FUJITSU

shaping tomorrow with you

# Neutron packet logging framework: Yesterday, Today and Tomorrow

Speakers:

- An Nguyen Phuong
- Nam Nguyen Hoai

# Agenda

- Who we are?
- Introduce the framework
- How do I setup logging API
- Demo
- Future plan

# Who we are?

- An Nguyen Phuong
  - Co-Author of neutron packet logging framework
  - Contributor of FWaaS V2
  - Neutron developer
  - IRC: annp
  - Email: annp@vn.fujitsu.com
  - Senior software engineer at FVL

# Who we are?

- Nam Nguyen Hoai
  - Co-organizer of Vienam openstack
  - Contributor of rolling upgrade
  - Barbican's developer
  - IRC: namnh
  - Email: namnh@vn.fujitsu.com
  - Software engineer at FVL

# Introduce the framework

- Yesterday:
  - Operator don’t know:
    - Which, where, when, … packets are ALLOW or DROP by security policy.
    - No way to make sure security rules works as expected.

# Introduce the framework

- Today:
  - Neutron packet logging just has been released in Queens with supporting for security group.
  - Now, Operators can know which packets are ALLOW or DROP at VMs.

# Introduce the framework

- Tomorrow…


*"tomorrow is coming"* ☺

# Introduce the framework

- Why logging API is introduced?
  - Can we extend security group extension with ‘log’ attribute? Of-course!
  - But we introduced new logging API because:
    - We’d like to provide a framework to collect packets log not only SG but also FW and SNAT …
    - Operators are so busy, so they need only once API instead of remember a lot of API.

# Introduce the framework

- How does logging API look like?

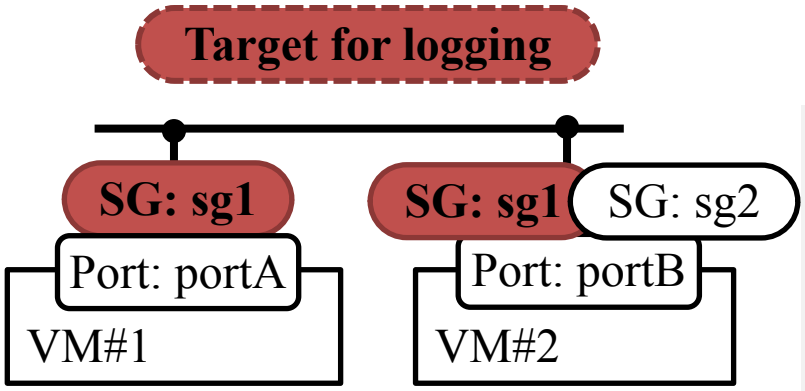
| HTTP method | URI | Description |
|---|---|---|
| GET | /v2.0/logging/loggable-resources | Show loggable neutron resource |
| POST | /v2.0/logging/logs | Create logging-resource |
| GET | /v2.0/logging/logs | List all logging-resources |
| GET | /v2.0/logging/logs/{logging-resource_id} | Show detail logging-resource |
| PUT | /v2.0/logging/logs/{logging-resource_id} | Update logging-resource |
| DELETE | /v2.0/logging/logs/{logging-resource_id} | Delete logging-resource |

# Introduce the framework

- A quick tour with logging API:
  - Check supported for neutron resources:

```
$ openstack network loggable resources list
+----------------+---------------+
| Supported Type |security_group |
+----------------+---------------+
```

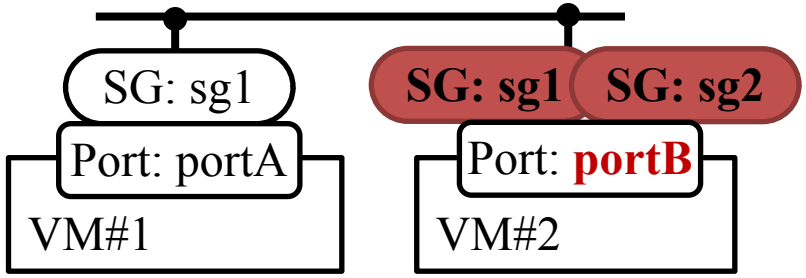
  - Create a logging resource for SG1 with DROP:




```
$ openstack network log create my-log \
--event drop \
--resource-type security_group \
--resource sg1
```

