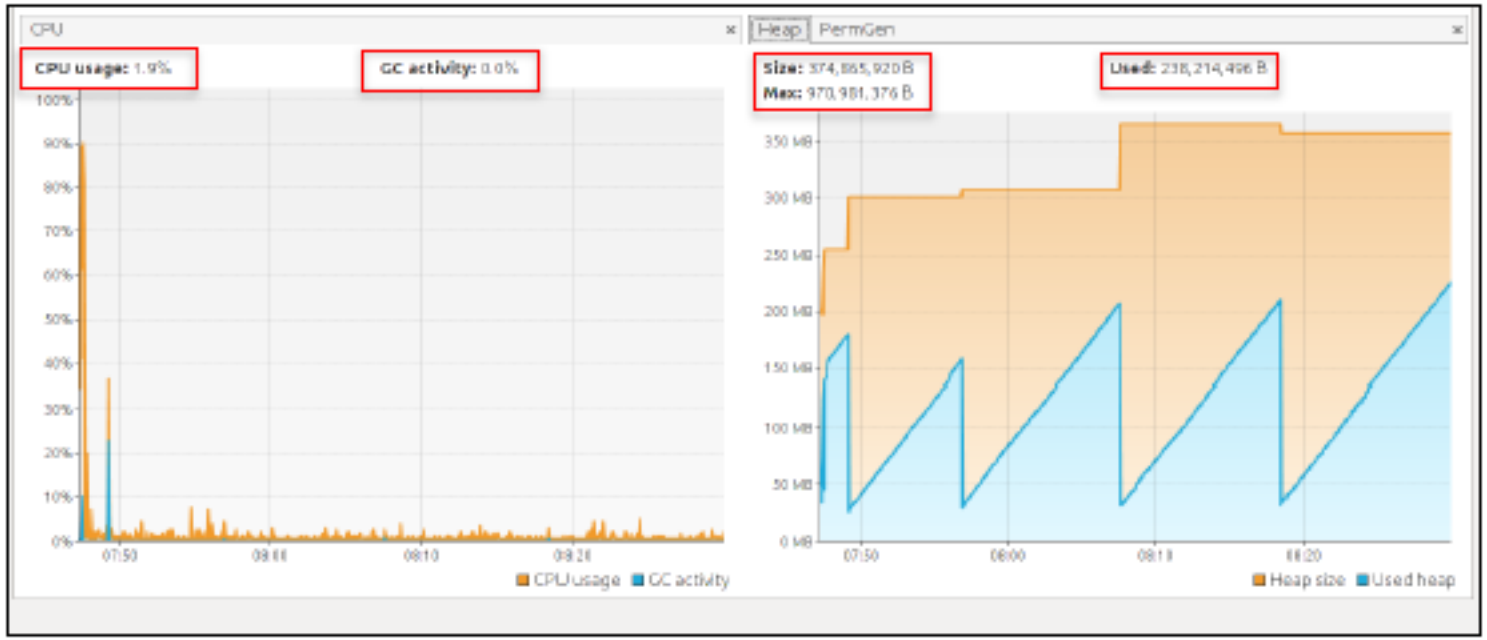
App Register/Resource Monitor



OperationCheckpoint and AppRegister

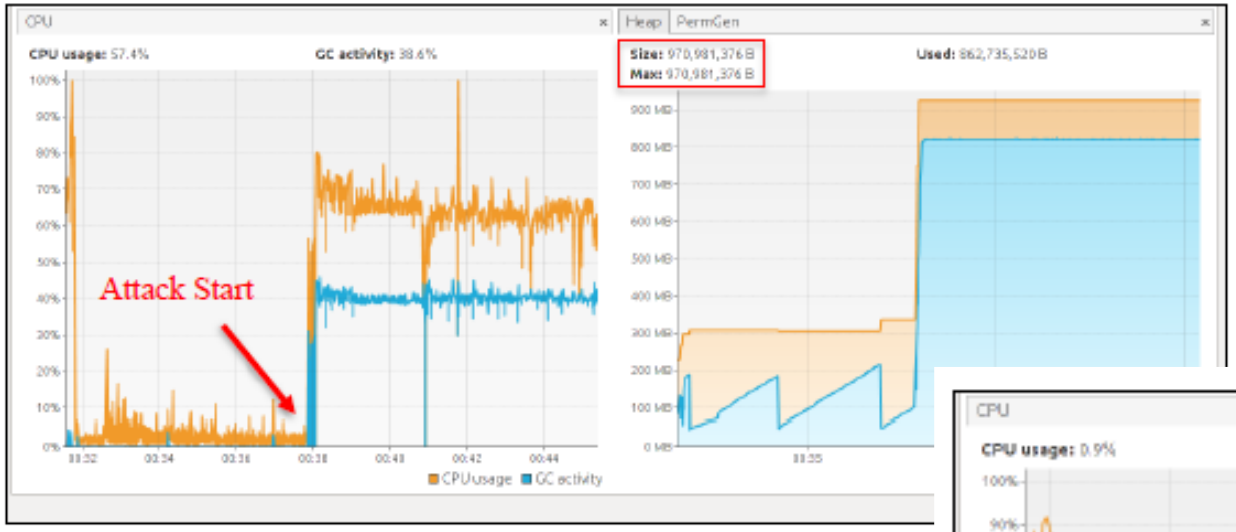
DEMO

Resource Monitor vs. Resource Exhaustion Attacks

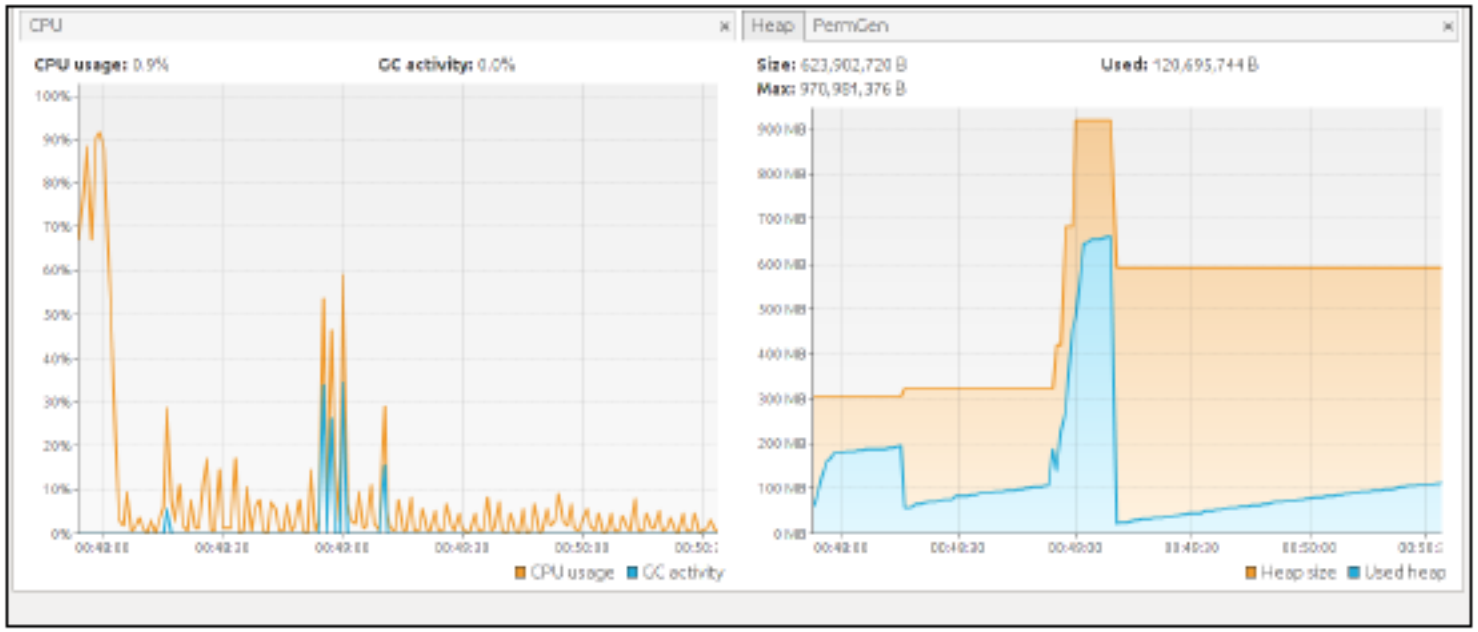


Floodlight Regular Resource Consumption

Resource Monitor vs. Resource Exhaustion Attacks

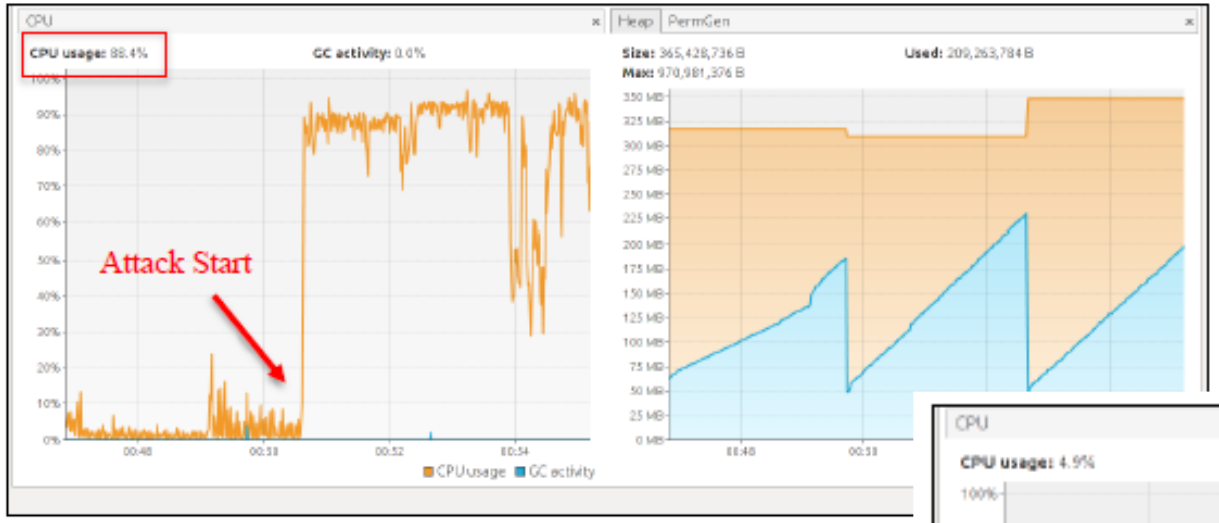


Memory Exhaustion Attack

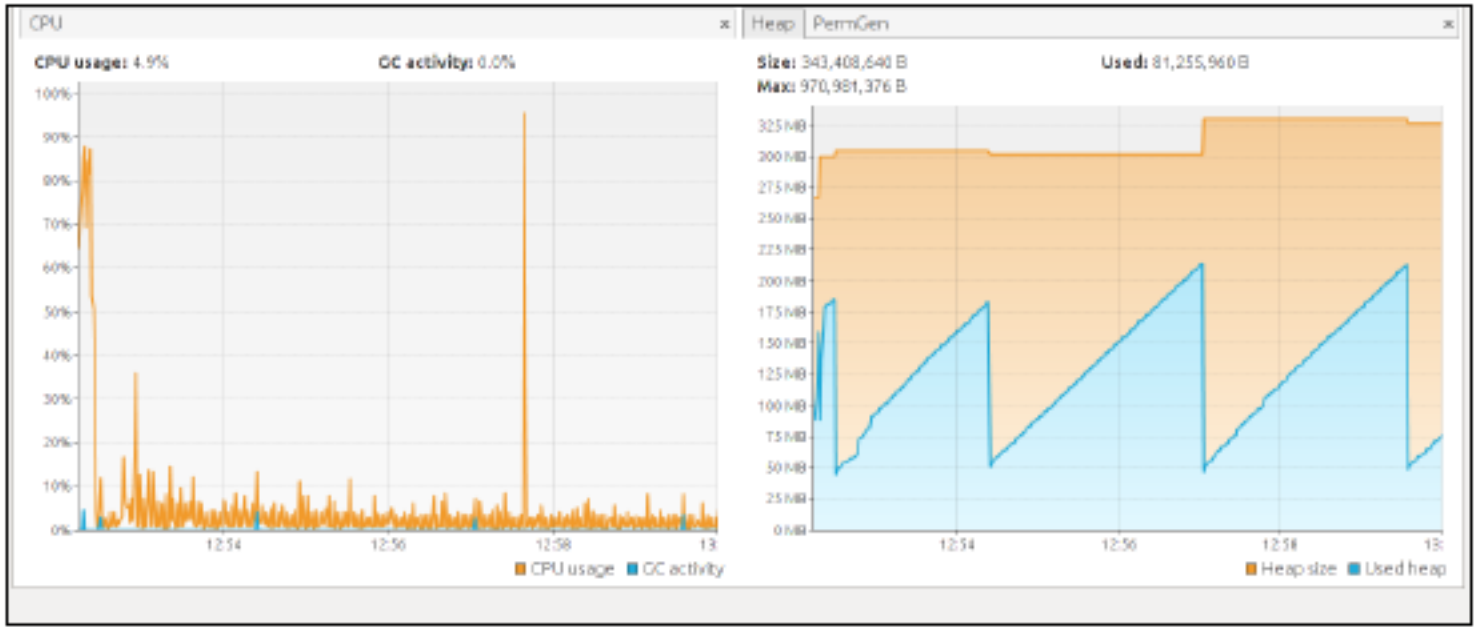


Memory Exhaustion Attack killed by Resource Monitor

Resource Monitor vs. Resource Exhaustion Attacks



CPU Exhaustion Attack



