KAIST Graduate School of Information Security

Misuse/Attack Possibilities and Defense in Open Source Software-Defined Networking

Seungsoo Lee
NSS LAB

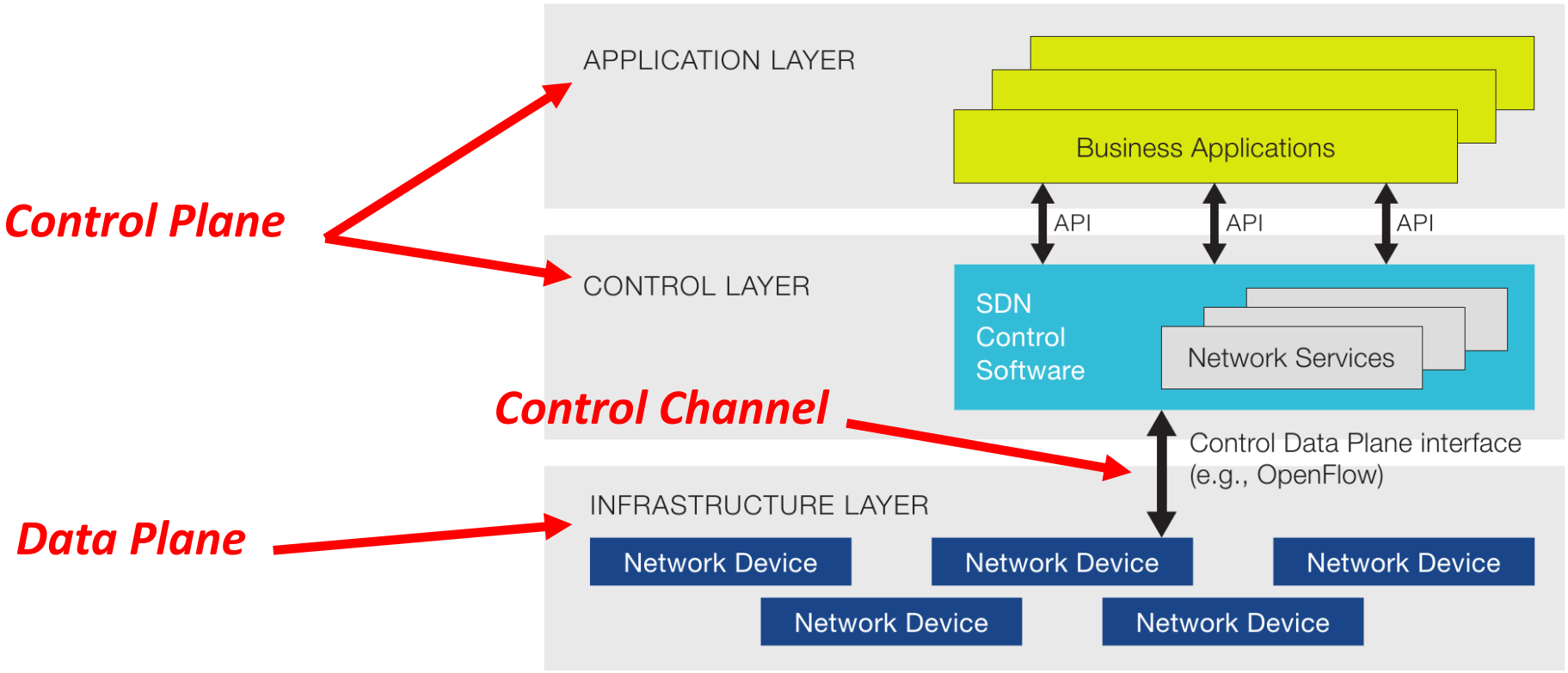
4/8/2016 (Fri.)

Open Networking Korea 2016 Spring

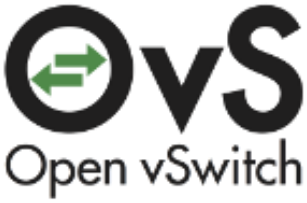
Contents

NS2 Network and System Security Laboratory

- SDN only leaves ***simple data plane functions*** (at which hardware is good) to a network device and moves complex and dynamic control plane functions into software applications at a separate box, a controller.

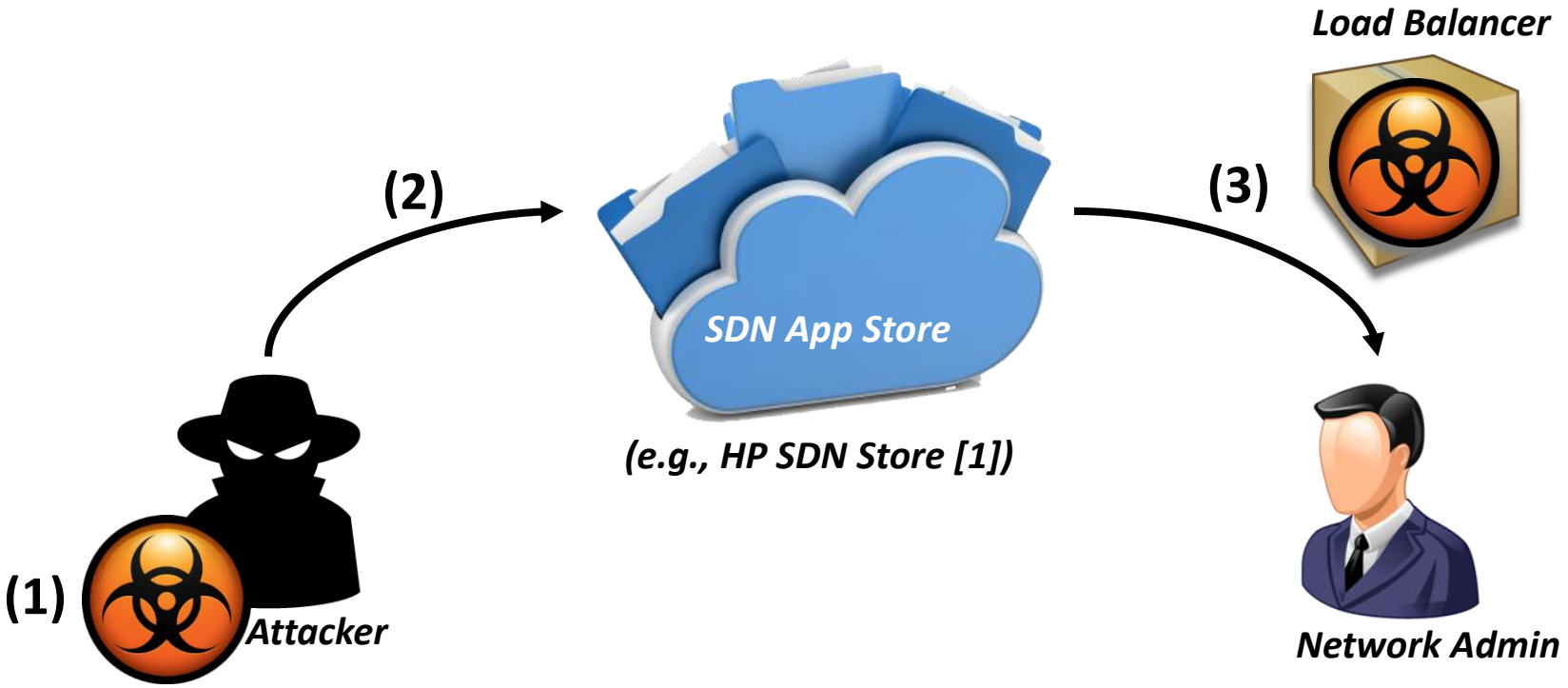


Open Source SDN Controller/Switch



- Controller (Network operating system) manages the entire SDN environment as the ***control plane***
- Popular ***open-source*** controllers (e.g., ONOS and OpenDaylight)

