SDN Switch Test Examples

June 2016

안종석
NAIM Networks, Inc.

테스트 구성

HP3500 / HP3800
IP: 192.168.99.3

ovs-ofctl

VMware ESXi

VM

Geni4 Host
200.200.200.12

VM

Geni5 Host
200.200.200.13

VM

Geni3 Host
200.200.200.11

Internet
DHCP Server
DNS Server
D/G 192.168.100.1

HP3500 OpenFlow 구성 정보

- **HP3500 CLI 명령어 :** show openflow instance test

```
HP-3500yl-48G-PoEP(config)# show openflow instance test

 Configured OF Version    : 1.0
 Egress Only Ports        : None
 Instance Name            : test
 Data-path Description    : test
 Admin. Status            : Enabled
 Member List              : VLAN 888
 Pipeline Model           : Standard Match
 Listen Port              : 6633
 Oper. Status             : Up
 Oper. Status Reason      : NA
 Datapath ID              : 0378e4115b748fc0
 Mode                     : Active
 Flow Location            : Hardware and Software
 No. of Hw Flows          : 1
 No. of Sw Flows          : 0
 Hw. Rate Limit           : 0 kbps
 Sw. Rate Limit           : 100 pps
 Conn. Interrupt Mode     : Fail-Secure
 Maximum Backoff Interval : 60 seconds
 Probe Interval           : 10 seconds
 Hw. Table Miss Count     : 1542888
 No. of Sw Flow Tables    : NA
 Table Model              : Single Table

 Controller Id Connection Status Connection State Secure Role
 ------------- ----------------- ---------------- ------ ------
 1             Disconnected      Connecting       No     Equal

HP-3500yl-48G-PoEP(config)#
```

HP3500 config 정보

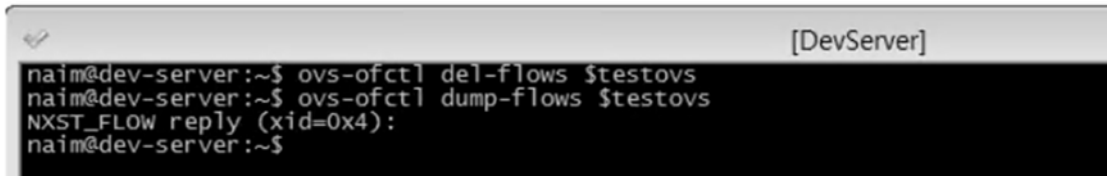
- **HP3500 CLI 명령어 :** show running-config
- **OVS 명령어 :** ovs-ofctl show tcp:192.168.99.3

```
HP-3500yl-48G-PoEP(openflow)# show running-config

Running configuration:

; J9311A Configuration Editor; Created on release #K.15.17.0007
; Ver #08:02.ff.f7.fc.7f.ff.3f.ef:ae
hostname "HP-3500yl-48G-PoEP"
module 1 type j93yya
module 2 type j93xxa
module 3 type j9312a
ip default-gateway 192.168.99.1
snmp-server community "public" unrestricted
openflow
   controller-id 1 ip 192.168.99.100 controller-interface vlan 999
   instance "test"
      listen-port
      member vlan 888
      controller-id 1
      enable
      exit
   enable
   exit
vlan 1
   name "DEFAULT_VLAN"
   no untagged 1-48
   untagged A1-A4
   ip address dhcp-bootp
   exit
vlan 888
   name "VLAN888"
   untagged 37-40
   ip address 192.168.88.3 255.255.255.0
   exit
vlan 999
   name "VLAN999"
   untagged 1-36,41-48
   ip address 192.168.99.3 255.255.255.0
   exit
```

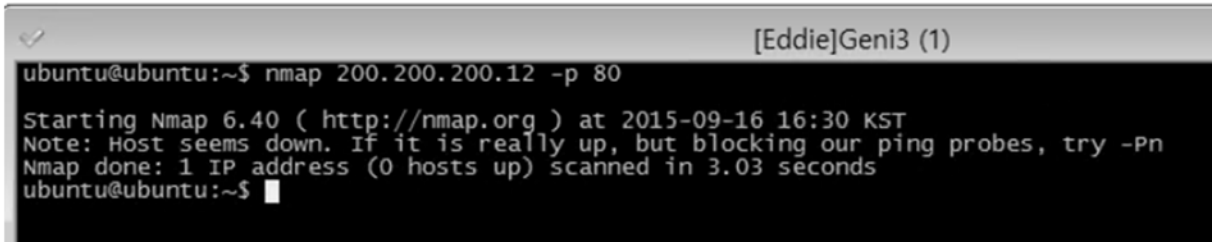
```
mininet@mininet-vm: ~

 * Documentation:  https://help.ubuntu.com/
Last login: Wed Sep 16 14:00:33 2015
mininet@mininet-vm:~$ ovs-ofctl show tcp:192.168.99.3
OFPT_FEATURES_REPLY (xid=0x2): dpid:0378e4115b748fc0
n_tables:16, n_buffers:0
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_N
W_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST
 37(37): addr:e4:11:5b:74:8f:db
     config:     0
     state:      STP_FORWARD
     current:    1GB-FD AUTO_NEG
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD 1GB-FD AUTO_NEG
     speed: 1000 Mbps now, 1000 Mbps max
 38(38): addr:e4:11:5b:74:8f:da
     config:     0
     state:      STP_FORWARD
     current:    1GB-FD AUTO_NEG
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD 1GB-FD AUTO_NEG
     speed: 1000 Mbps now, 1000 Mbps max
 39(39): addr:e4:11:5b:74:8f:d9
     config:     0
     state:      STP_FORWARD
     current:    1GB-FD AUTO_NEG
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD 1GB-FD AUTO_NEG
     speed: 1000 Mbps now, 1000 Mbps max
 40(40): addr:e4:11:5b:74:8f:d8
     config:     0
     state:      STP_FORWARD
     current:    1GB-FD AUTO_NEG
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD 1GB-FD AUTO_NEG
     speed: 1000 Mbps now, 1000 Mbps max
 LOCAL(local): addr:e4:11:5b:74:8f:c0
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
mininet@mininet-vm:~$
```

테스트 단말기 화면 - OF 1.0

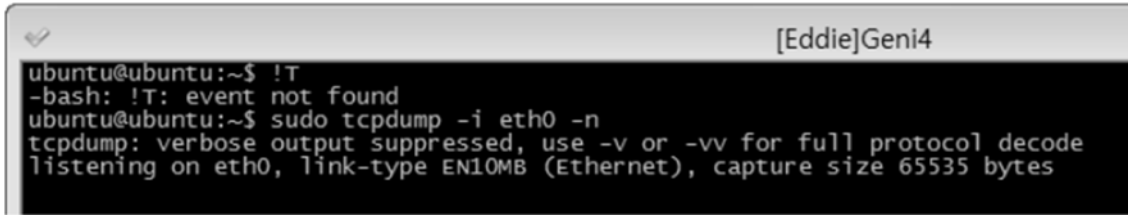
- **ovs-ofctl 명령 실행 창**



- **geni3 = sender = 200.200.200.11**



- **geni4 = receiver= 200.200.200.12**



Rule for ARP - OF 1.0

- **기본적인 arp 통신을 위한 룰을 셋팅함**
  - $testovs = tcp:192.168.99.3:6633
- **룰**
  - ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal

Version Check - OF 1.0

- **Version 확인: 1.0**

```
2015-09-16T05:32:26Z|00023|ofctl|DBG|connecting to tcp:192.168.99.3:6633
2015-09-16T05:32:26Z|00024|poll_loop|DBG|wakeup due to [POLLOUT] on fd 7 (192.168.99.77:49048<->192.168.99.3:6633) at ../lib/stream-fd.c:120
2015-09-16T05:32:26Z|00025|poll_loop|DBG|wakeup due to [POLLOUT] on fd 7 (192.168.99.77:49048<->192.168.99.3:6633) at ../lib/stream-fd.c:120
2015-09-16T05:32:26Z|00026|vconn|DBG|tcp:192.168.99.3:6633: sent (Success): OFPT_HELLO (xid=0x3):
 version bitmap: 0x01
2015-09-16T05:32:26Z|00027|poll_loop|DBG|wakeup due to [POLLIN] on fd 7 (192.168.99.77:49048<->192.168.99.3:6633) at ../lib/stream-fd.c:124
2015-09-16T05:32:26Z|00028|poll_loop|DBG|wakeup due to [POLLIN] on fd 7 (192.168.99.77:49048<->192.168.99.3:6633) at ../lib/stream-fd.c:124
2015-09-16T05:32:26Z|00029|vconn|DBG|tcp:192.168.99.3:6633: received: OFPT_HELLO (xid=0x131):
 version bitmap: 0x01
2015-09-16T05:32:26Z|00030|vconn|DBG|tcp:192.168.99.3:6633: negotiated OpenFlow version 0x01 (we support version 0x01, peer supports version 0x01)
2015-09-16T05:32:26Z|00031|vconn|DBG|tcp:192.168.99.3:6633: sent (Success): OFPT_GET_CONFIG_REQUEST (xid=0x4):
2015-09-16T05:32:26Z|00032|poll_loop|DBG|wakeup due to [POLLIN] on fd 7 (192.168.99.77:49048<->192.168.99.3:6633) at ../lib/stream-fd.c:124
2015-09-16T05:32:26Z|00033|vconn|DBG|tcp:192.168.99.3:6633: received: OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Allow TCP 80 - OF 1.0

Allow TCP 80

```
ovs-ofctl add-flow tcp:192.168.99.3:6633
priority=37000,tcp,nw_src=200.200.200.1,tp_dst=80,actions=normal
ovs-ofctl add-flow tcp:192.168.99.3:6633
priority=37000,tcp,nw_dst=200.200.200.1,tp_src=80,actions=normal
ovs-ofctl dump-flows tcp:192.168.99.3:6633
```

Allow TCP 80 - OF 1.0

- **ovs-ofctl, Geni3, Geni4 사용**

[DevServer]

