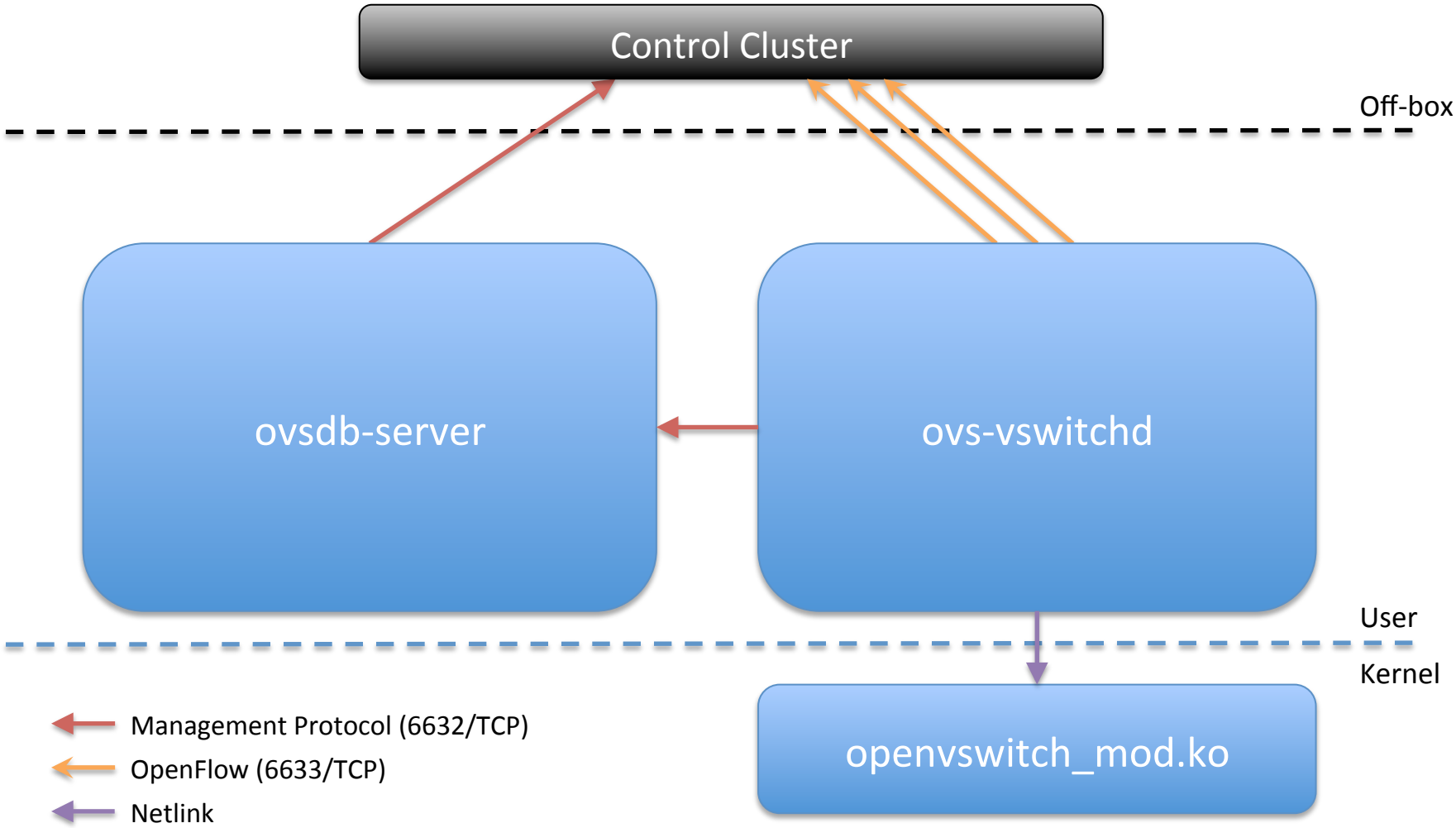
# Debugging OVS

Justin Pettit

April 14, 2011

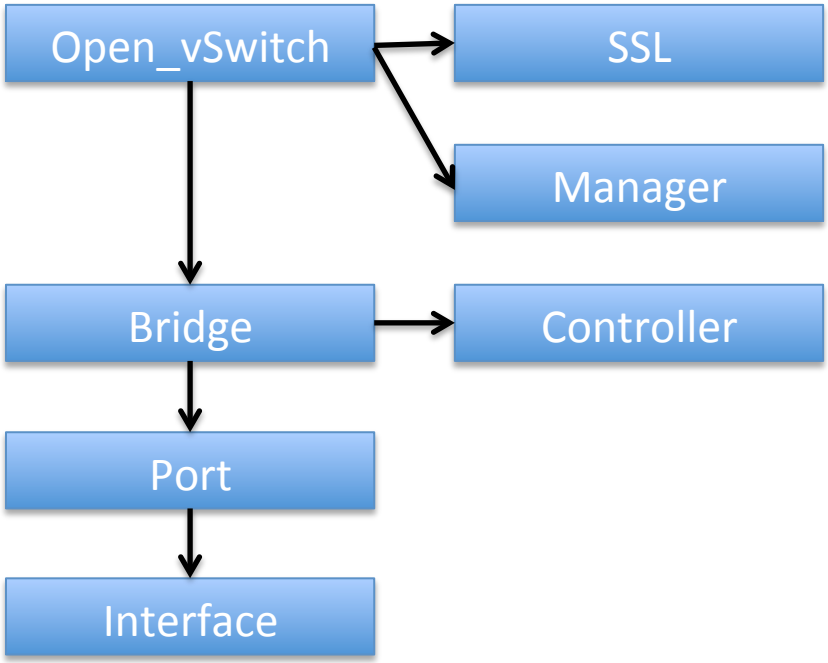
# Main Components



# Debugging the Database

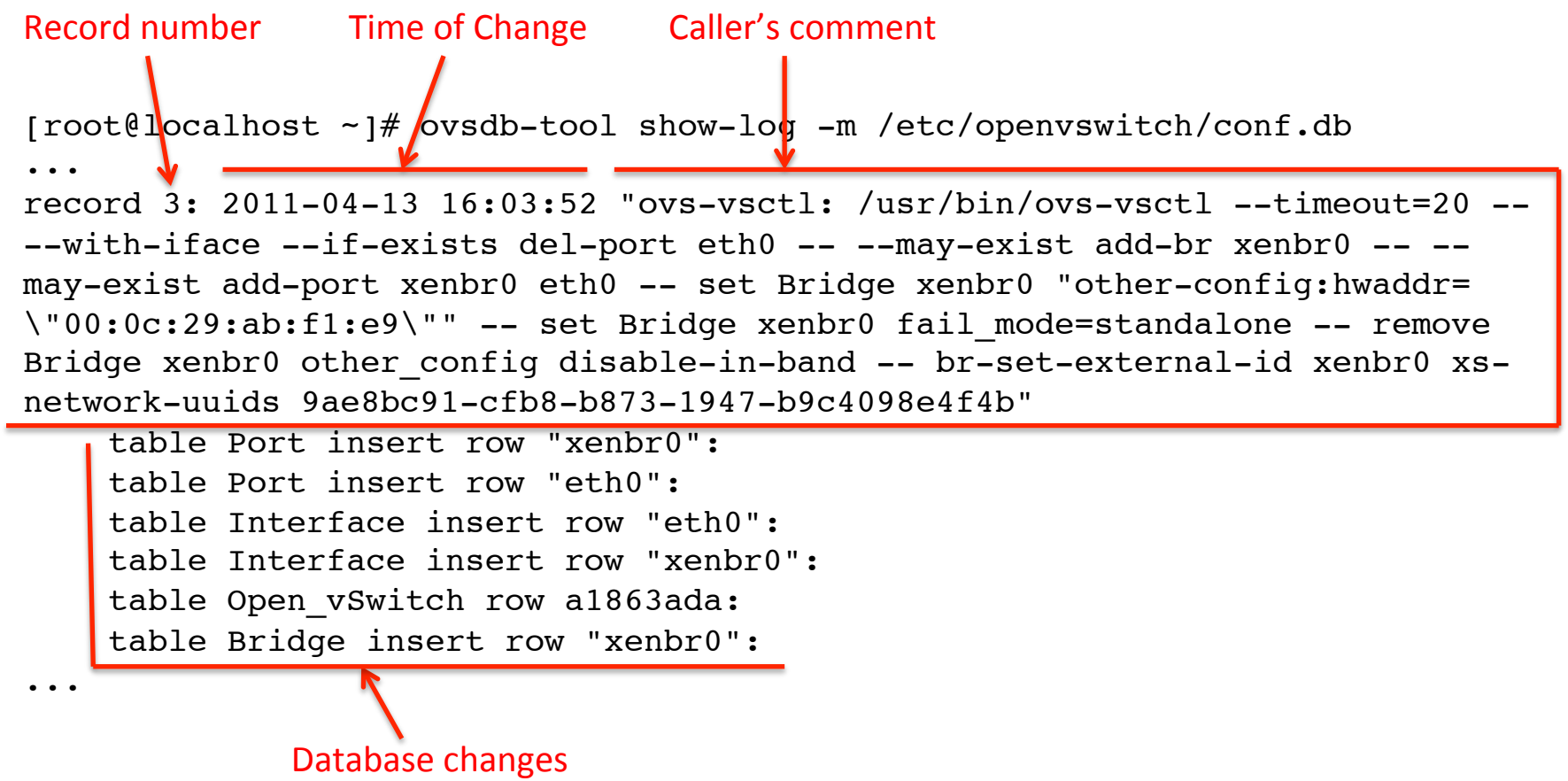
- ovs-vsctl: Configures ovs-vswitchd, but really a high-level interface for database
  - ovs-vsctl list-br
  - ovs-vsctl list-ports <bridge>
  - ovs-vsctl get-manager <bridge>
  - ovs-vsctl get-controller <bridge>
  - ovs-vsctl list <table>
- ovsdb-tool: Command-line tool for managing database file
  - ovsdb-tool show-log [-mmm] <file>