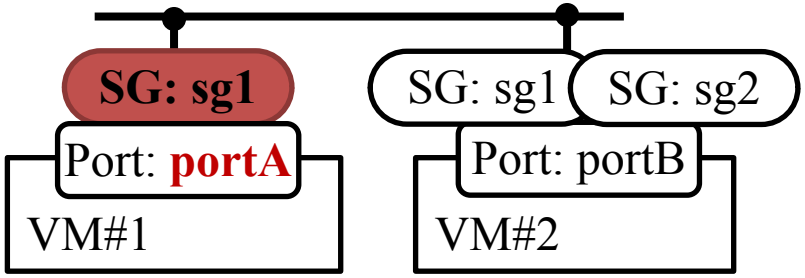
# Introduce the framework

- A quick tour with logging API:
  - Create a logging resource for only VM#2.




```
$ openstack network log create my-log \
--event drop \
--resource-type security_group \
--target portB
```

  - Create a logging resource for only VM#1 and SG1




```
$ openstack network log create my-log \
--event drop \
--resource-type security_group \
--resource sg1 --target portB
```

# Introduce the framework

- A quick tour with logging API:
  - Now, we can start logging!

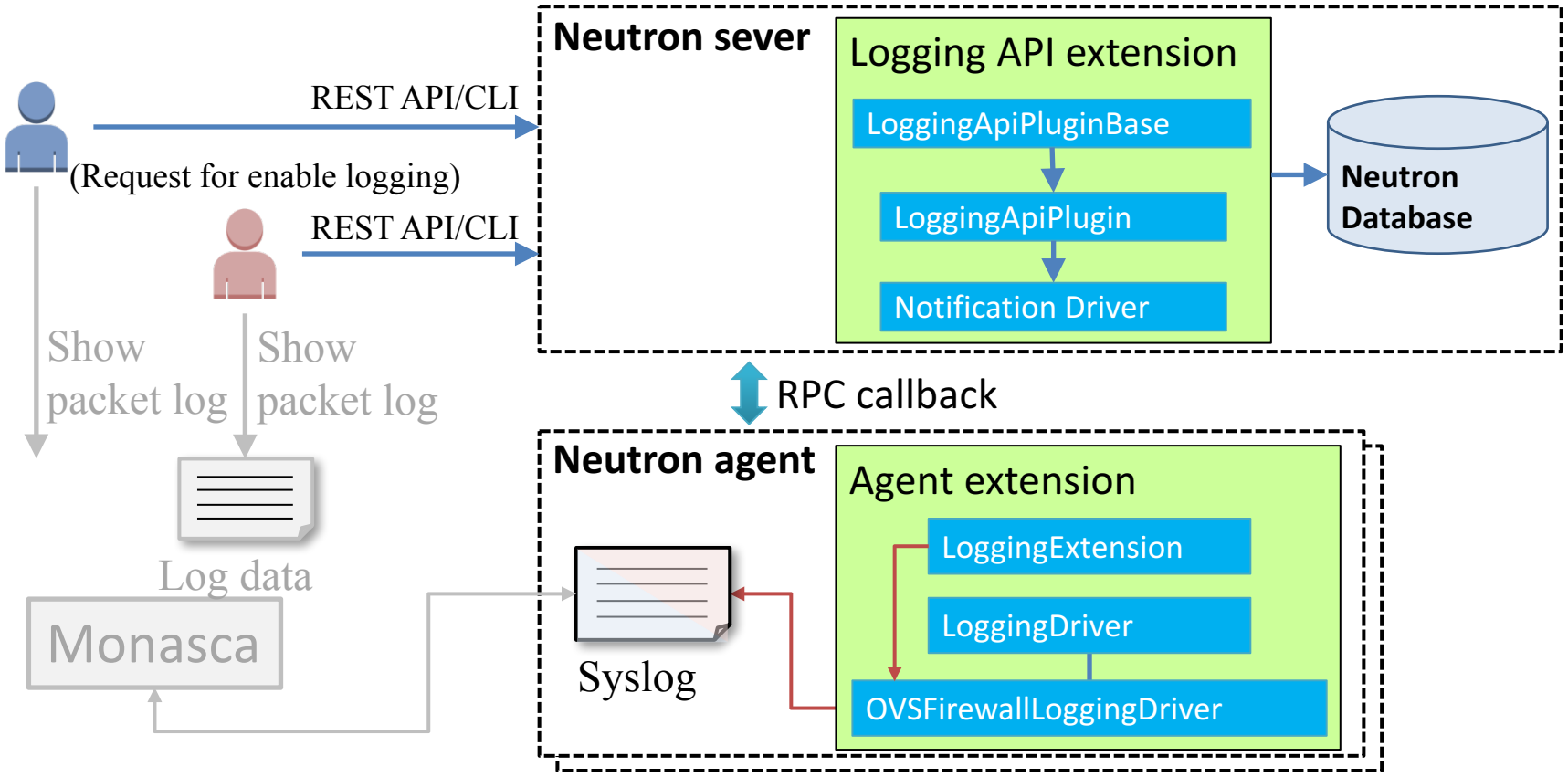
```
+-------------+--------------------------------------+
| Field       | Value                                |
+-------------+--------------------------------------+
| ID          | ef46cbbd-757e-4679-a928-f18b7c08bd35 |
| Description |                                      |
| Enabled     | True                                 |
| Name        | my-log                               |
| Target      | None                                 |
| Project     | b6b6e02d580d471caa1556a90bc3034d     |
| Resource    | 19e4d57c-8703-4a86-92a5-586436d39a6b |
| Type        | security_group                       |
| Event       | drop                                 |
+-------------+--------------------------------------+
```