Motivation Example

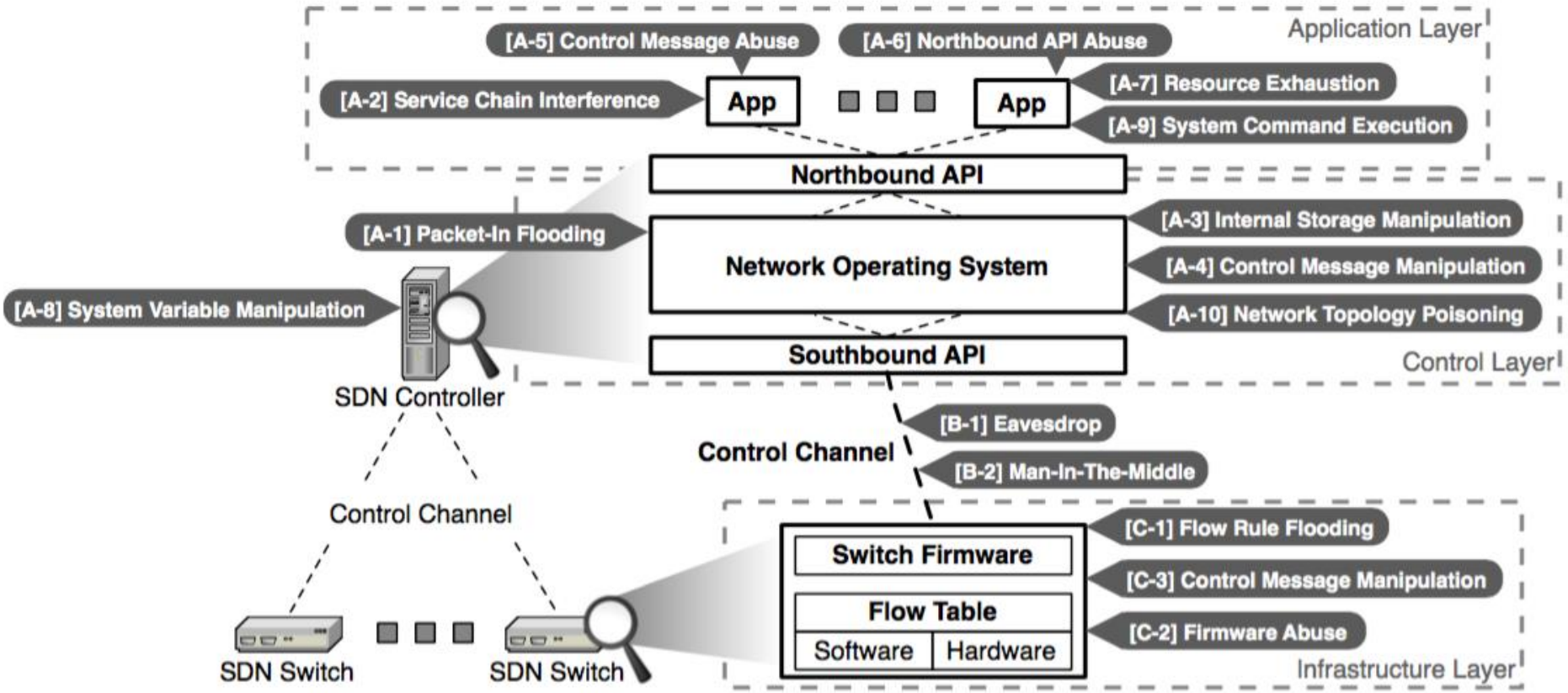


i) Controller plays a role as the traditional operating systems
→ There are also **security holes !**

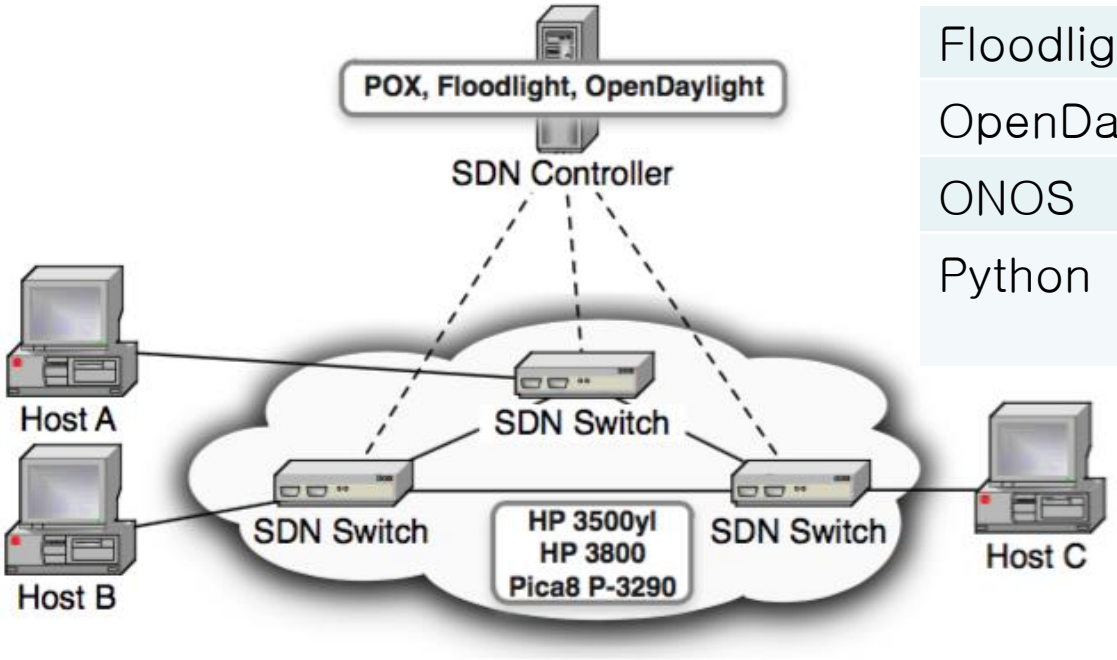
ii) Anyone can easily develop and distribute SDN applications

[1] HP. Hp sdn dev center: Sdn app store. http://www.hp.com/go/sdndevcenter.

Overview of Misuse/Attacks



- **Control Plane** : Application Layer + Control Layer (A-1 ~ A-10)
- **Control Channel** : Channel between Controller and Switch (B-1 ~ B-2)
- **Data Plane** : Intrastructure Layer (C-1 ~ C-3)



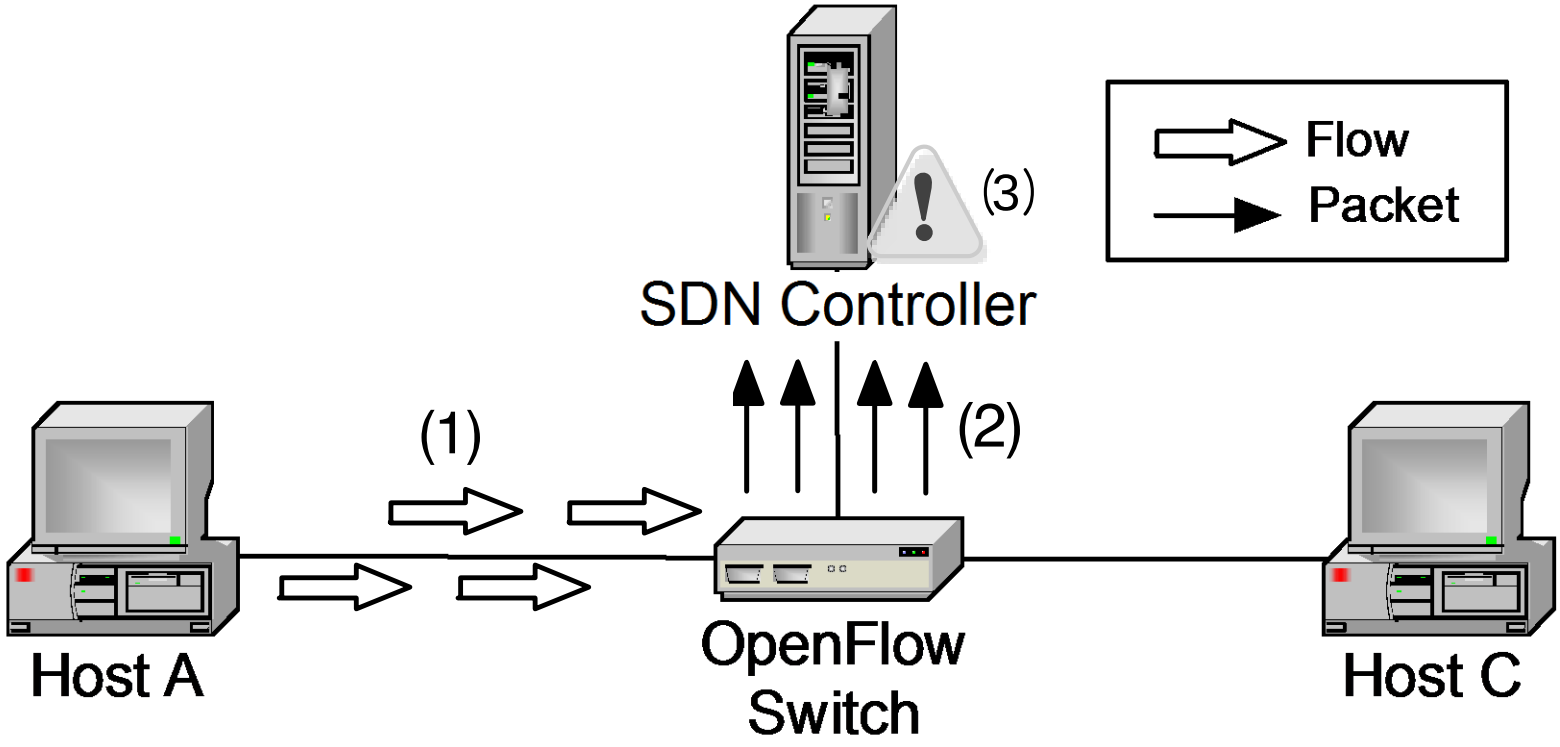
LIST OF TESTED SDN CONTROLLERS

Name	*Language*	*Version*
Floodlight	JAVA	v0.91
OpenDaylight	JAVA	helium-sr3
ONOS	JAVA	v1.1
Python	Python	Active: dart-2 01 4-summer

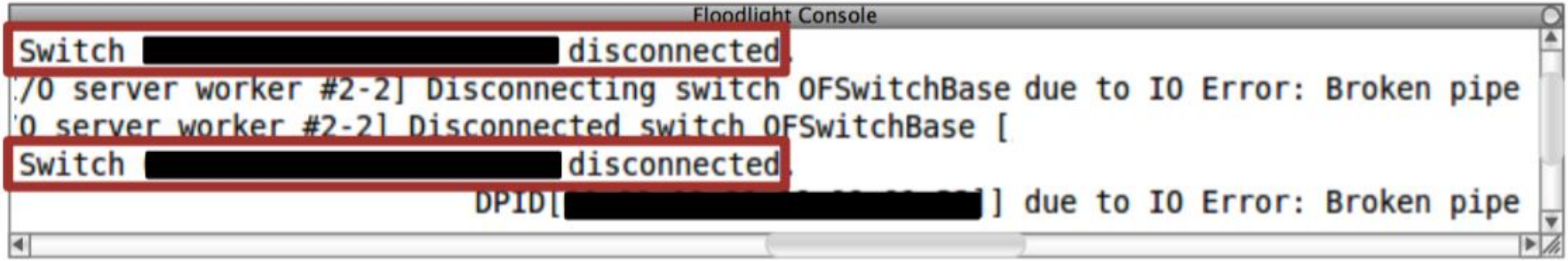
LIST OF TESTED SDN SWITCHES

Vendor	*Switch Model*	*Firmware Version*
HP	3500yl	K.15.12.0010
HP	3800	KA.15.13.0005
Pica8	P-3290	PicOS 2.3