```
naim@dev-server:~$ ovs-ofctl del-flows $testovs
naim@dev-server:~$ ovs-ofctl dump-flows $testovs
NXST_FLOW reply (xid=0x4):
naim@dev-server:~$
```

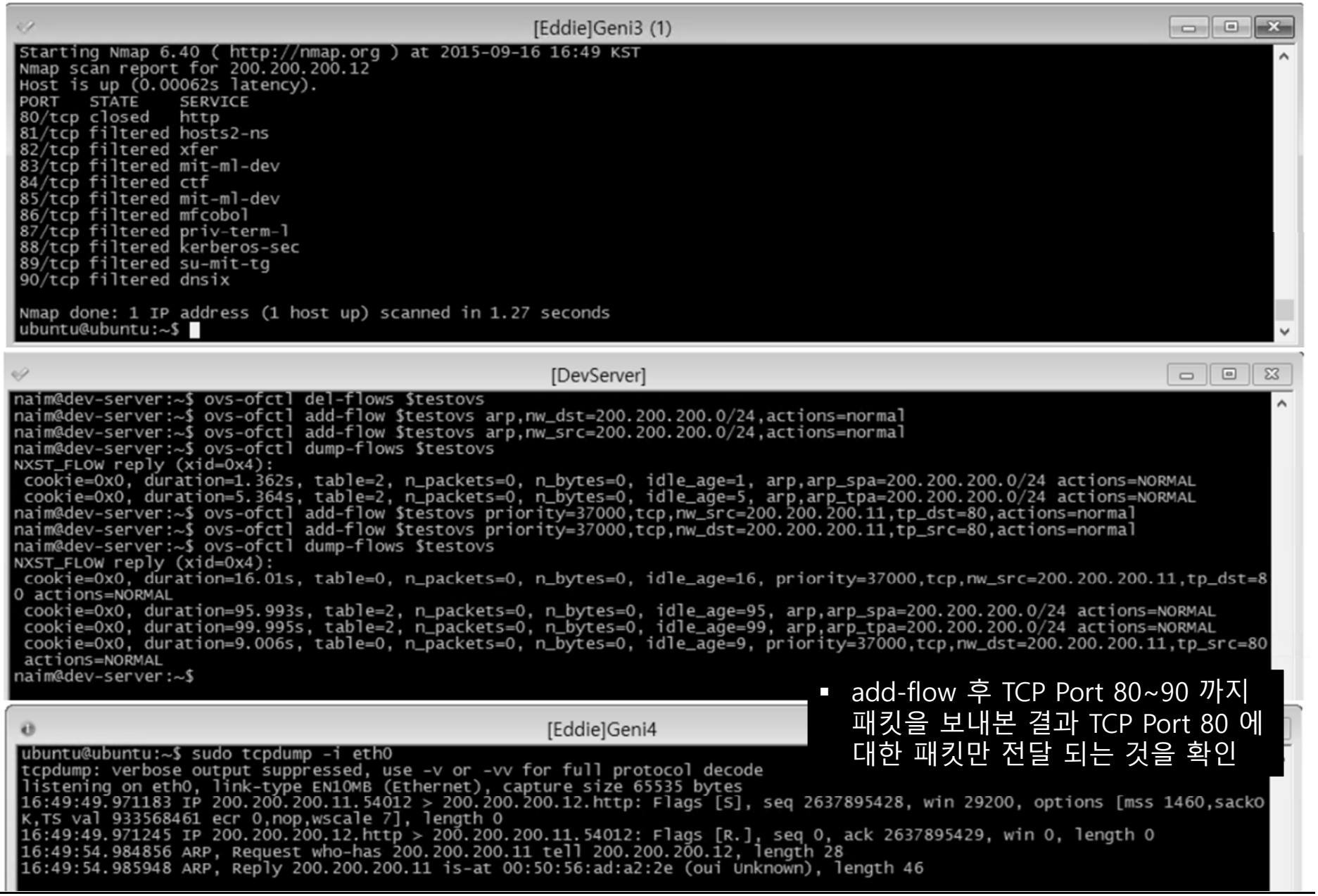
[Eddie]Geni3 (1)

```
ubuntu@ubuntu:~$ nmap 200.200.200.12 -p 80

Starting Nmap 6.40 ( http://nmap.org ) at 2015-09-16 16:30 KST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.03 seconds
ubuntu@ubuntu:~$
```

[Eddie]Geni4