CPU Exhaustion Attack killed by Resource Monitor

Resource Monitor vs. Resource Exhaustion Attacks

```
12:15:53.633 INFO [attacks.Attacks:Dispatcher: Thread-22] Setting Attacks to true
12:16:38.565 INFO [attacks.Attacks:New I/O server worker #2-3] [ATTACK] Mem Exhaustion: ClassLoaderLeak
```

```
12:27:09.916 INFO [attacks.Attacks:Dispatcher: Thread-24] Setting Attacks to true
12:46:23.068 INFO [attacks.Attacks:New I/O server worker #2-3] [ATTACK] CPU exhaustion
```

```
java.lang.OutOfMemoryError: Java heap space
        at attacks.ClassLoaderLeakExample$LoadedInChil
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:274)
        at attacks.ClassLoaderLeakExample.loadAndDisca
        at attacks.ClassLoaderLeakExample$LongRunningT
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
java.lang.OutOfMemoryError: Java heap space
```

```
*** Adding links:
(s1, s2) (s1, s3) (s2, h1) (s2, h2) (s3, h3) (s3, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3
*** Starting CLI:
mininet> h1 ping h4 -c1
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable

--- 10.0.0.4 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss

mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X
h2 -> h1 X X
h3 -> X X h4
h4 -> X X h3
*** Results: 66% dropped (4/12 received)
mininet>
```

threats.log (~/floodlight-0.91) - gedit

Open Save Undo

threats.log

```
2015-05-18 11:18:43.727 WARN [ResourceMonitor] Application [ClassLoaderLeakExample] exceeds [Memory_Usage] soft limit
2015-05-18 11:18:44.364 WARN [ResourceMonitor] Application [ClassLoaderLeakExample] exceeds [Memory_Usage] hard limit
2015-05-17 22:54:33.075 WARN [ResourceMonitor] Application [CPUExhaustion] exceeds [CPU_Usage] soft limit
2015-05-17 23:02:14.158 WARN [ResourceMonitor] Application [CPUExhaustion] exceeds [CPU_Usage] hard limit
```

End of Session 5