- Scenario

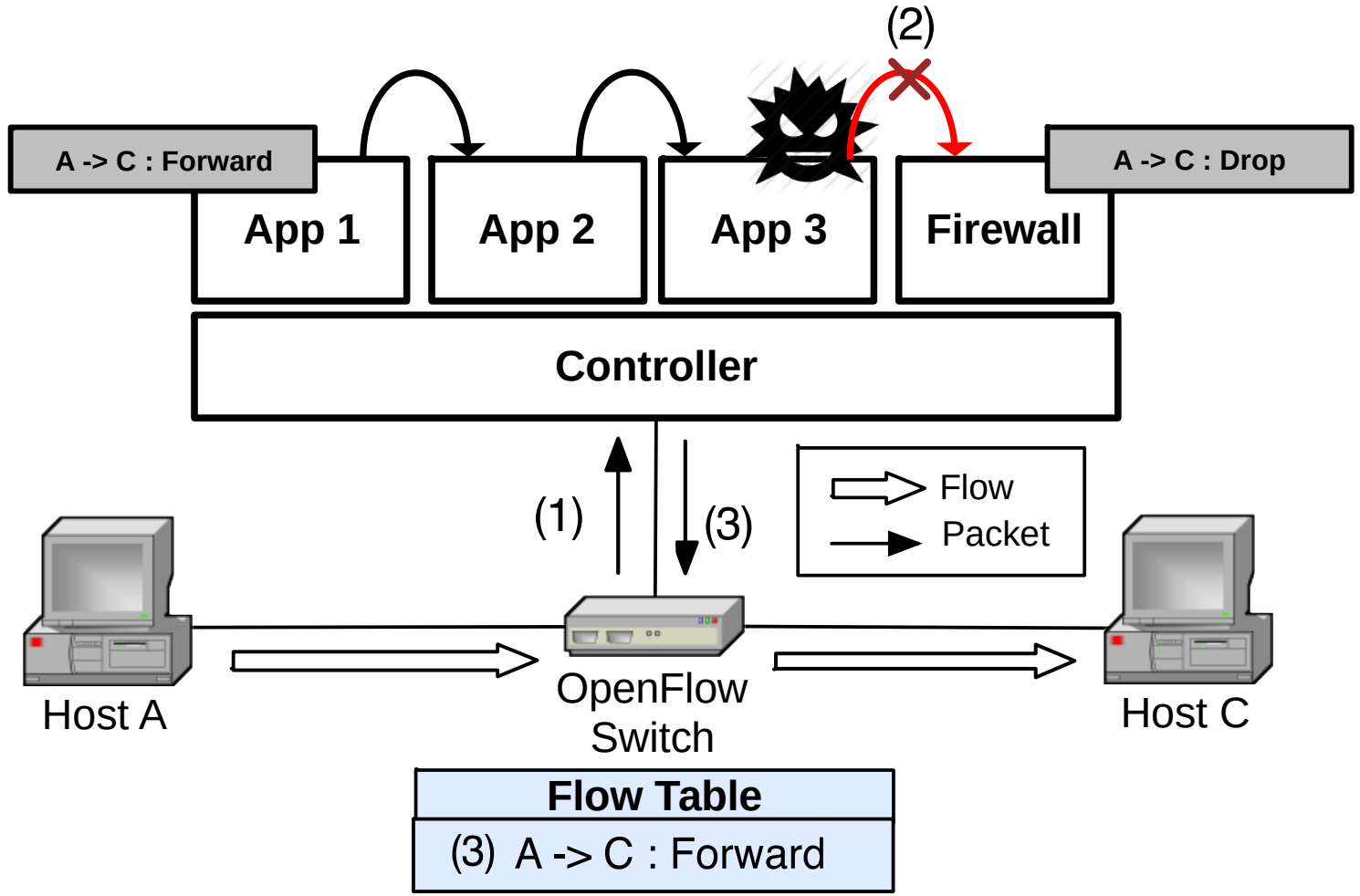

(3)
SDN Controller
Flow
Packet
(1)
(2)
Host A
OpenFlow
Switch
Host C

- Switch disconnection

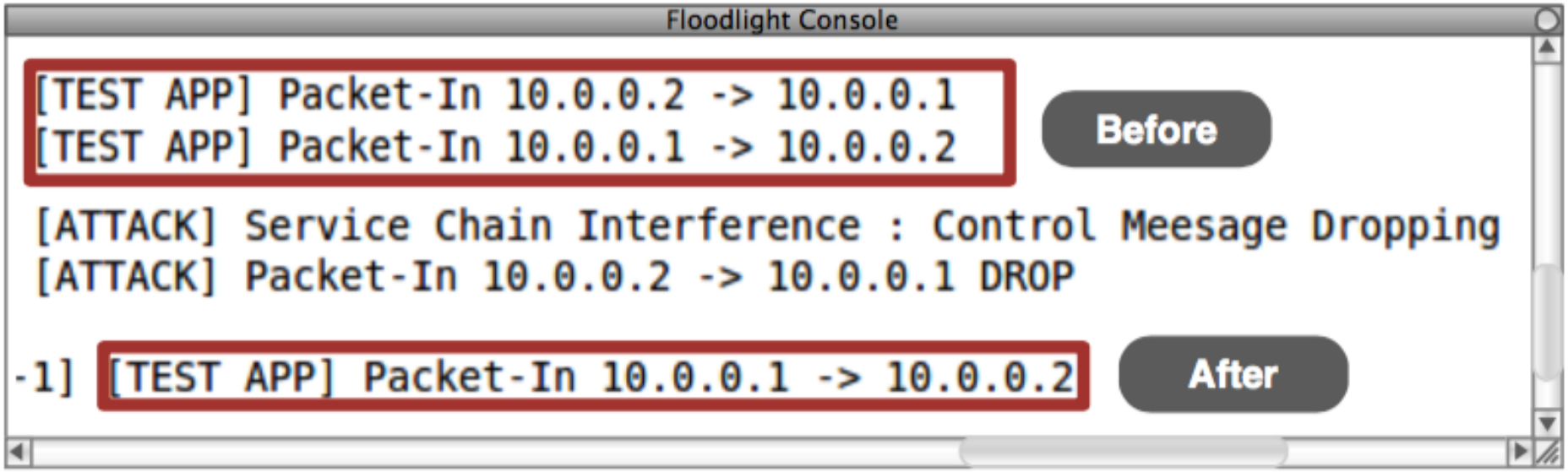


[A-2-i] Control Message Drop

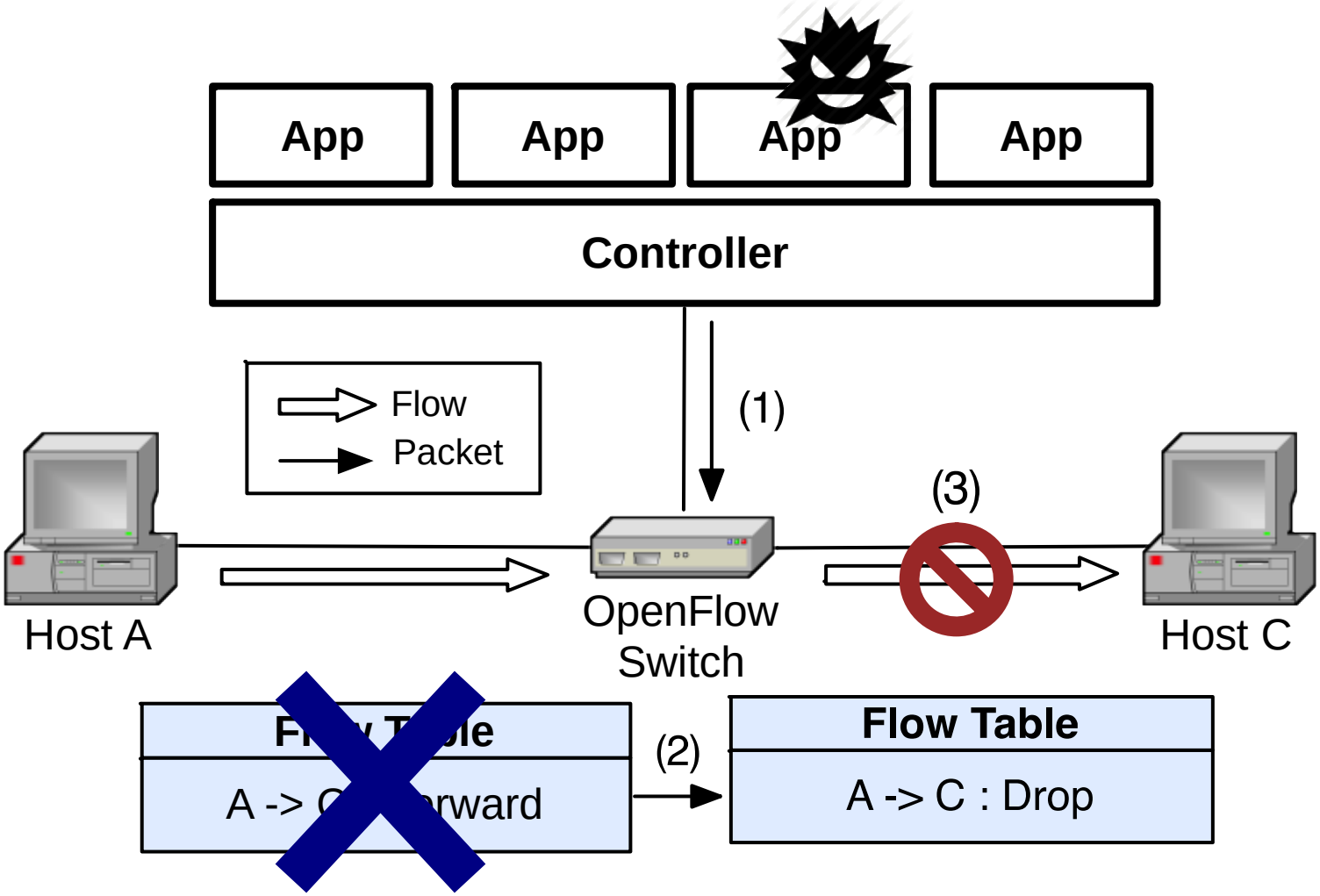