```
ubuntu@ubuntu:~$ !T
-bash: !T: event not found
ubuntu@ubuntu:~$ sudo tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
```

Allow TCP 80 - OF 1.0



- add-flow 후 TCP Port 80~90 까지 패킷을 보내본 결과 TCP Port 80 에 대한 패킷만 전달 되는 것을 확인

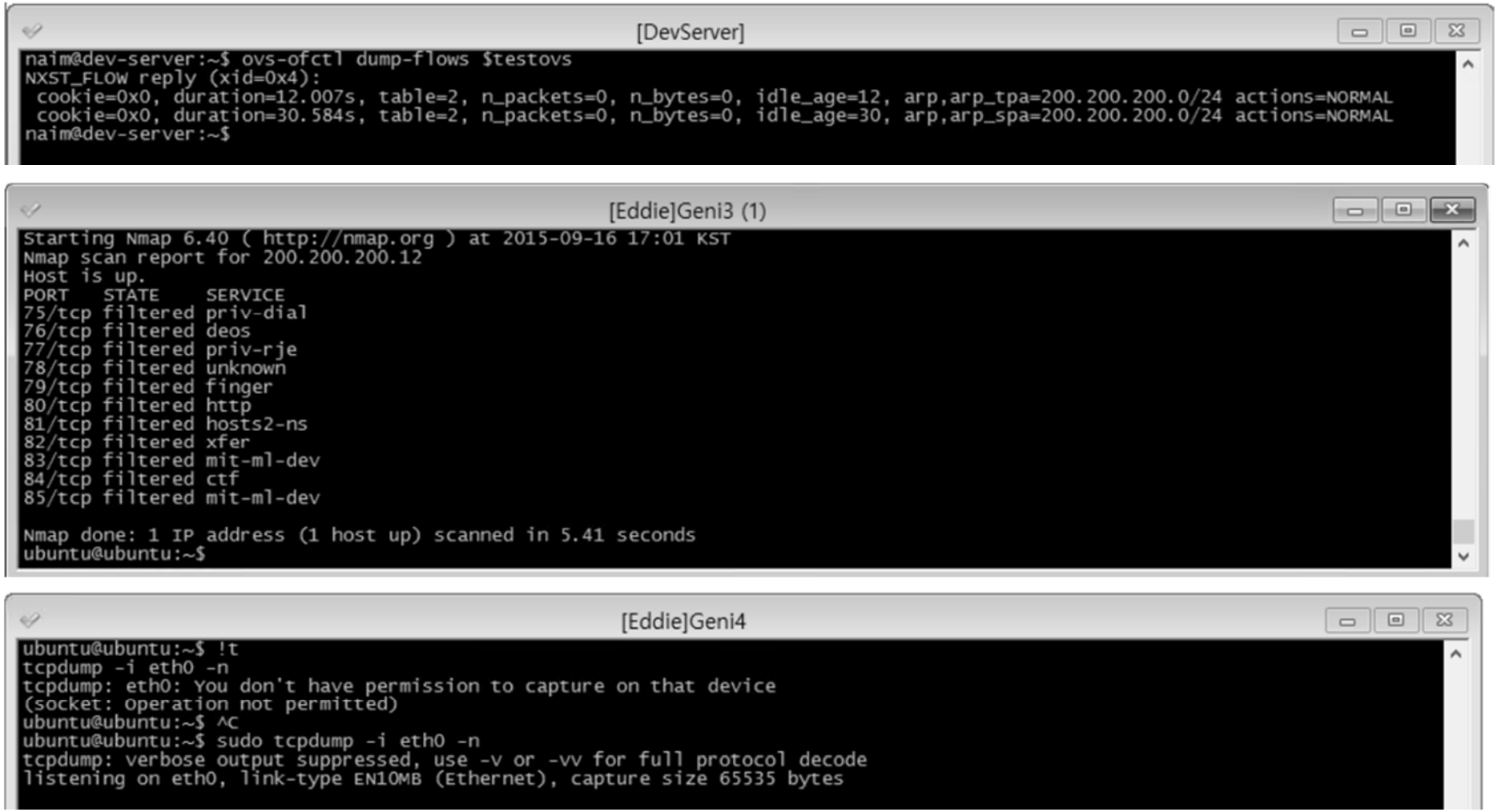
Allow All - OF 1.0

Allow All

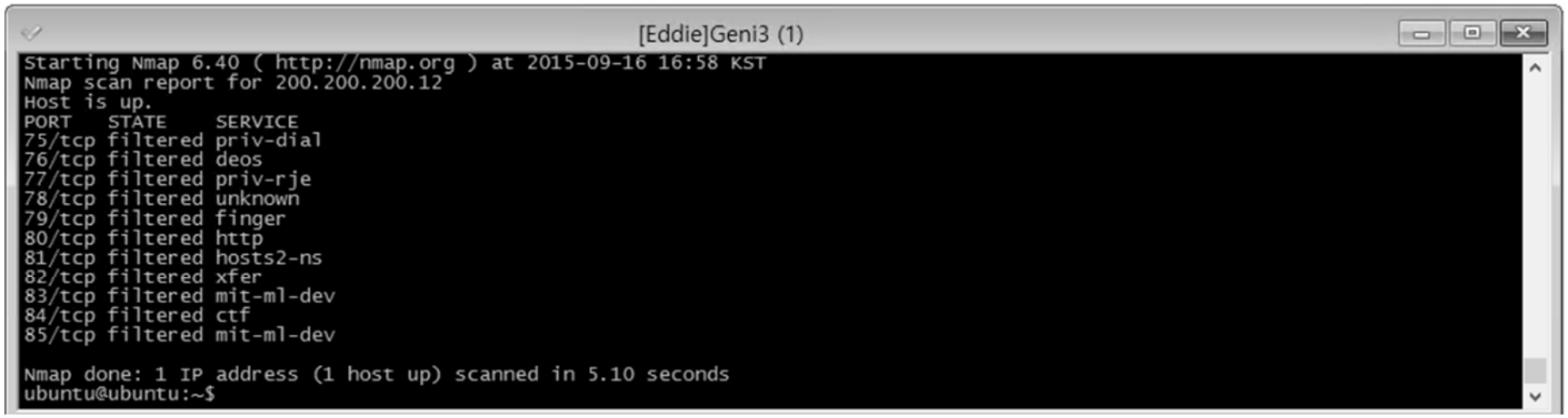
```
ovs-ofctl add-flow tcp:192.168.99.3:6633 ip,nw_src=200.200.200.11,actions=normal
ovs-ofctl add-flow tcp:192.168.99.3:6633 ip,nw_dst=200.200.200.12,actions=normal
ovs-ofctl dump-flows tcp:192.168.99.3:6633
```

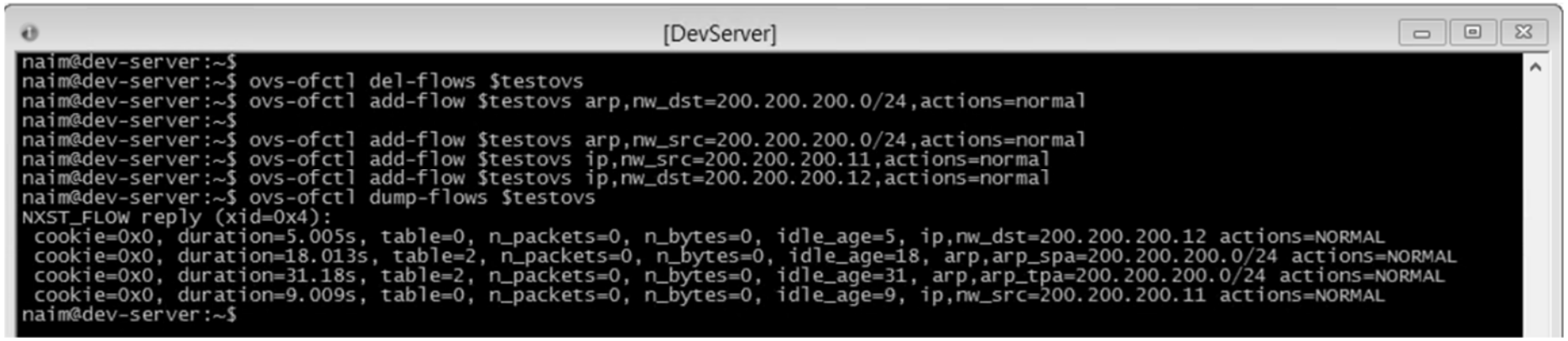
Allow All - OF 1.0

- **ovs-ofctl, Geni3, Geni4 사용**



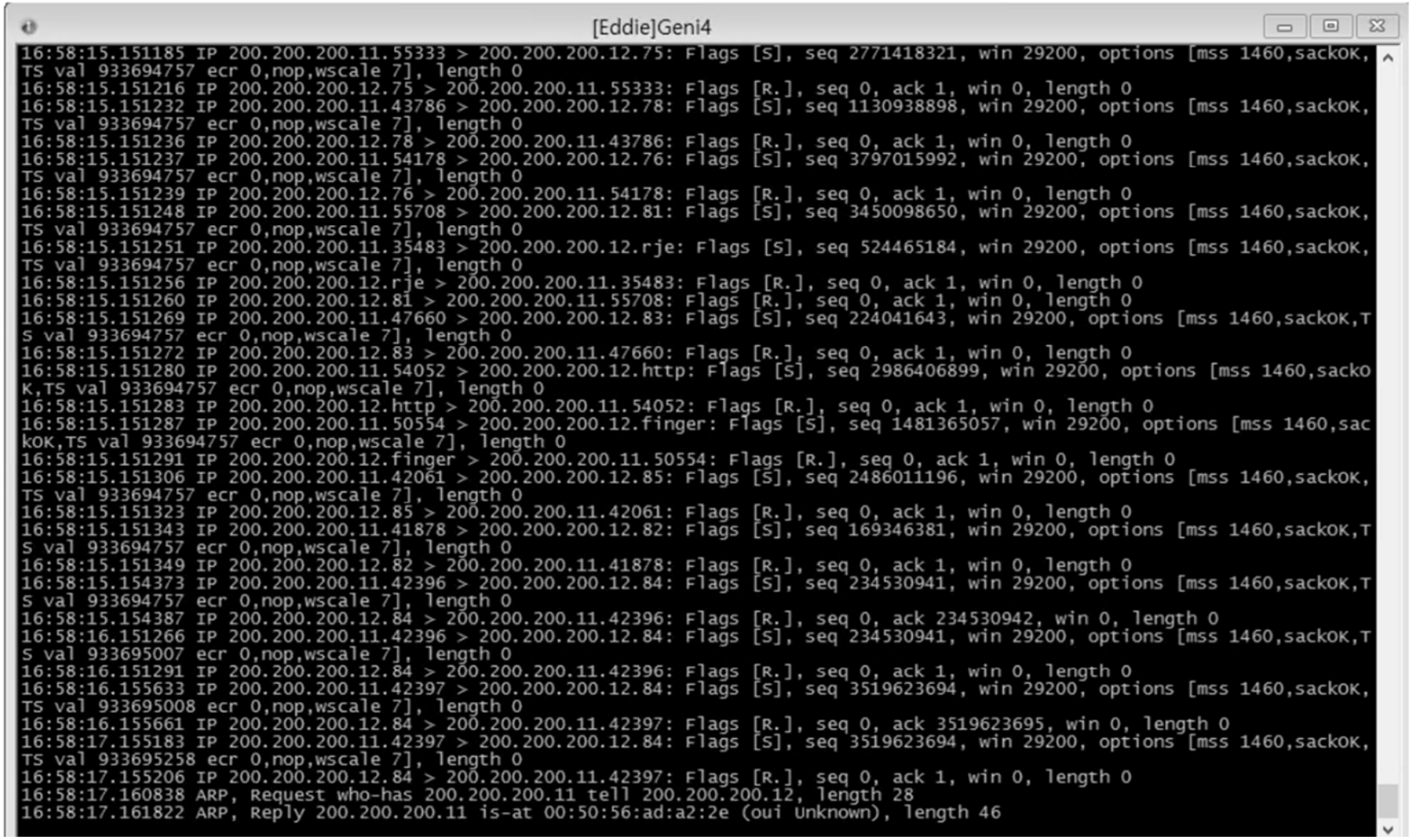
Allow All - OF 1.0





Allow All - OF 1.0

- 200.200.200.11 -> 200.200.200.12의 단방향 패킷만 전달만 됨.
- src = 200.200.200.11, dst = 200.200.200.12 의 패킷에 대한 룰만 들어있음.
- 양방향에 대한 룰을 설정해야 함



Allow All - OF 1.0

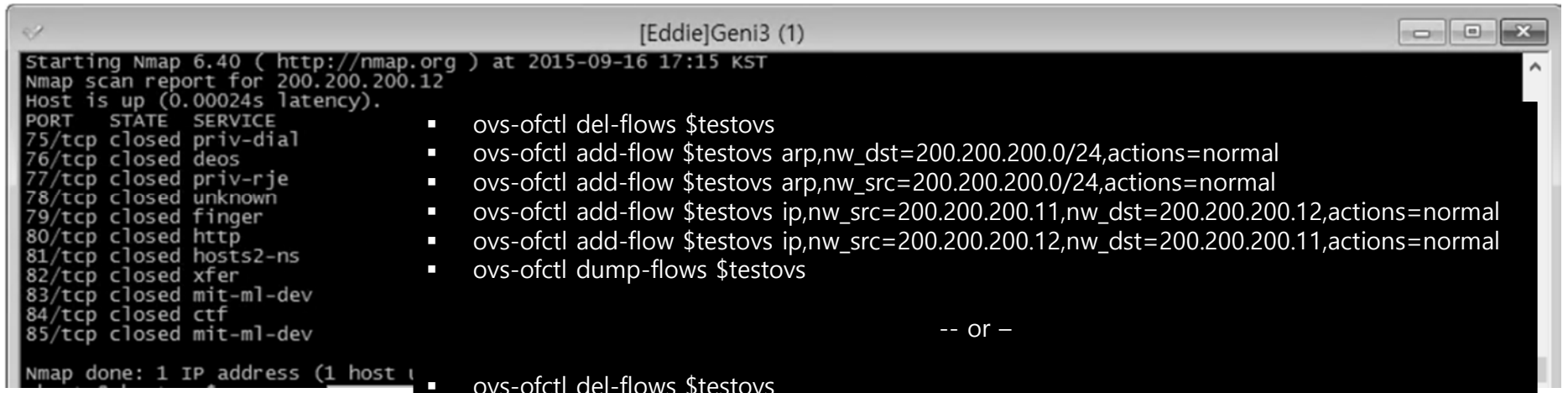
- ovs-ofctl del-flows $testovs
- ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.11,nw_dst=200.200.200.12,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.12,nw_dst=200.200.200.11,actions=normal
- ovs-ofctl dump-flows $testovs

-- or –

- ovs-ofctl del-flows $testovs
- ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.12,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_dst=200.200.200.12,actions=normal
- ovs-ofctl dump-flows $testovs

위의 룰 적용 시 정상적으로 양방향 통신이 가능

Allow All - OF 1.0

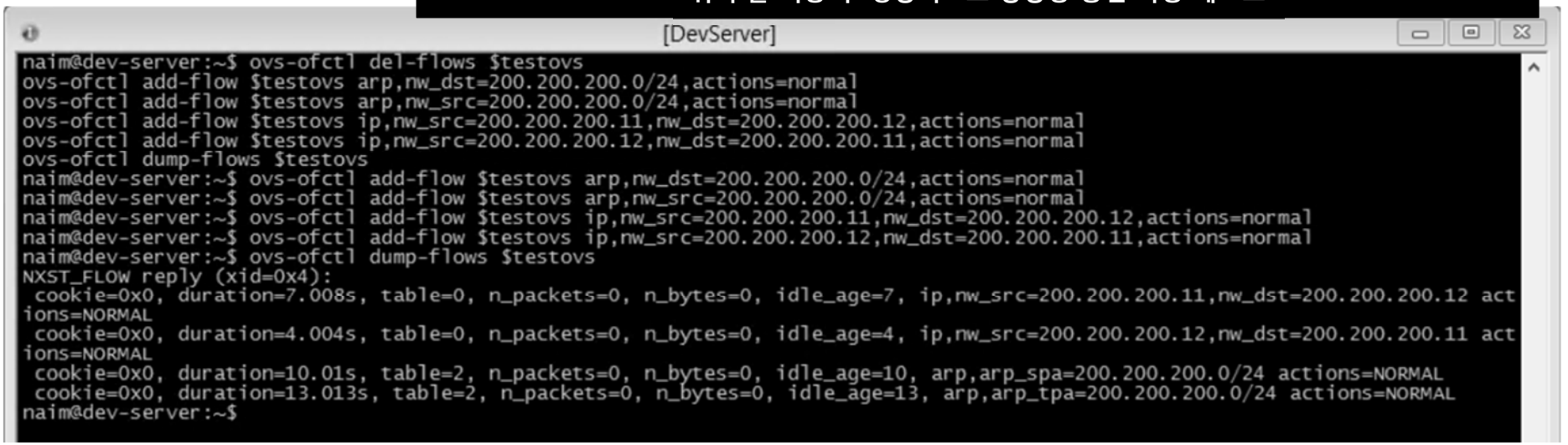


- ovs-ofctl del-flows $testovs
- ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.11,nw_dst=200.200.200.12,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.12,nw_dst=200.200.200.11,actions=normal
- ovs-ofctl dump-flows $testovs

-- or –

- ovs-ofctl del-flows $testovs
- ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.12,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_dst=200.200.200.12,actions=normal
- ovs-ofctl dump-flows $testovs

위의 룰 적용 후 정상적으로 양방향 통신 가능 테스트



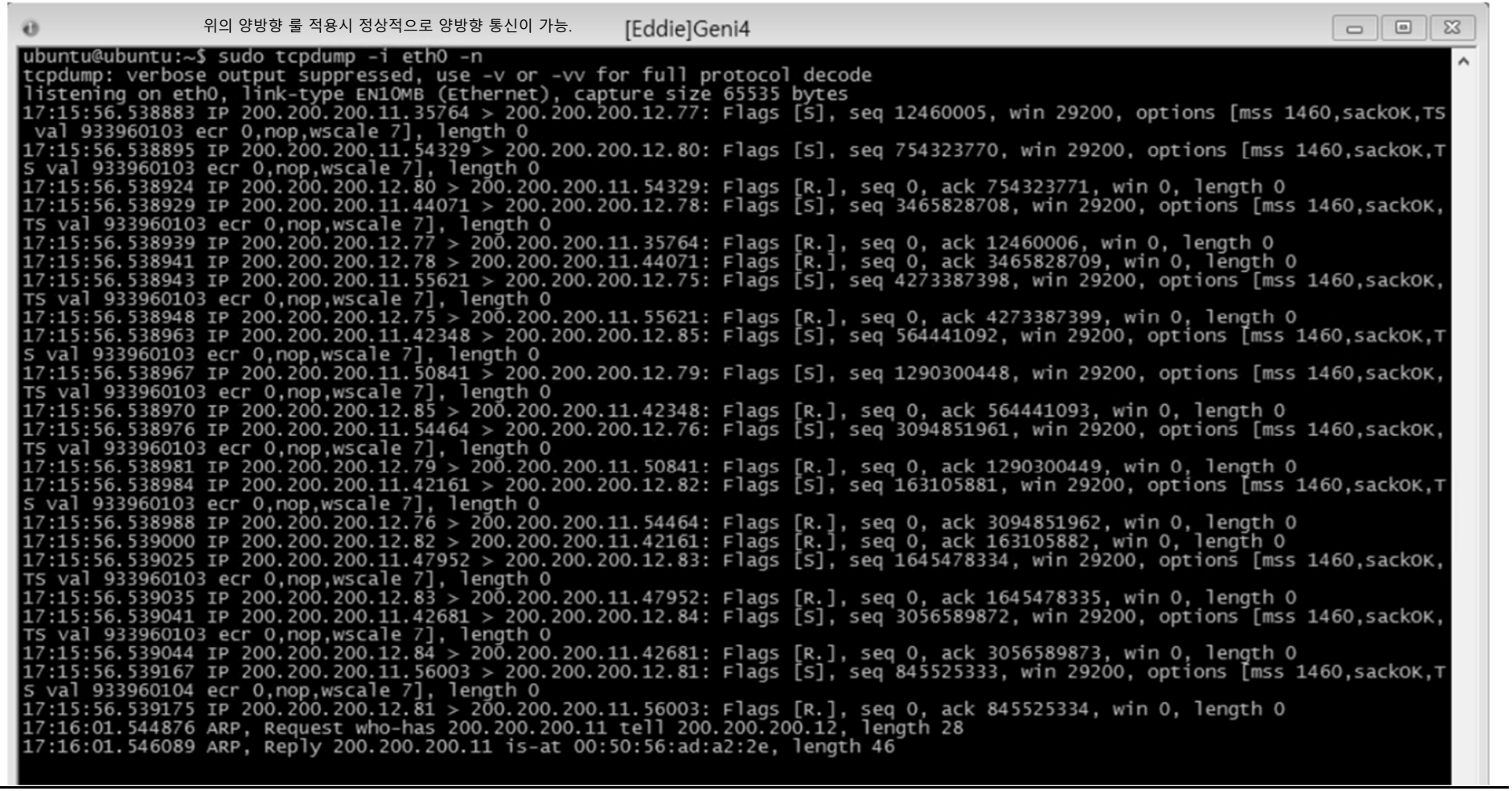
Allow All - OF 1.0

- ovs-ofctl del-flows $testovs
- ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.11,nw_dst=200.200.200.12,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.12,nw_dst=200.200.200.11,actions=normal
- ovs-ofctl dump-flows $testovs

-- or --

- ovs-ofctl del-flows $testovs
- ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_src=200.200.200.12,actions=normal
- ovs-ofctl add-flow $testovs ip,nw_dst=200.200.200.12,actions=normal