# Core Tables




“Open_vSwitch” is the root table and there is always only a single row. The tables here are the ones most commonly used; a full entity-relationship diagram is available in the ovs-vswitchd.conf.db man page.

# ovsdb-tool show-log



# OpenFlow

- ovs-ofctl speaks to OpenFlow module
  - ovs-ofctl dump-flows <bridge>
  - ovs-ofctl snoop <bridge>
- OpenFlow 1.0 plus extensions
  - Resubmit Action: Simulate multiple tables in a single table
  - NXM: Extensible match
  - Registers: Four 32-bit metadata registers
- See “hidden” flows (in-band, fail-open, etc):
  - ovs-appctl bridge/dump-flows <bridge>

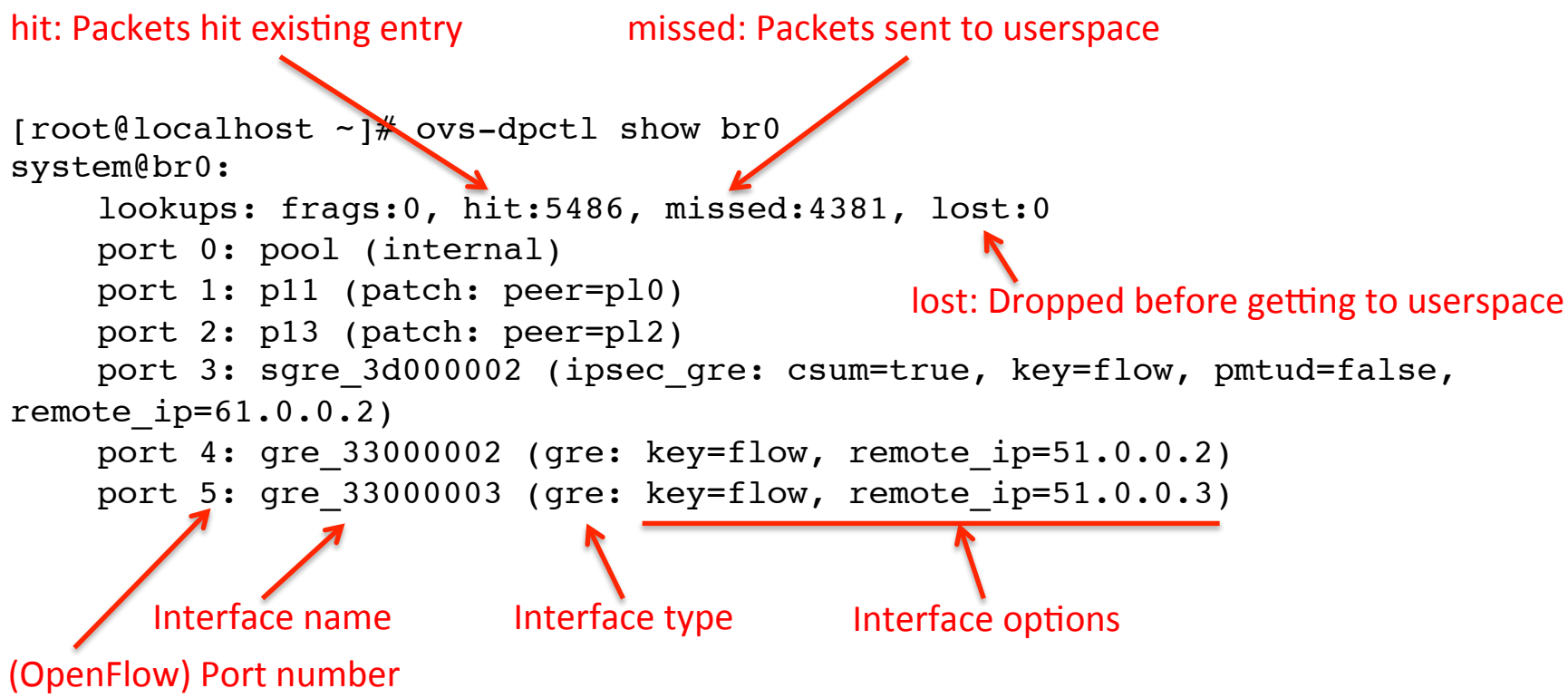
# Connectivity to Control Cluster

- State of connection tracked in database
  - ovs-vsctl list controller
  - ovs-vsctl list manager
- "status" column may contain the following members:
  - state: ACTIVE indicates that connection is good
  - sec_since_connect
  - sec_since_disconnect
  - last_error

# Kernel Datapath

- ovs-dpctl speaks to kernel module
- See datapaths and their attached interfaces:
  - ovs-dpctl show [bridge]
- Exact match flows cached in datapath:
  - ovs-dpctl dump-flows <bridge>