- Scenario



- Packet-In cut-off

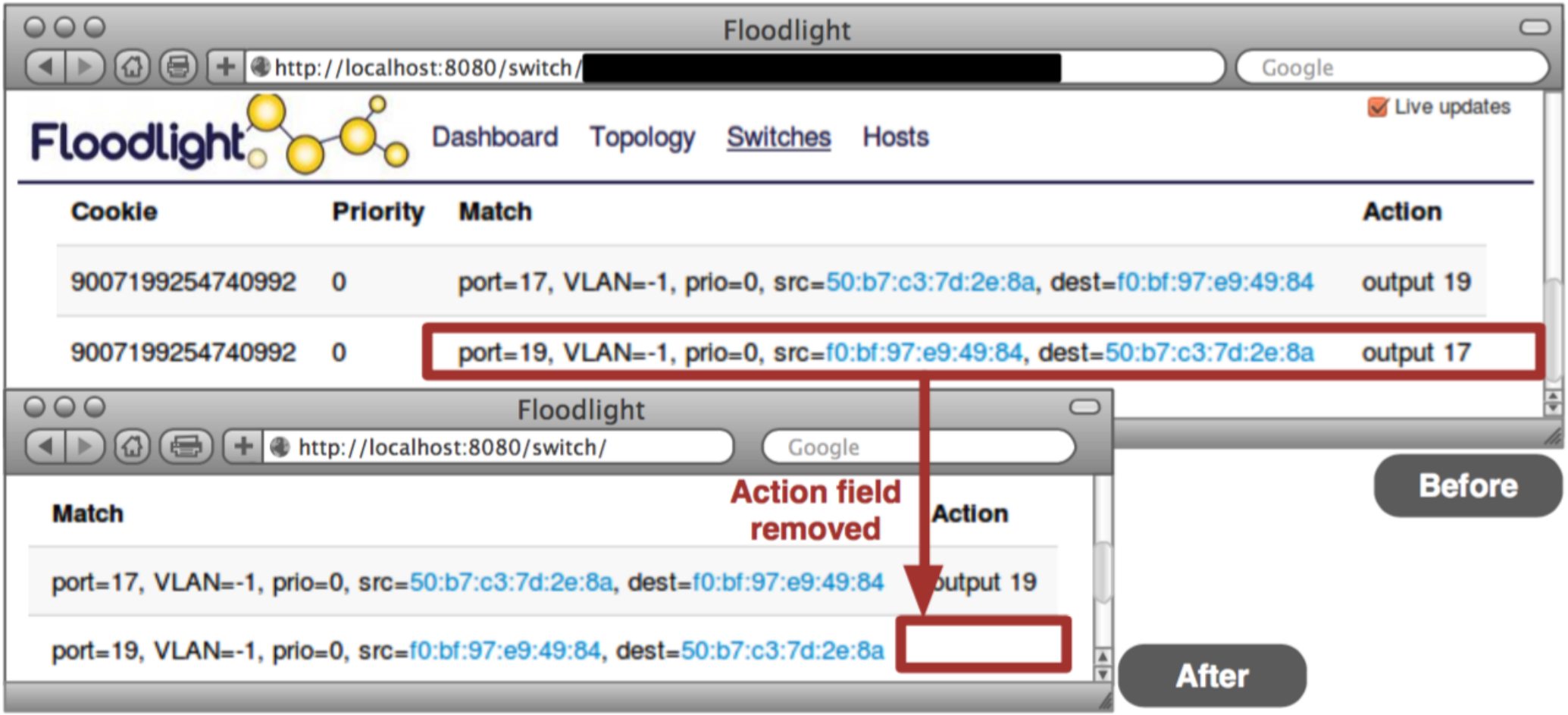


- Scenario

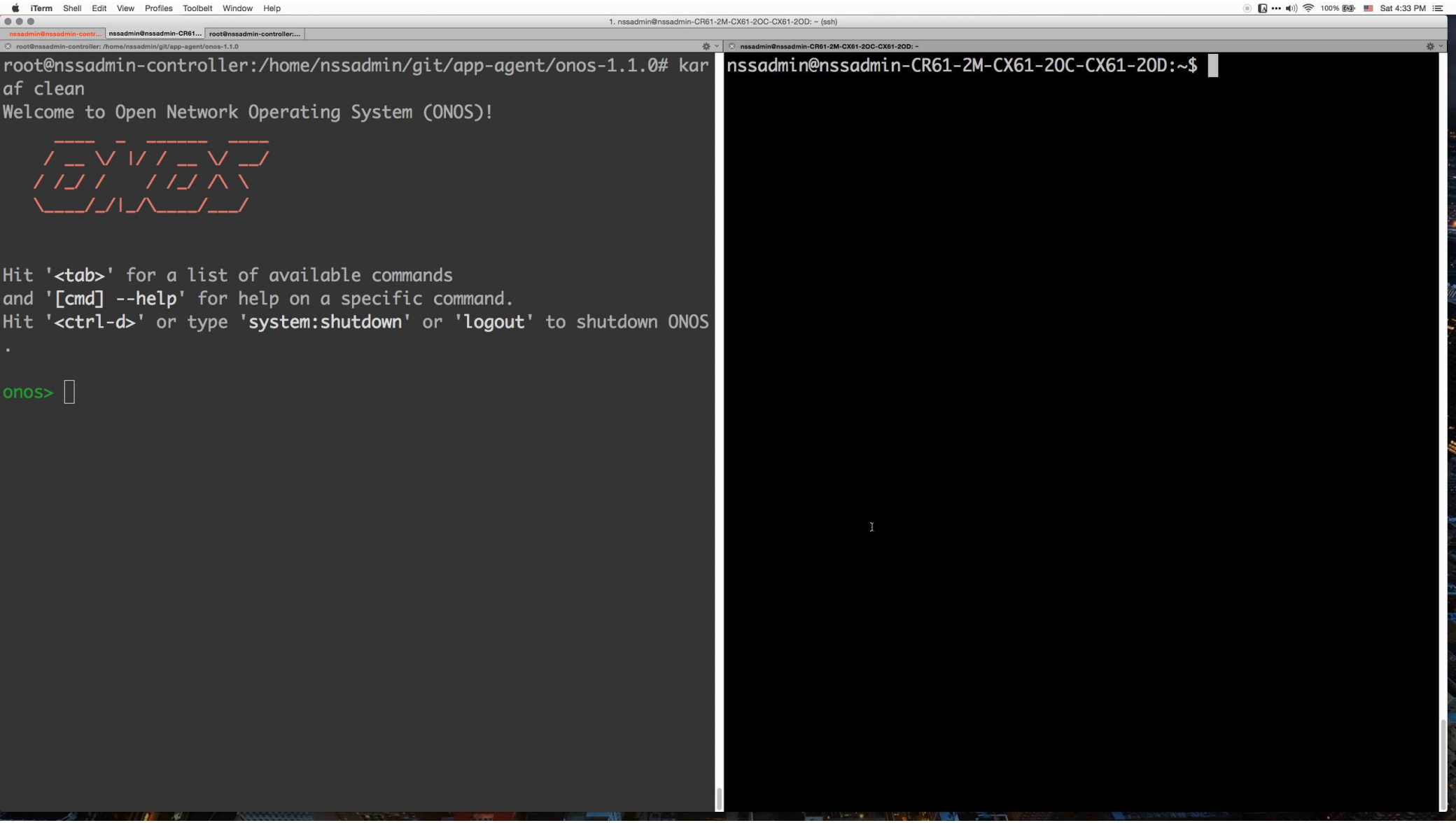

App
App
App
App
Controller
Flow
Packet
(1)
(2)
(3)
Host A
OpenFlow Switch
Host C
Flow Table
A -> C : Drop