- ovs-ofctl dump-flows $testovs

위의 양방향 룰 적용시 정상적으로 양방향 통신이 가능.



ARP - OF 1.0

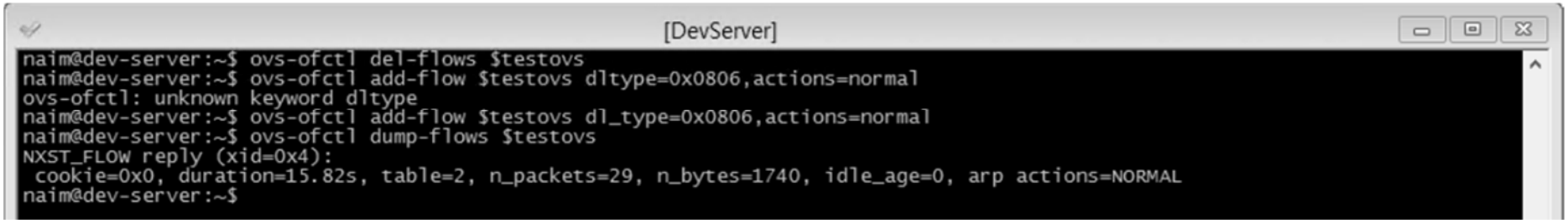
ARP

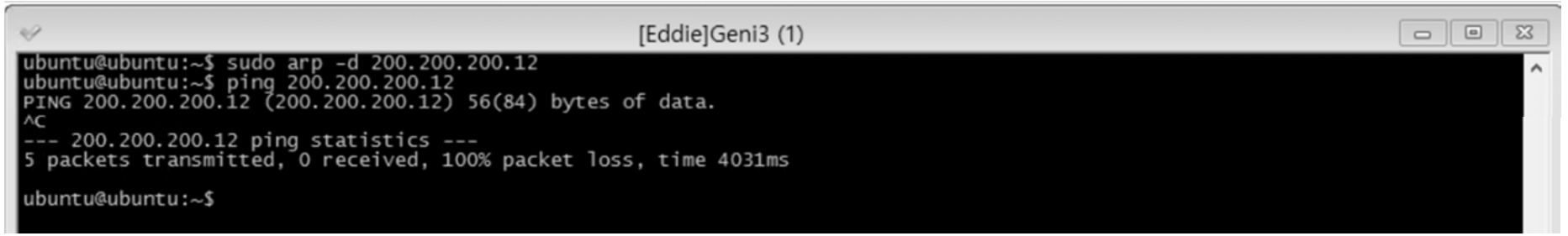
```
ovs-ofctl add-flow tcp:192.168.99.3:6633 dl_type=0x0806,actions=normal
ovs-ofctl dump-flows tcp:192.168.99.3:6633
```

ARP - OF 1.0

- ovs-ofctl add-flow $testovs dltype=0x0806,actions=normal
- 위 룰 적용 결과 sender에서 ping을 전송 시 arp 만 전송되고 그 후 icmp 패킷은 전달되지 않음.

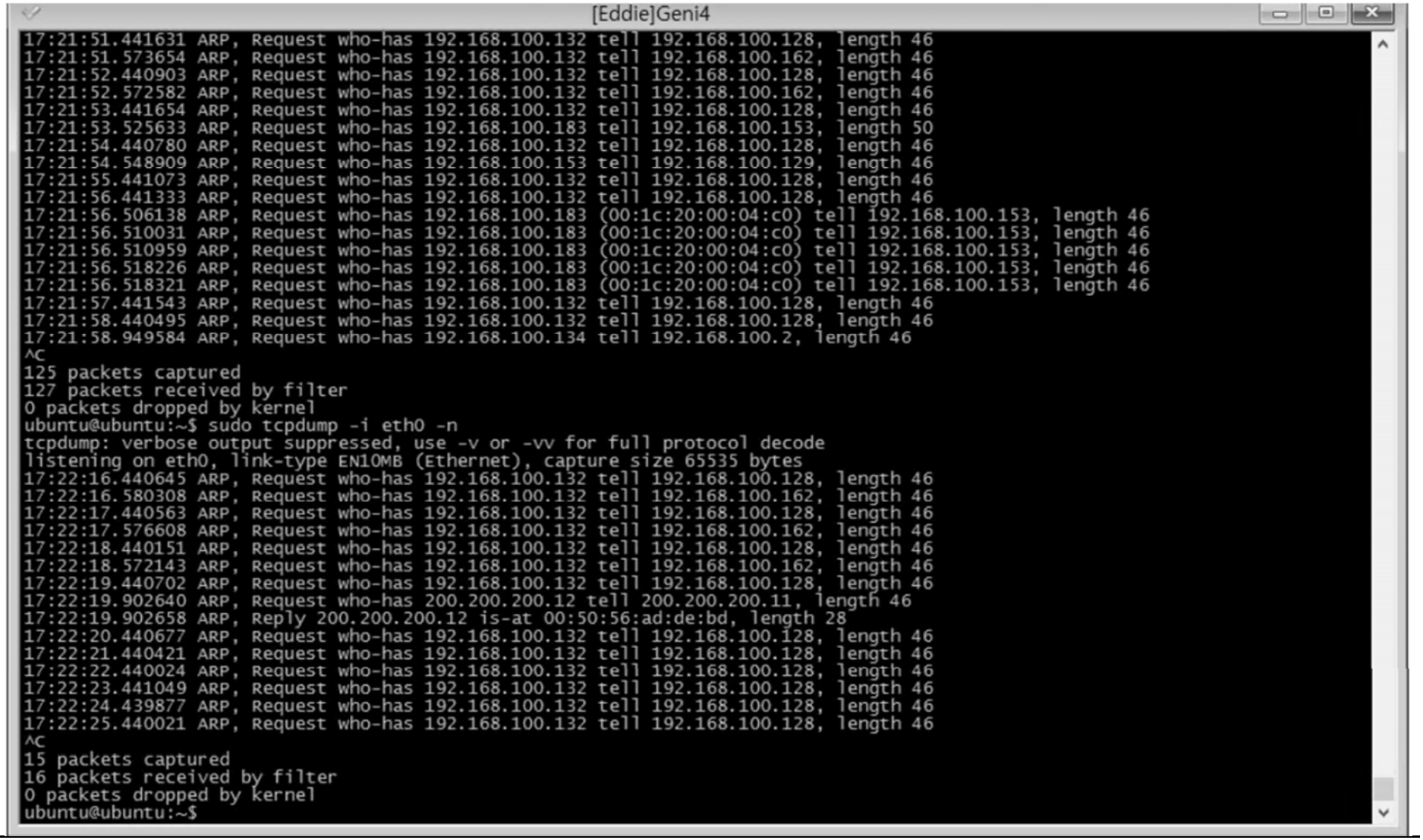
- ovs-ofctl add-flow $testovs arp,actions=normal
- 위 룰도 같은 결과를 볼 수 있다.





ARP - OF 1.0

- **ARP 테스트 결과:** ping을 전송 시 arp 만 전송되고 그 후 icmp 패킷은 전달되지 않음.



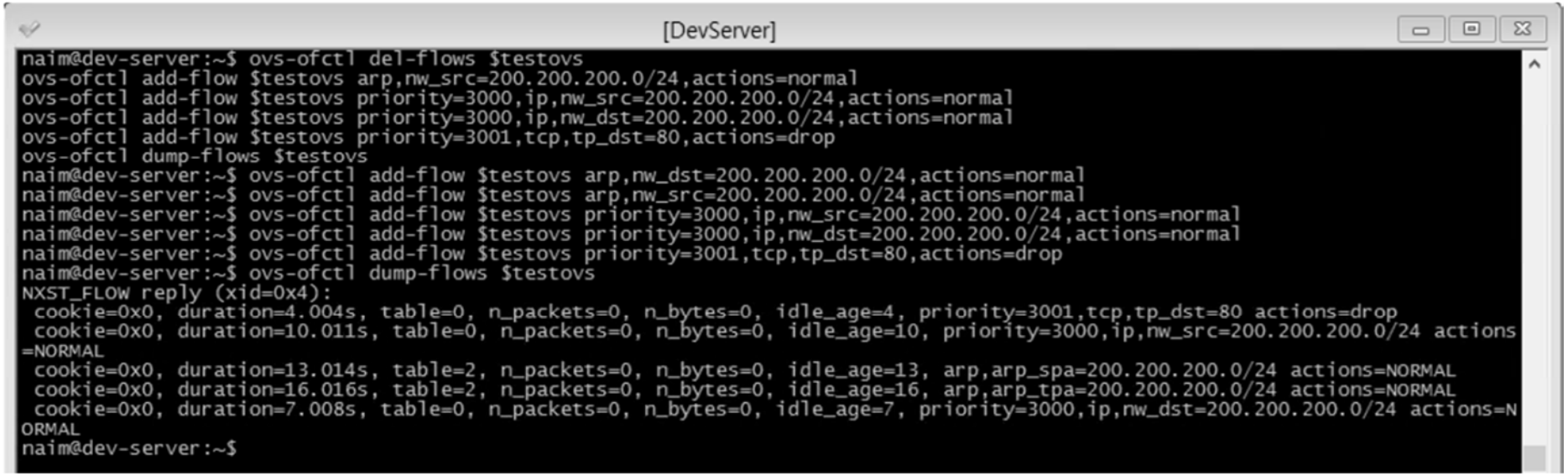
Block TCP 80

Block TCP 80

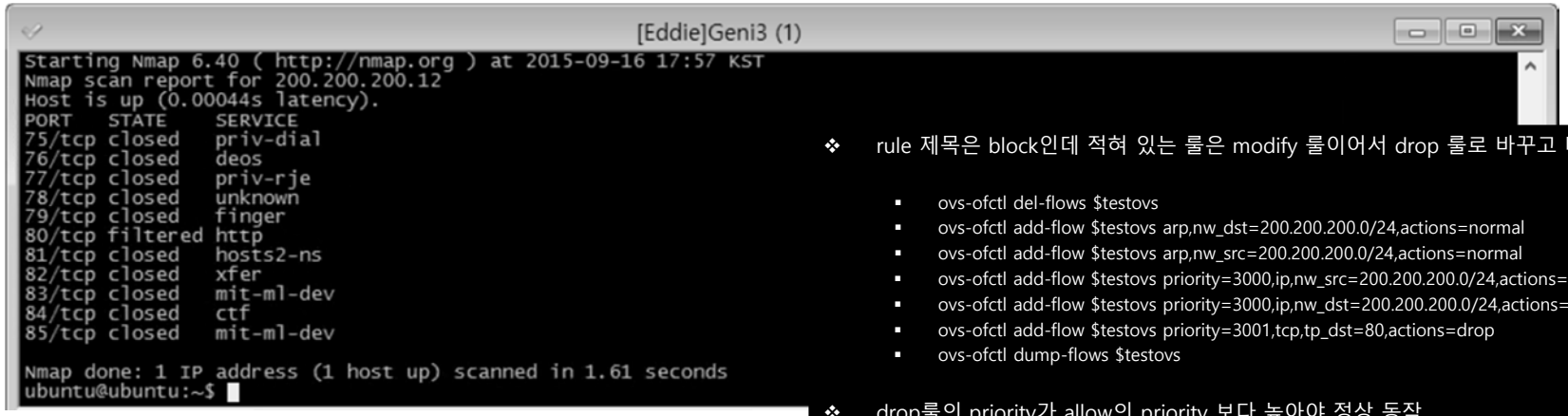
```
#ovs-ofctl add-flow tcp:192.168.99.3:6633 tcp,nw_src=200.200.200.0,tp_dst=80,actions=TABLE
#ovs-ofctl add-flow tcp:192.168.99.3:6633 tcp,nw_src=200.200.200.0/24,tp_dst=80,actions=mod
_nw_dst:200.200.200.12,mod_dl_dst:d8:50:e6:57:b8:ba
#ovs-ofctl add-flow tcp:192.168.99.3:6633 priority=45000,tcp,tp_dst=80,actions=mod_nw_dst:2
00.200.200.12,mod_dl_dst:ac:22:0b:85:5b:63,output:17
#ovs-ofctl add-flow tcp:192.168.99.3:6633 priority=45000,tcp,tp_dst=80,actions=mod_dl_dst:cc:
cc:cc:cc:cc:cc,output:17