# ovs-dpctl show



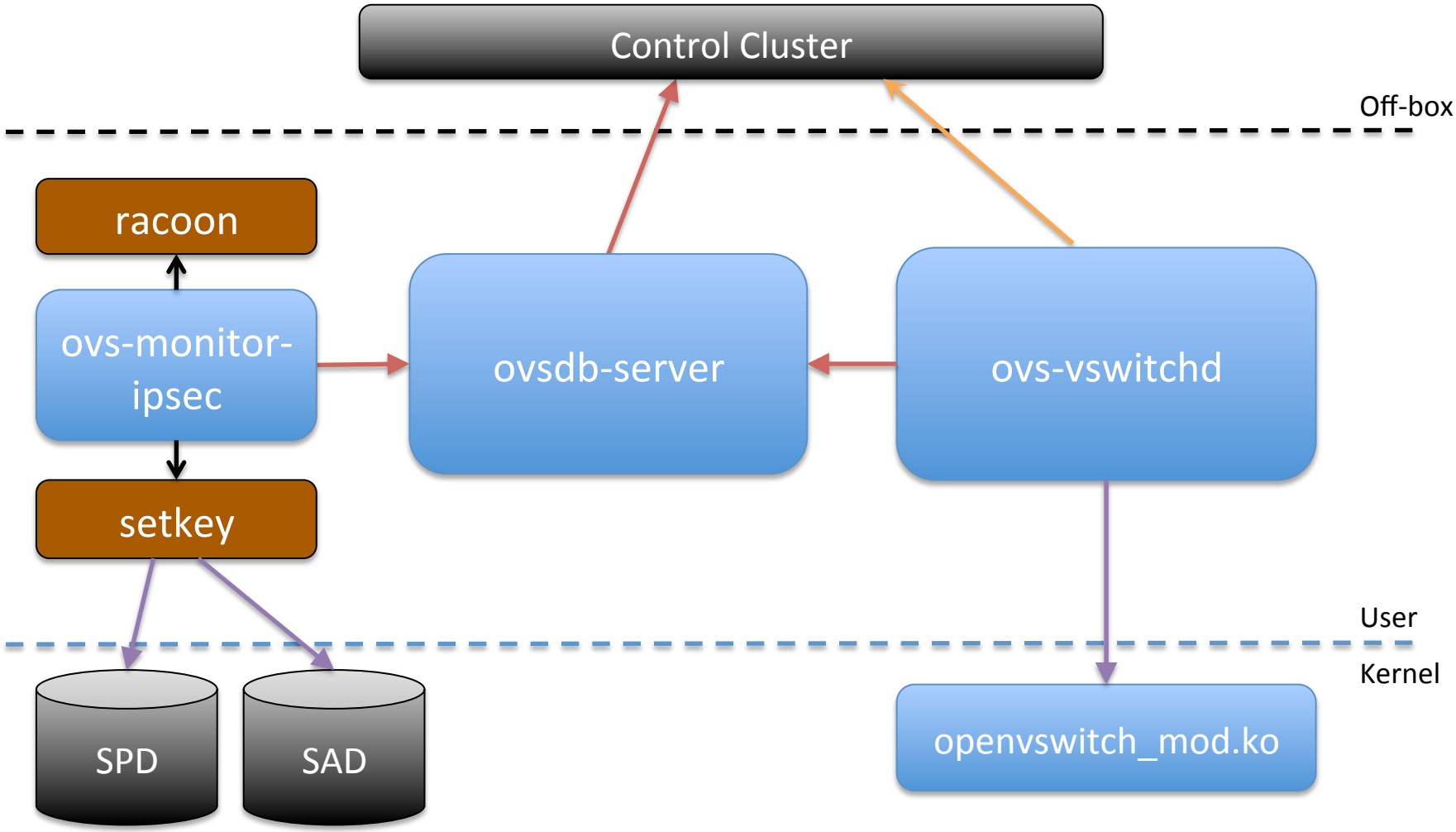
# Tunnels

- Tunnels in OVS are just virtual ports with own OpenFlow port number
- Keys set statically at creation time or dynamically through OpenFlow action
- Types:
  - GRE
  - GRE-over-IPsec
  - CAPWAP
- Visible in kernel datapath:
  - ovs-dpctl show

# IPsec Tunnels

- ovs-monitor-ipsec monitors database for changes and updates IPsec configuration
- racoon handles key negotiation (IKE)
- setkey configures security kernel databases
- SPD (Security Policy Database) determines when traffic should be encrypted
  - Dump SPD: setkey -DP
  - Flush SPD: setkey -FP
- SAD (Security Association Database) contains state for active flows
  - Dump SAD: setkey -D
  - Flush SAD: setkey -F

# IPsec Components



# IPsec Traffic Analysis

- Encrypted traffic on the PIF

```
root@squeeze-2:~# tcpdump -ni eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
12:36:45.974167 IP 172.16.5.57 > 172.16.3.79: ESP(spi=0x0baaab15,seq=0x33), length 124
12:36:45.974249 IP 172.16.3.79 > 172.16.5.57: ESP(spi=0x014d5d92,seq=0x35), length 124
```

- Decrypted traffic on the bridge

```
root@squeeze-2:~# tcpdump -ni br0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on br0, link-type EN10MB (Ethernet), capture size 65535 bytes
12:36:54.971521 IP 12.0.0.1 > 12.0.0.2: ICMP echo request, id 28173, seq 50, length 64
12:36:54.971536 IP 12.0.0.2 > 12.0.0.1: ICMP echo reply, id 28173, seq 50, length 64
```