[A-5-i] Flow Rule Modification

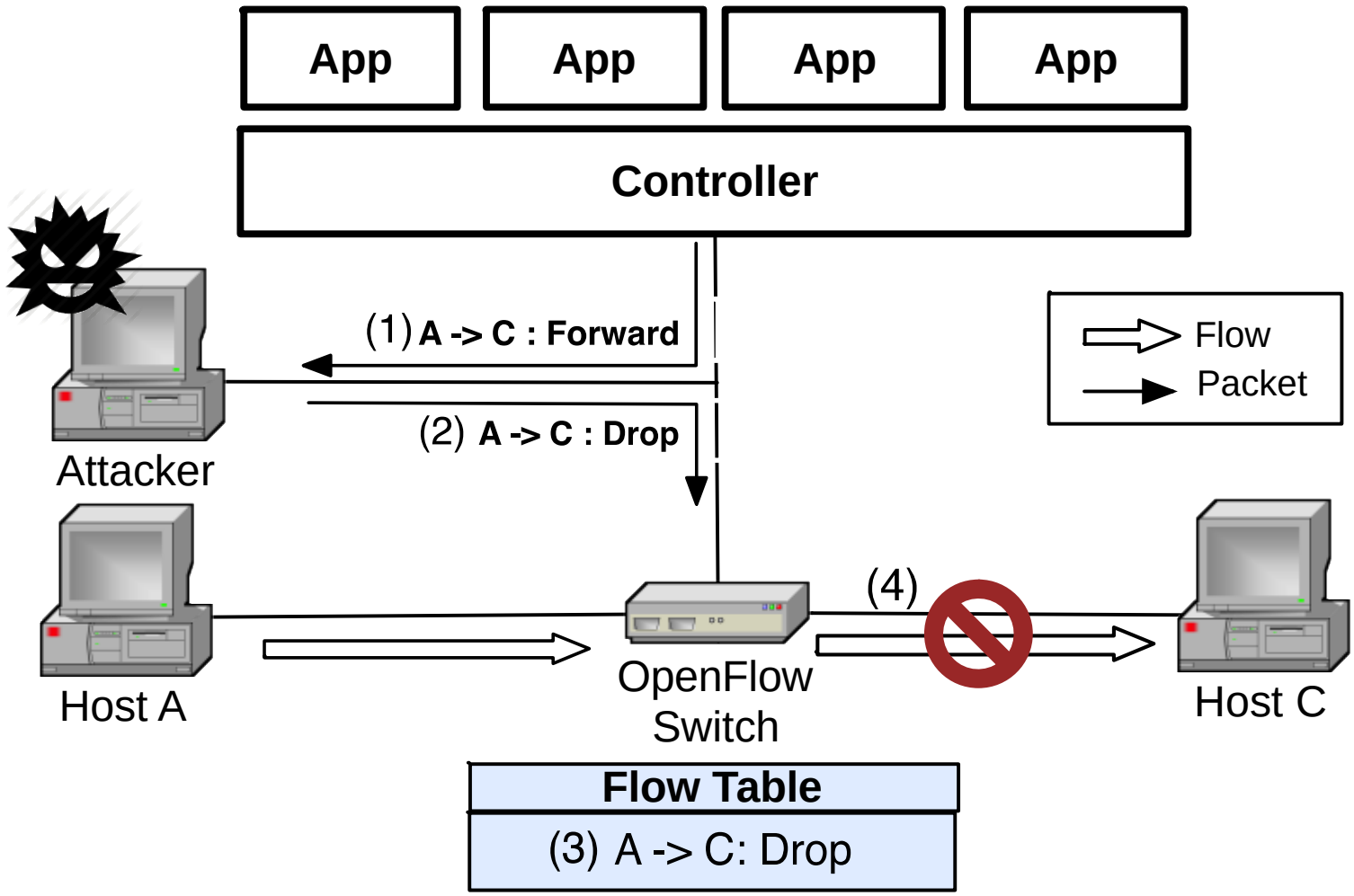
- Modification Result

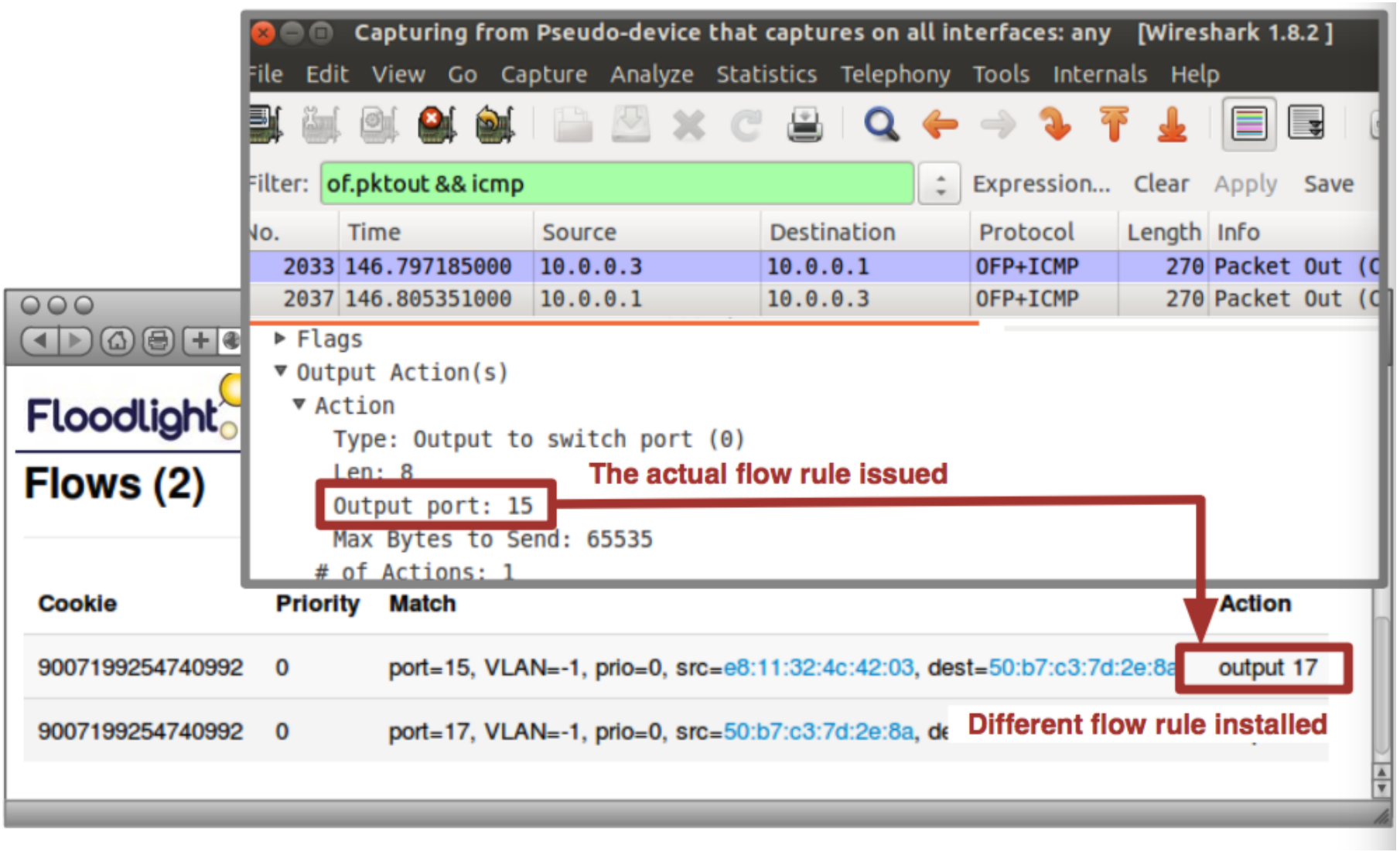


[A-5-i] Flow Rule Modification

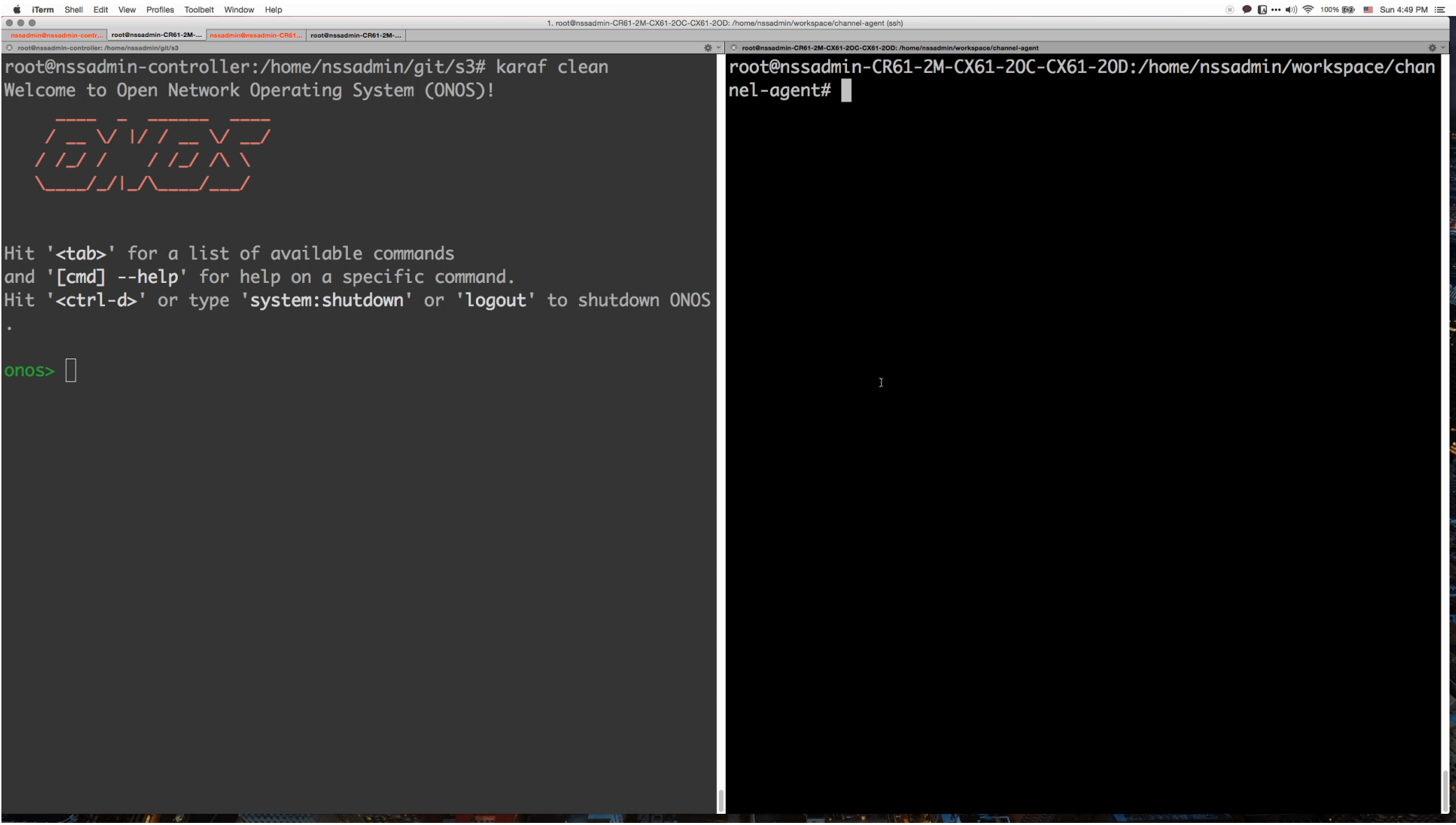


- Scenario


App
App
App
App
Controller
(1) A -> C : Forward
(2) A -> C : Drop
Flow
Packet
Attacker
Host A
OpenFlow Switch
(4)
Host C
Flow Table
(3) A -> C: Drop