ovs-ofctl add-flow tcp:192.168.99.3:6633 "tcp,tp_dst=80,actions=mod_dl_dst:ac:22:0b:85:5b:63"

#ovs-ofctl add-group tcp:192.168.99.3:6633 group_id=1,type=indirect,bucket=mod_dl_dst:33:33
:33:33:33:33

#ovs-ofctl add-flow tcp:192.168.99.3:6633 priority=45000,tcp,tp_dst=80,actions=output:17
#ovs-ofctl add-flow tcp:192.168.99.3:6633 priority=45000,tcp,in_port=17,actions=NORMAL
#ovs-ofctl add-flow tcp:192.168.99.3:6633 tcp,nw_src=200.200.200.0,tp_dst=80,actions=LOCAL
ovs-ofctl dump-flows tcp:192.168.99.3:6633
```

Block TCP 80 - OF 1.0

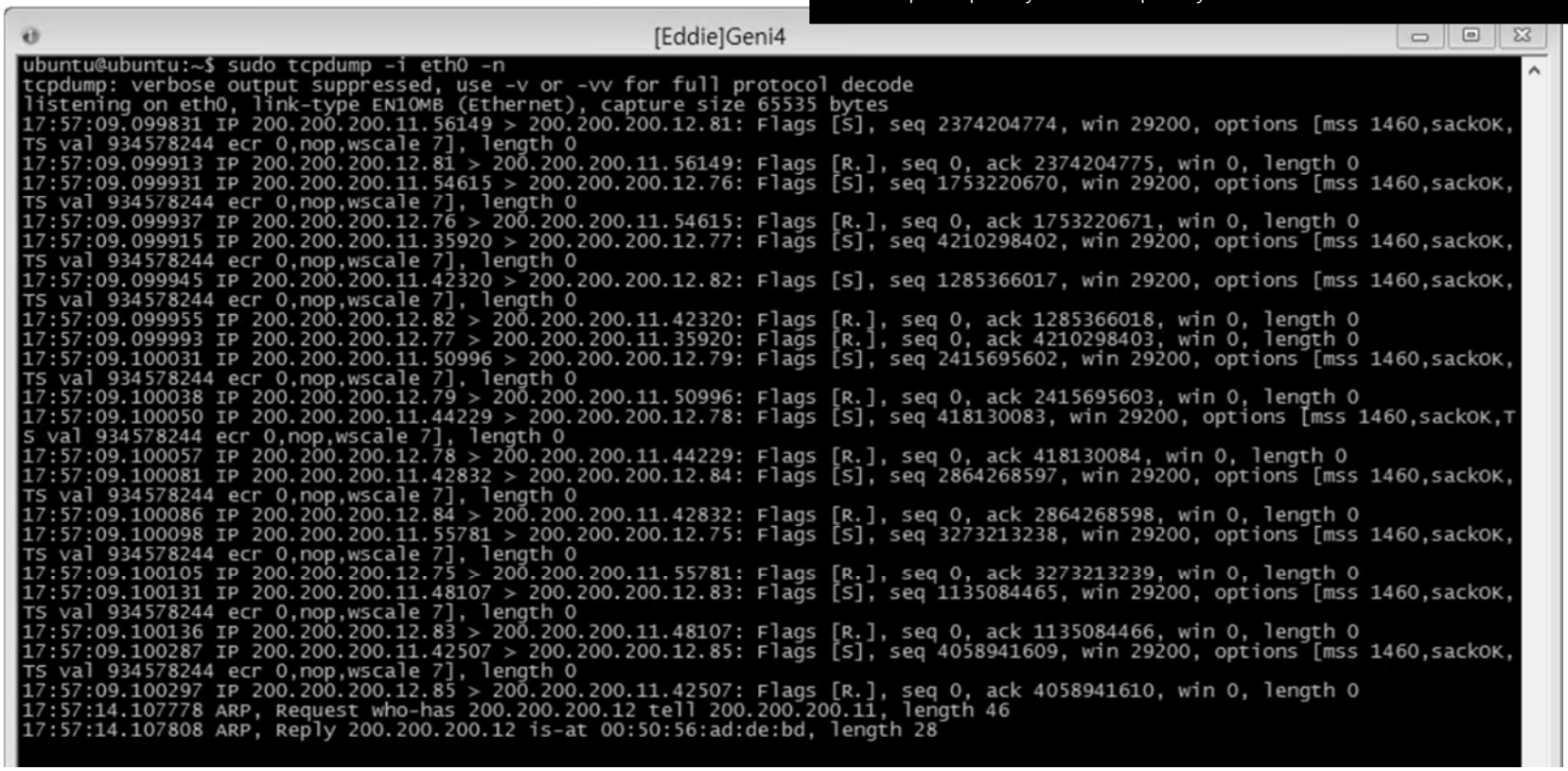
- **제목이 block인데 룰 내용은 modify 이어서 drop 으로 변경 테스트**
  - ovs-ofctl del-flows $testovs
  - ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs priority=3000,ip,nw_src=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs priority=3000,ip,nw_dst=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs priority=3001,tcp,tp_dst=80,actions=drop
  - ovs-ofctl dump-flows $testovs
- **drop룰의 priority가 allow의 priority 보다 높아야 정상 동작.**



Block TCP 80 - OF 1.0



- rule 제목은 block인데 적혀 있는 룰은 modify 룰이어서 drop 룰로 바꾸고 테스트함.
  - ovs-ofctl del-flows $testovs
  - ovs-ofctl add-flow $testovs arp,nw_dst=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs arp,nw_src=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs priority=3000,ip,nw_src=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs priority=3000,ip,nw_dst=200.200.200.0/24,actions=normal
  - ovs-ofctl add-flow $testovs priority=3001,tcp,tp_dst=80,actions=drop
  - ovs-ofctl dump-flows $testovs
- drop룰의 priority가 allow의 priority 보다 높아야 정상 동작.



Block Basic - OF 1.0

Block Basic

```
#ovs-ofctl add-flow tcp:192.168.99.3:6633 ip,nw_src=200.200.200.0/24,actio
ns=drop
ovs-ofctl add-flow tcp:192.168.99.3:6633 priority=30000,dl_type=0x0800,ac
tions=drop
ovs-ofctl dump-flows tcp:192.168.99.3:6633
```

Block Basic - OF 1.0

- arp 패킷은 전달 되나, ip 프로토콜에 속하는 icmp 패킷은 전달 되지 않음.
- 패킷을 전달하는 룰보다 priority가 높아야함.

[DevServer]

```
naim@dev-server:~$ ovs-ofctl del-flows $testovs
ovs-ofctl add-flow $testovs priority=29999,actions=normal
ovs-ofctl add-flow $testovs priority=30000,dl_type=0x0800,actions=drop
ovs-ofctl dump-flows $testovs
naim@dev-server:~$ ovs-ofctl add-flow $testovs priority=29999,actions=normal
naim@dev-server:~$ ovs-ofctl add-flow $testovs priority=30000,dl_type=0x0800,actions=drop
naim@dev-server:~$ ovs-ofctl dump-flows $testovs
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=4.004s, table=0, n_packets=0, n_bytes=0, idle_age=4, priority=30000,ip actions=drop
 cookie=0x0, duration=7.007s, table=0, n_packets=1, n_bytes=0, idle_age=7, priority=29999 actions=NORMAL
naim@dev-server:~$
```

[Eddie]Geni3 (1)

```
ubuntu@ubuntu:~$ arp -d 200.200.200.12
SIOCDARP(dontpub): Operation not permitted
ubuntu@ubuntu:~$ sudo arp -d 200.200.200.12
ubuntu@ubuntu:~$ ping 200.200.200.12
PING 200.200.200.12 (200.200.200.12) 56(84) bytes of data.
^C
--- 200.200.200.12 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4032ms

ubuntu@ubuntu:~$ sudo arp -d 200.200.200.12
ubuntu@ubuntu:~$ ping 200.200.200.12
PING 200.200.200.12 (200.200.200.12) 56(84) bytes of data.
^C
--- 200.200.200.12 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3024ms