# XenServer

- Is it running?  An upgrade doesn't necessarily enable OVS:
  - service openvswitch status
- Enable OVS:
  - xe-switch-network-backend openvswitch
- Disable OVS:
  - xe-switch-network-backend bridge

# Logging

- ovs-appctl configures running OVS daemons
- Most common use is to modify logging levels
- By default configures ovs-vswitchd, but “-t” option changes target
- Default level for log files is “info”, only thing lower is “dbg”

```
[root@localhost ~]# ovs-appctl vlog/list
                 console    syslog    file
                 -------    ------    ------
bridge            EMER        ERR       INFO
vswitchd          EMER        ERR       INFO
...
[root@localhost ~]# ovs-appctl vlog/set ofproto:file:dbg
```

# Log Files

- Open vSwitch logs: /var/log/openvswitch/*
  - ovs-vswitchd.log
  - ovsdb-server.log
- System: /var/log/messages
- IPsec: /var/log/daemon.log

# Debugging Tips

- Test basic connectivity
  - Remote side up?
  - STP learning state?
- Use tcpdump to see if expected packets are on the wire
- Try it without OVS

# Catastrophe!

- Bug details:
  - What were you doing when it happened?
  - OVS build number
  - OS version
- Collect logs and system state to aid debugging:
  - XenServer: xen-bugtool
  - Debian: ovs-bugtool
- Core dump
  - Check the version number, it may be old:
    - strings <core> | grep version
- Kernel Panic
  - Take picture of screen may be easiest

# Final Thought

Read the documentation…it’s pretty good!