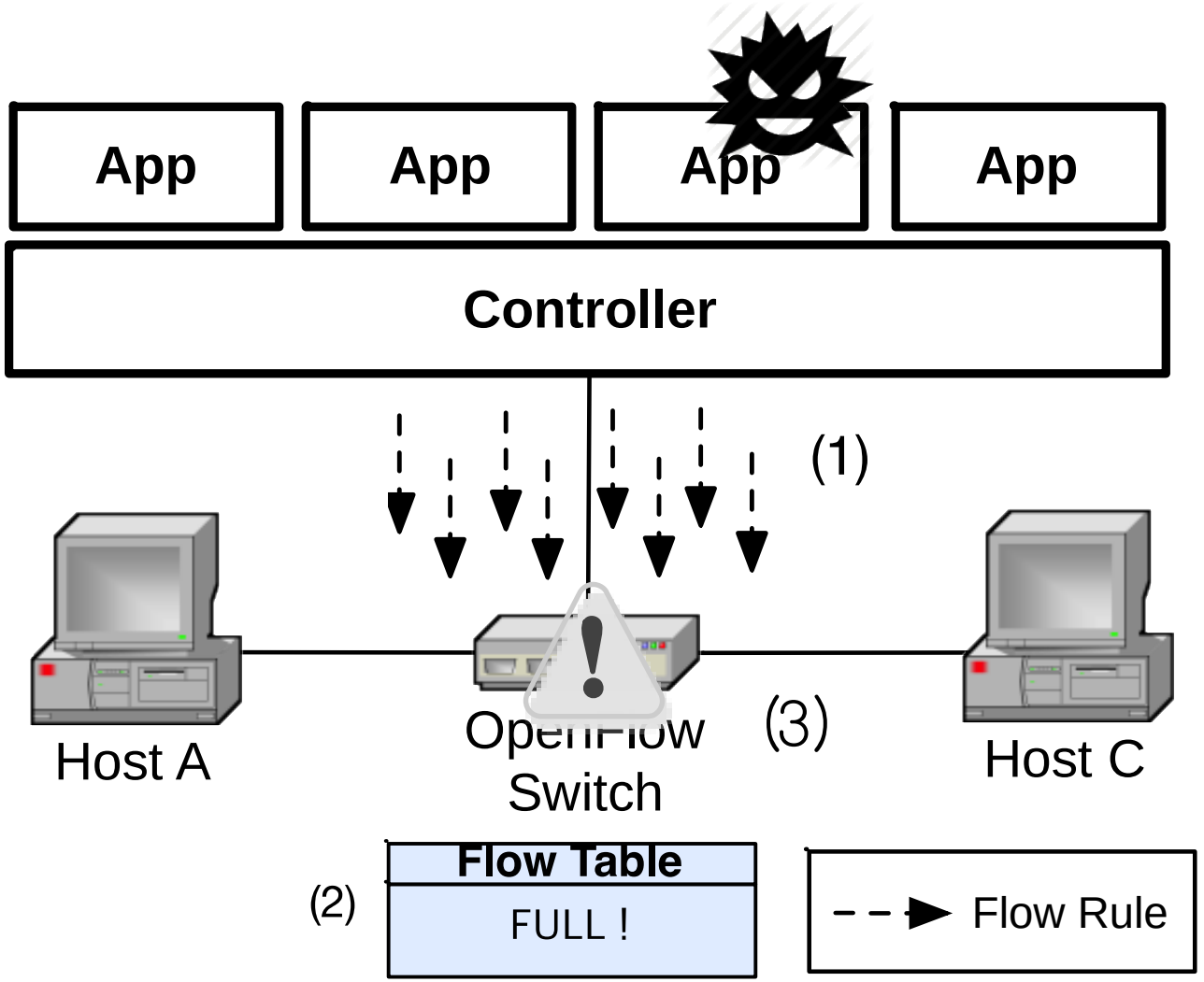
- 
Capturing from Pseudo-device that captures on all interfaces: any [Wireshark 1.8.2]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: of.pktout && icmp
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
2033 146.797185000 10.0.0.3 10.0.0.1 OFP+ICMP 270 Packet Out (C
2037 146.805351000 10.0.0.1 10.0.0.3 OFP+ICMP 270 Packet Out (C
Flags
Output Action(s)
Action
Type: Output to switch port (0)
Len: 8
Output port: 15
Max Bytes to Send: 65535
# of Actions: 1
The actual flow rule issued
Floodlight
Flows (2)
Cookie
Priority
Match
Action
9007199254740992 0 port=15, VLAN=-1, prio=0, src=e8:11:32:4c:42:03, dest=50:b7:c3:7d:2e:8a output 17
9007199254740992 0 port=17, VLAN=-1, prio=0, src=50:b7:c3:7d:2e:8a, de
Different flow rule installed