ubuntu@ubuntu:~$
```

[Eddie]Geni4

```
ubuntu@ubuntu:~$ sudo tcpdump -i eth0 src host 200.200.200.11
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
18:07:07.675258 ARP, Request who-has 200.200.200.12 tell 200.200.200.11, length 46
18:07:22.493171 ARP, Request who-has 200.200.200.12 tell 200.200.200.11, length 46
18:07:29.208781 ARP, Request who-has 200.200.200.12 tell 200.200.200.11, length 46
```

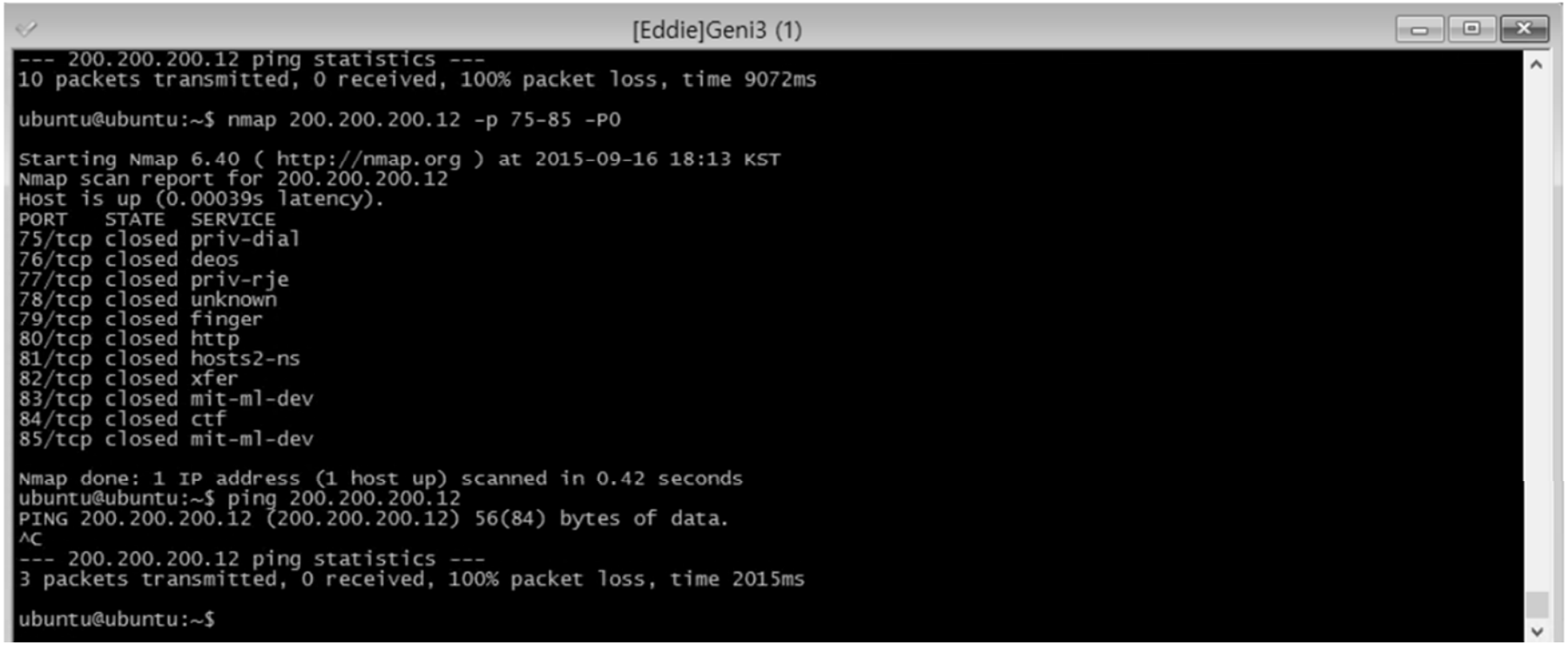
Block ICMP - OF 1.0

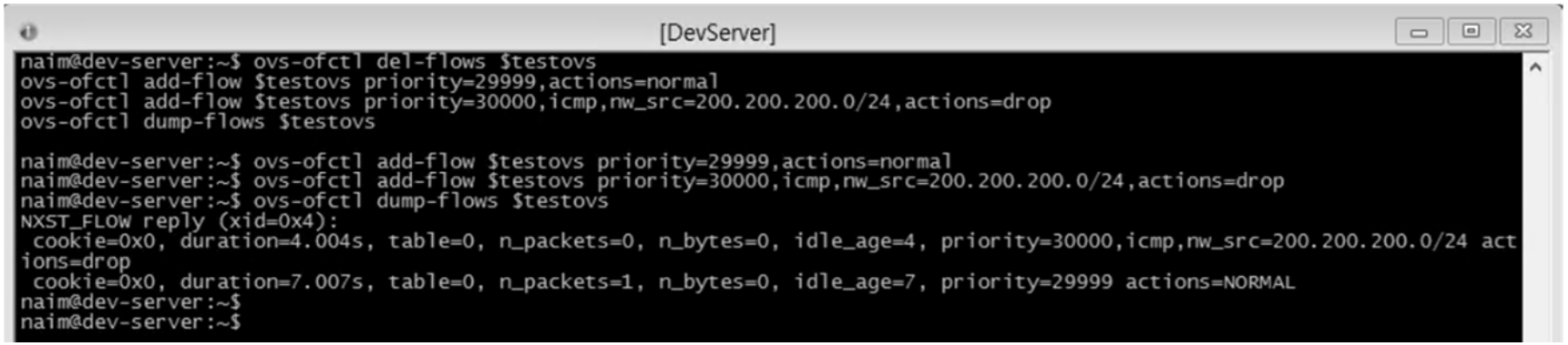
Block ICMP

```
ovs-ofctl add-flow tcp:192.168.99.3:6633 icmp,nw_src=200.200.200.0/24,actions=drop
ovs-ofctl dump-flows tcp:192.168.99.3:6633
```

Block ICMP - OF 1.0

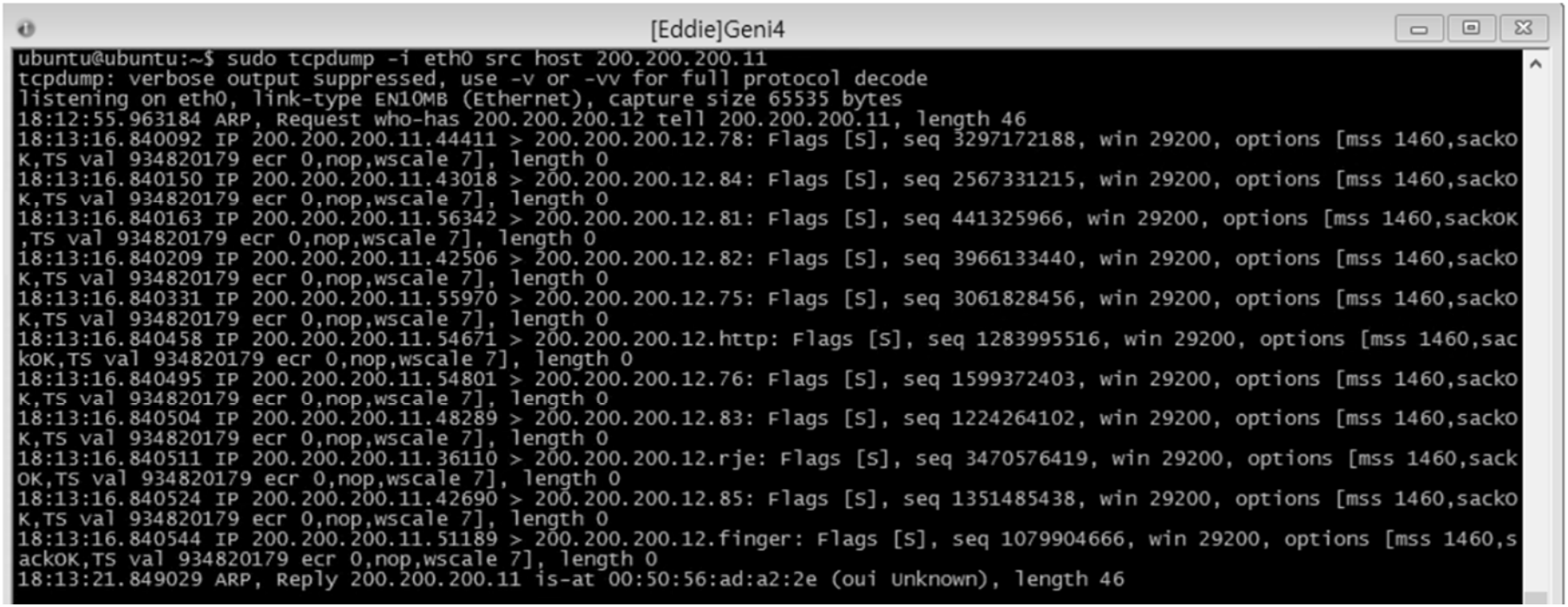
- arp, tcp의 다른패킷은 전달되나 icmp패킷은 전달되지 않음.





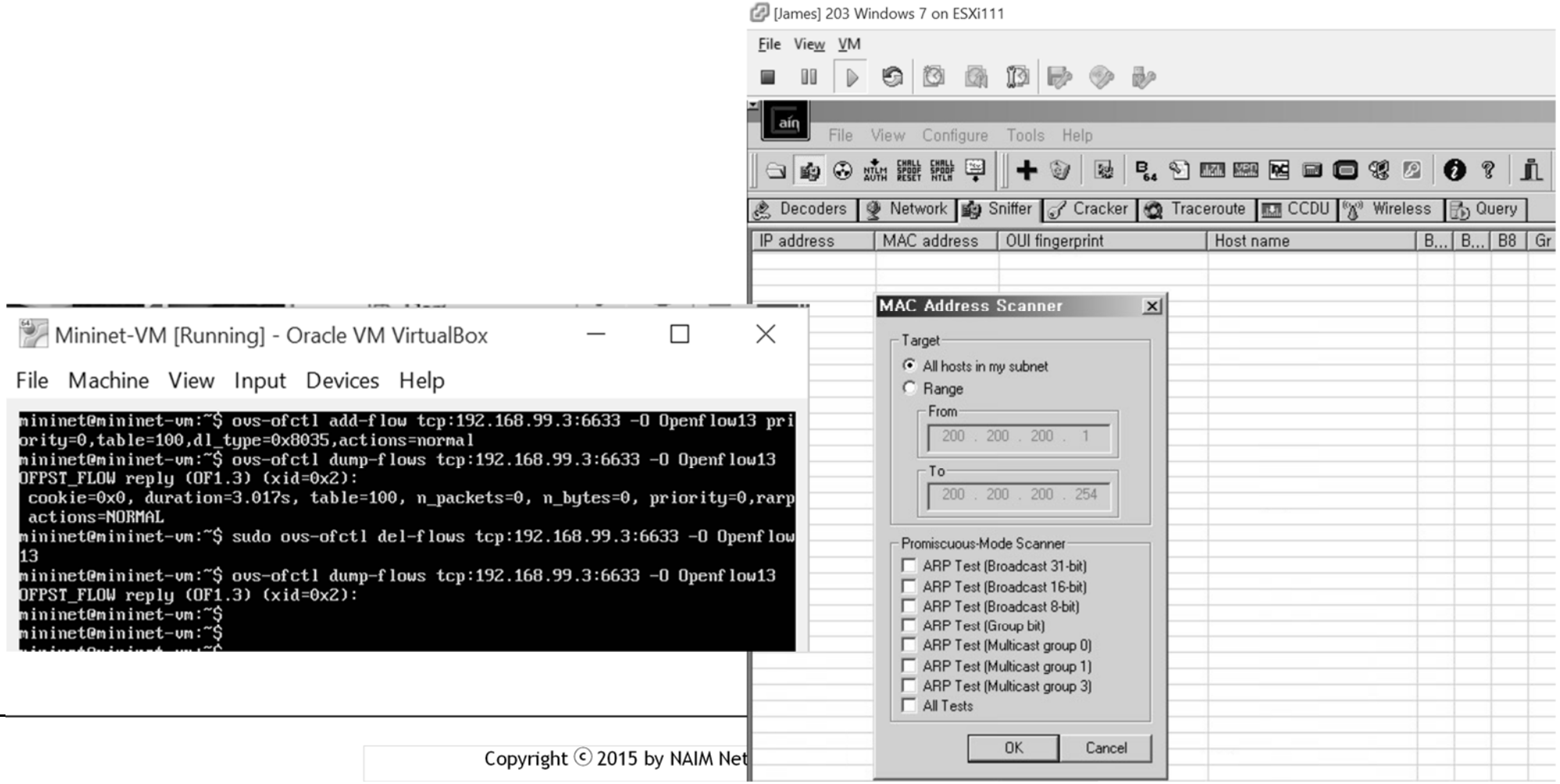
Block ICMP - OF 1.0

- arp, tcp의 다른패킷은 전달되나 icmp패킷은 전달되지 않음.



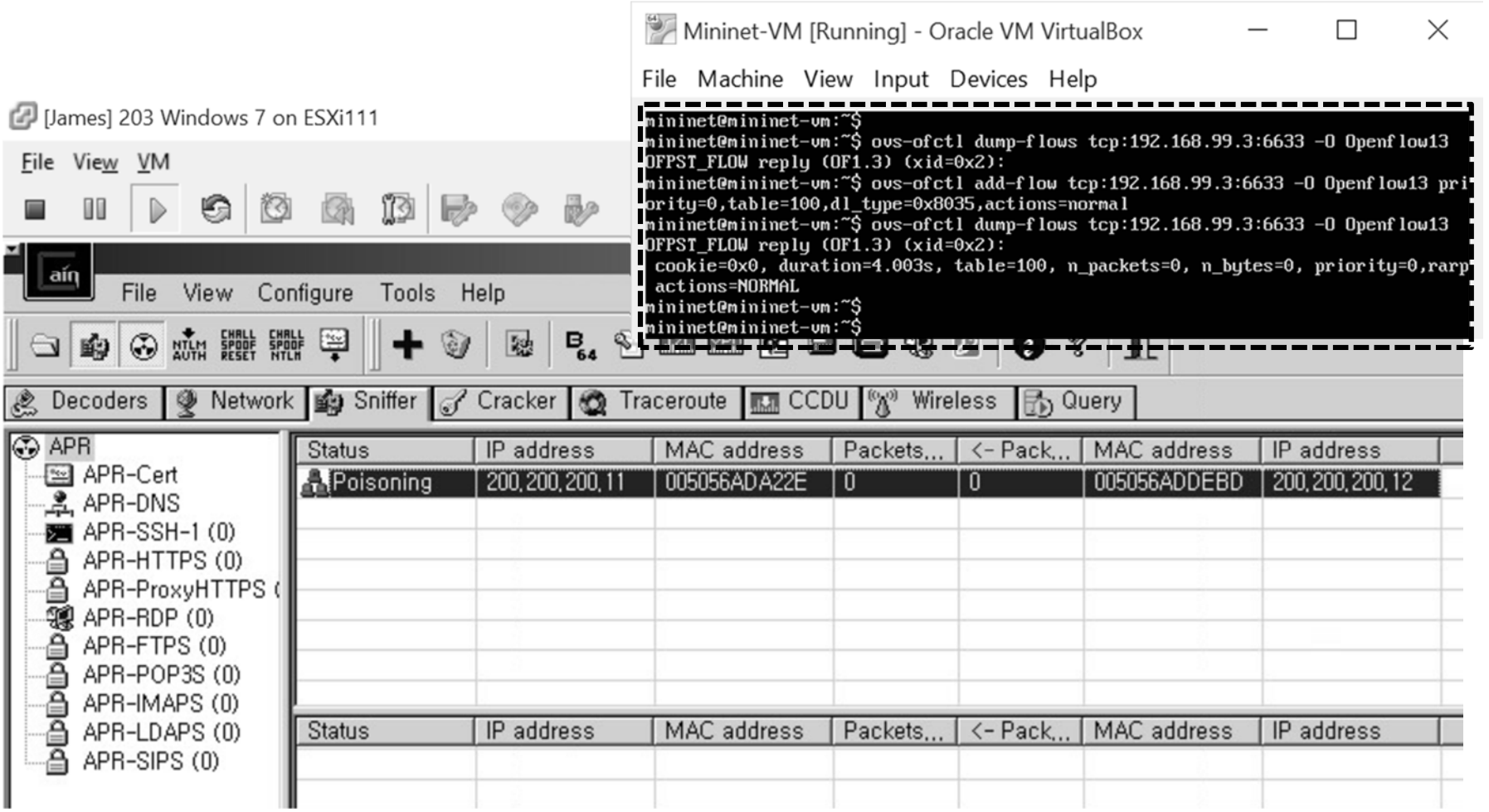
RARP

- **RARP 허용하지 않으면 Security Tool (Cain&Abel) 동작 하지 않음**
  - **./ovs-ofctl dump-flows tcp:192.168.99.3:6633 -O Openflow13**



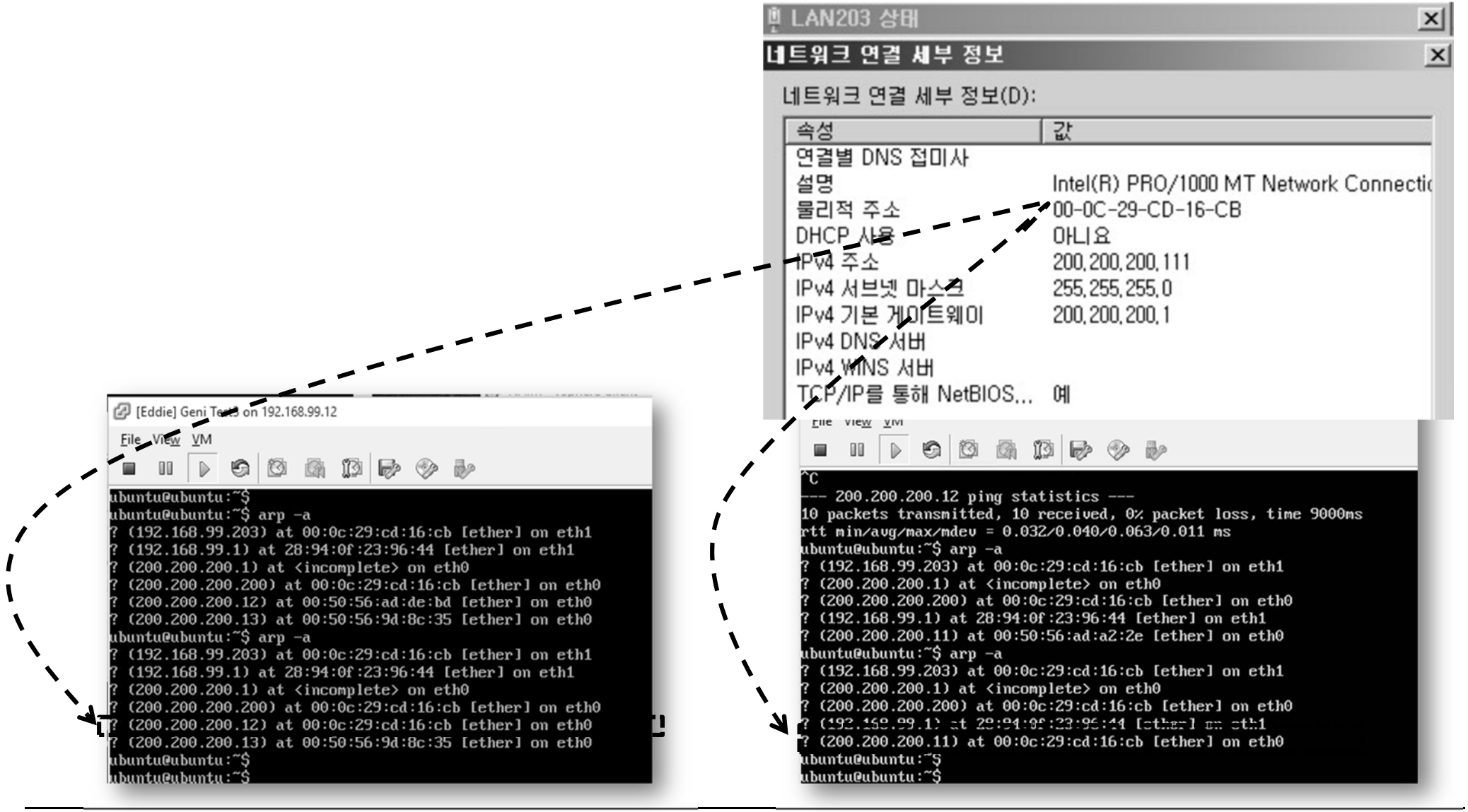
RARP

- **RARP 허용하면 Security Tool (Cain&Abel) 동작함**
  - **./ovs-ofctl dump-flows tcp:192.168.99.3:6633 -O Openflow13**



RARP

- **RARP 허용하면 Security Tool (Cain&Abel)에 의해 ARP 변경 가능**



Thank you very much