[B-2] MITM attack

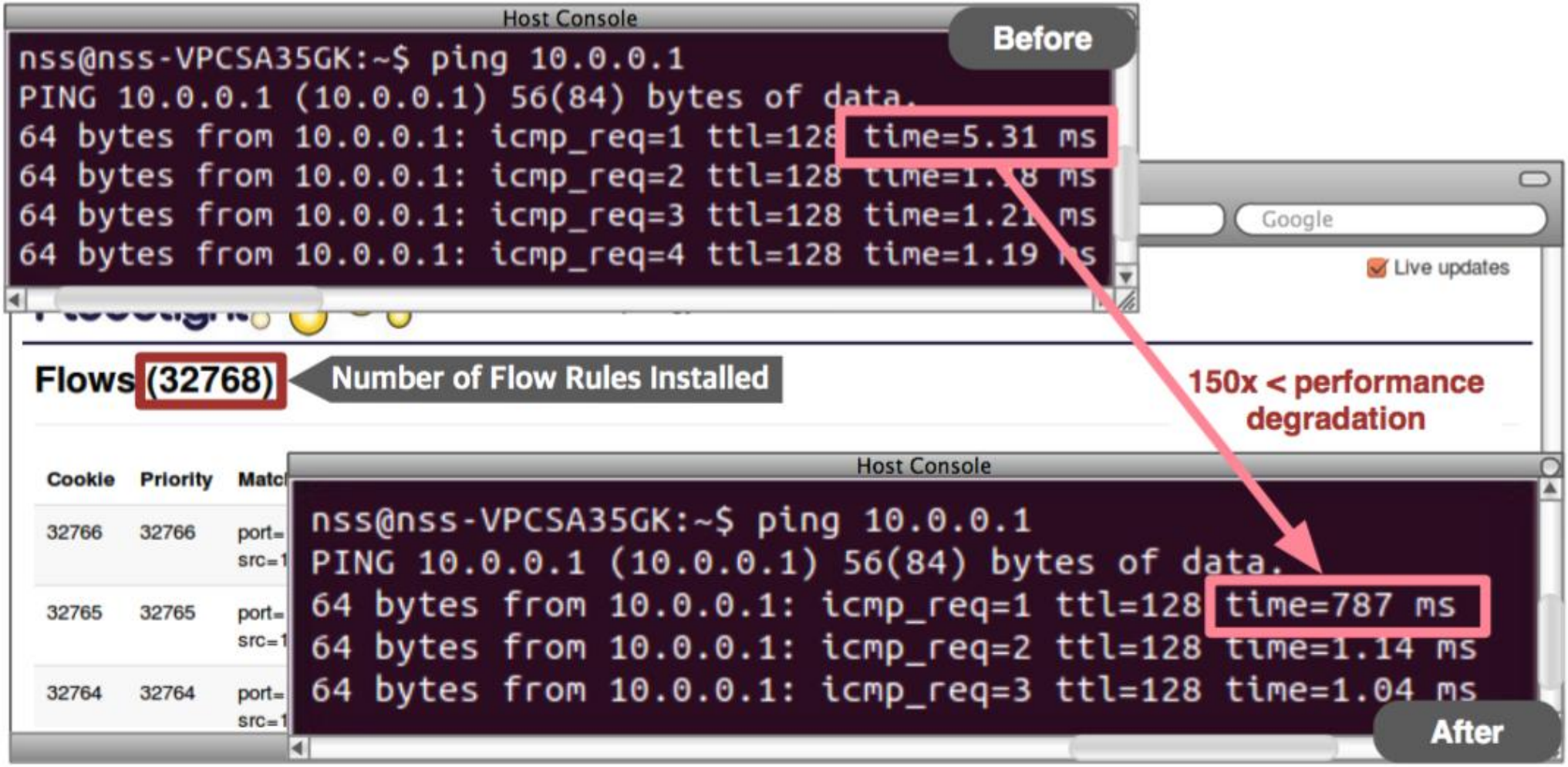


[C-1-A] Flow Rule Flooding

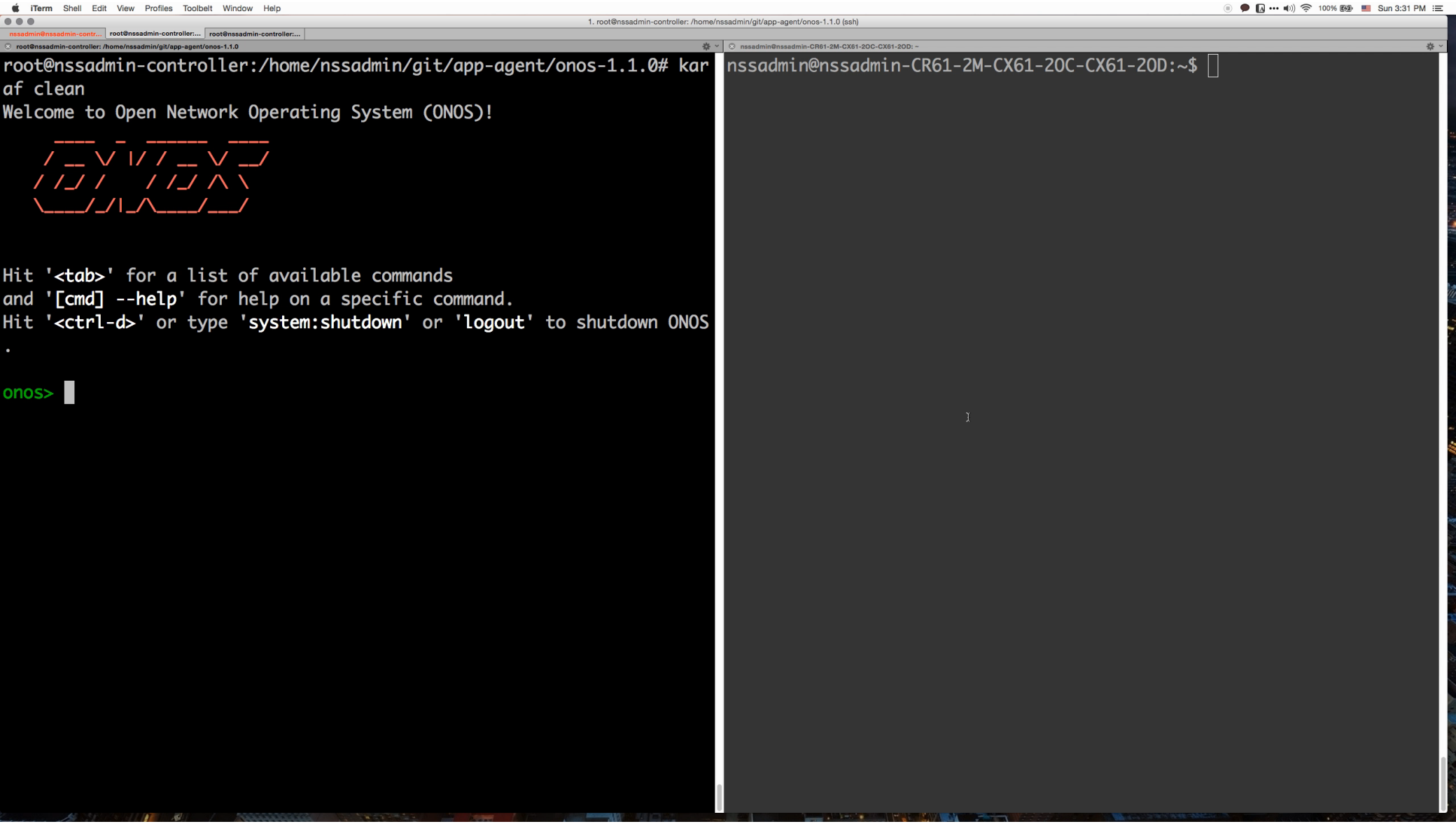
- Scenario



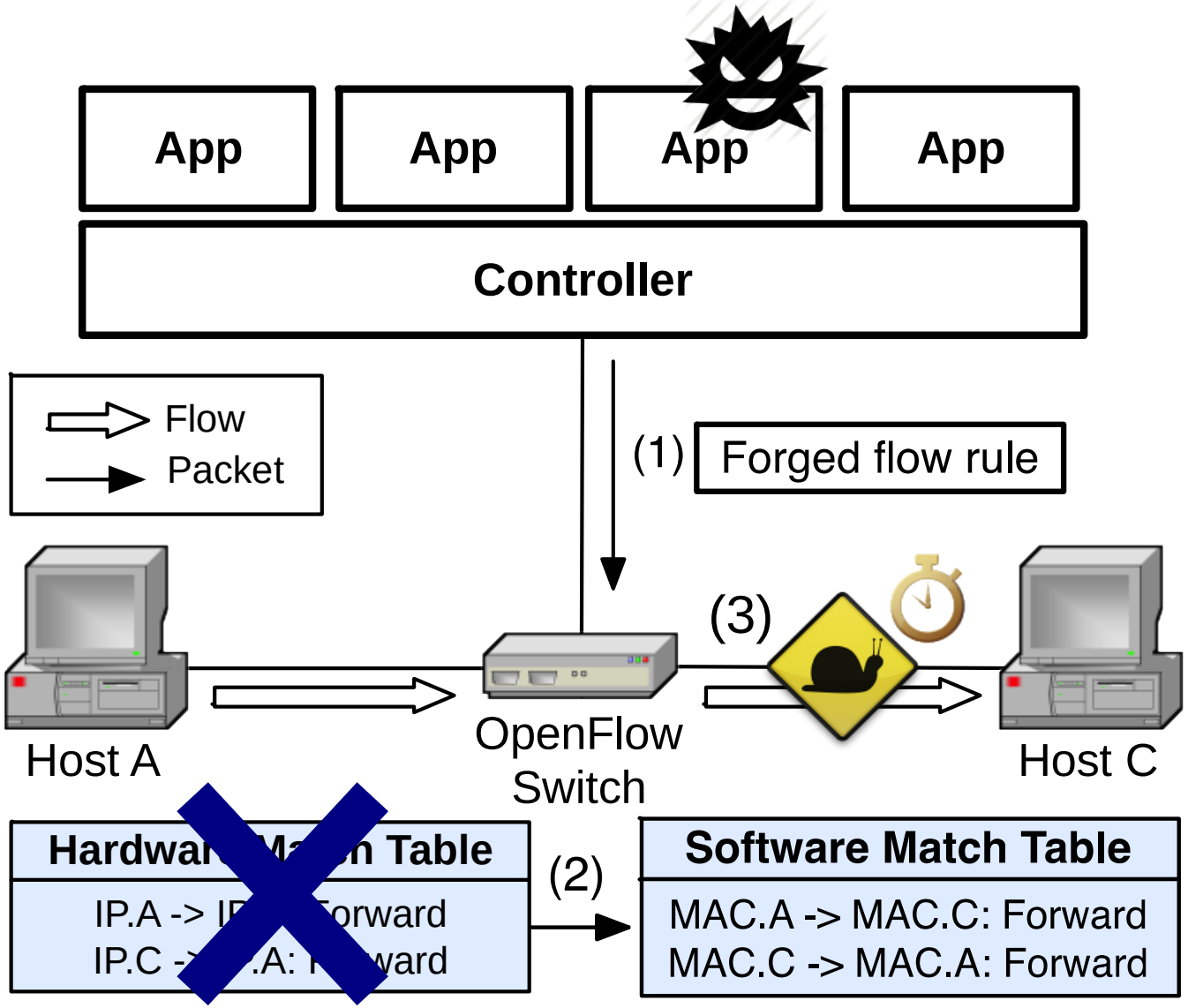
[C-1-A] Flow Rule Flooding



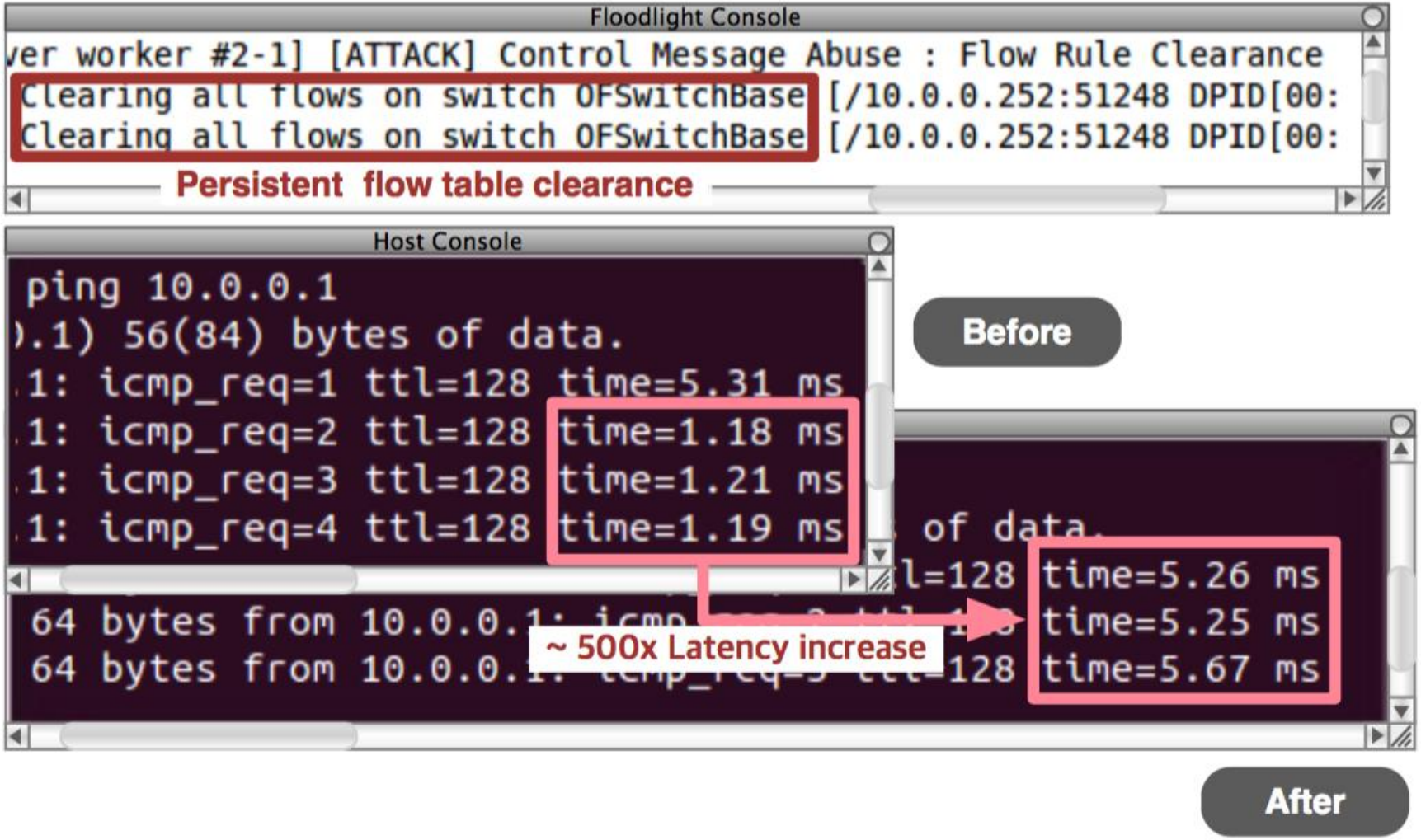
[C-1-A] Flow Rule Flooding



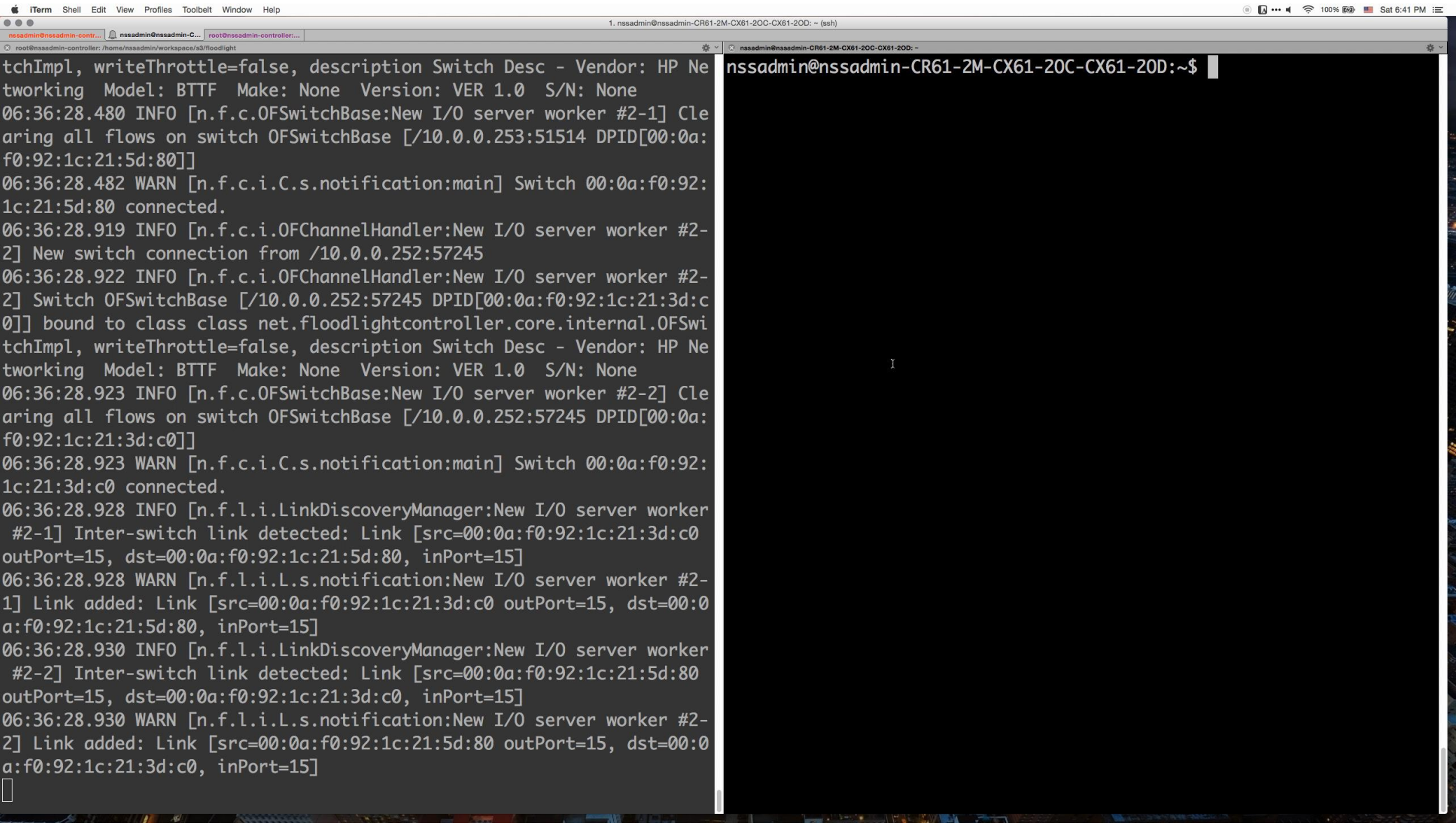
- Scenario


App
App
App
App
Controller
Flow
Packet
(1)
Forged flow rule
(3)
Host A
OpenFlow Switch
Host C
Hardware Match Table
IP.A -> IP.C: Forward
IP.C -> IP.A: Forward
(2)
Software Match Table
MAC.A -> MAC.C: Forward
MAC.C -> MAC.A: Forward

- Degrading network performance

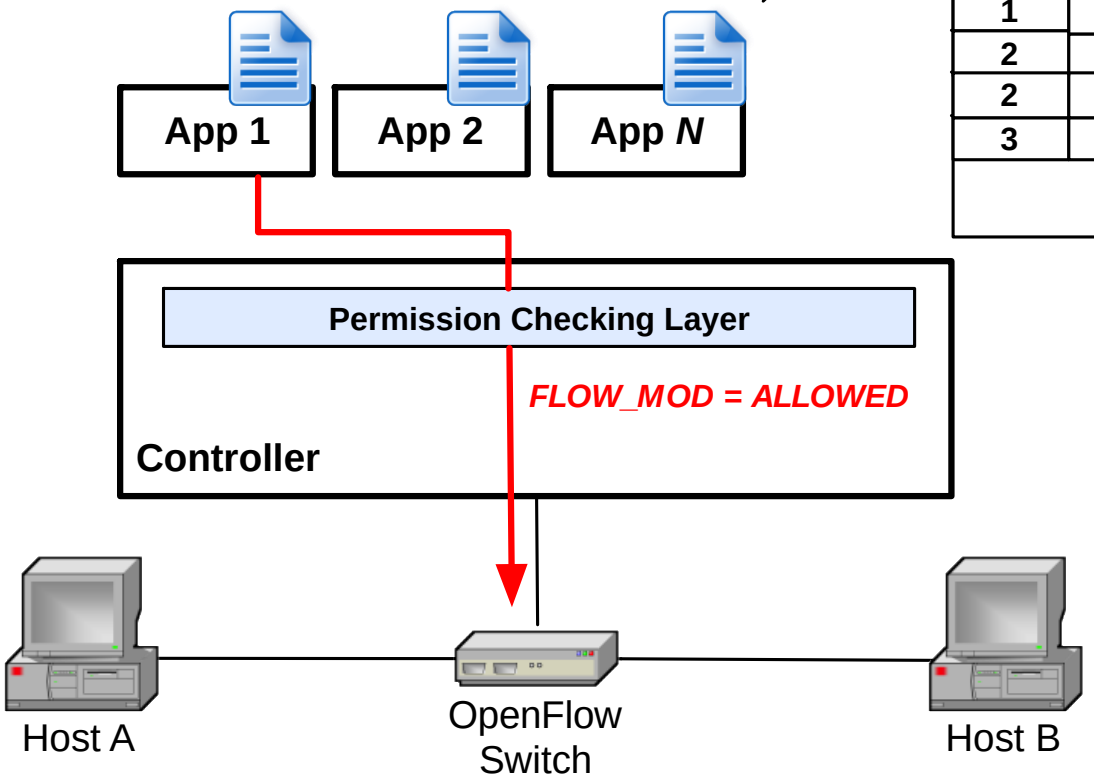


[C-2-M] Switch Firmware Misuse



KAIST

- **Checking List**
  - i) OpenFlow Message
  - ii) System Resource
  - iii) APIs (e.g., SM-ONOS)



Permission File

Type	Permission	Allowed
1	PACKET_OUT	O
1	FLOW_MOD_ADD	O
2	CPU	15 %
2	MEM	20 MB
3	sendMgs()	O
	⋮	

1	*OpenFlow*
2	*System Resource*
3	*APIs*

- **Static/Dynamic Analysis of SDN applications**
  - **Static manner:**
    Control Flow Graph and API call list
  - **Dynamic manner:**
    Sufficient test inputs (e.g., control messages such as PACKEK_IN)

    → Make a decision whether the target application is malicious or n ot.
- Need a definition of malicious behavior for each controller


?
Q
&
&
A
!