  - If you'd like to stop logging:

```
$ openstack network log set my-log --disable-log
```

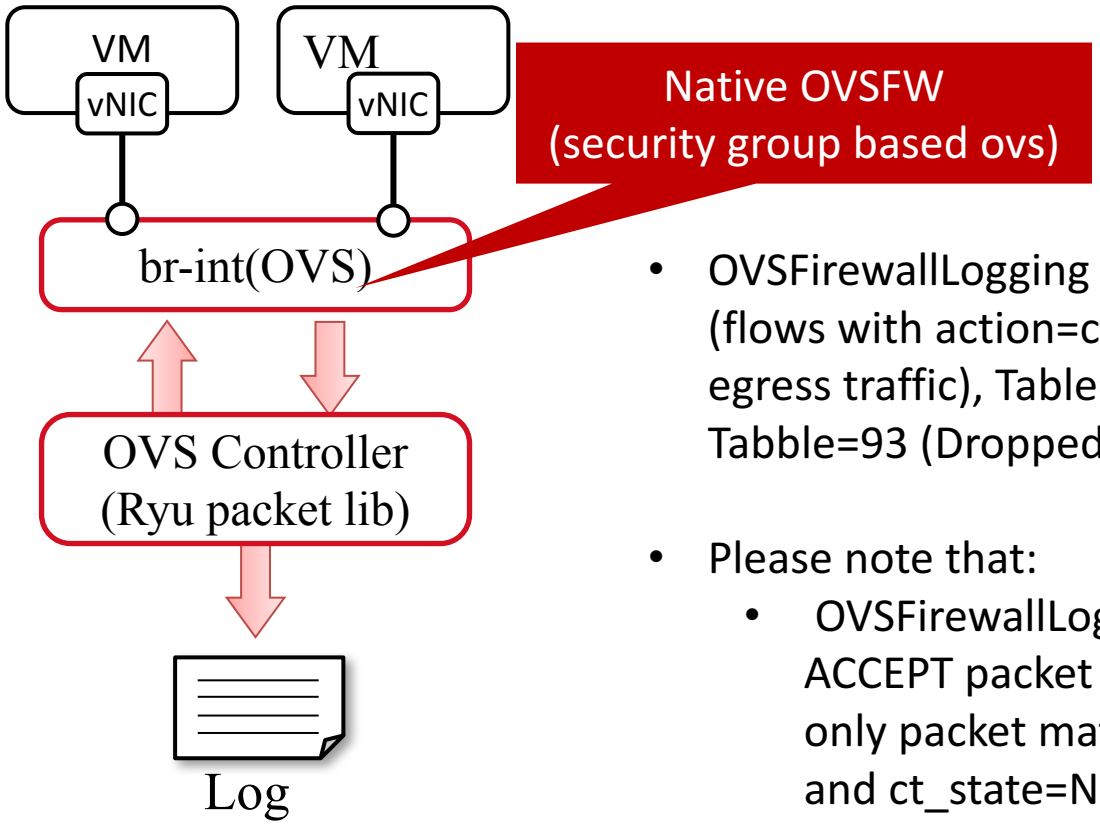
# Introduce the framework

- Overview architecture:



# Introduce the framework

- How packet log is collect?




- OVSFirewallLogging driver inserted flow log rules (flows with action=controller) at Table=91(Accepted egress traffic), Table=92(Accepted ingress traffic) and Tabble=93 (Dropped traffic).

- Please note that:
  - OVSFirewallLogging driver only logs first ACCEPT packet for each session. That means, only packet matched with security group rules and ct_state=NEW, then it will be logged.
  - Every drop packets will be logged.

# Introduce the framework

- How does packet log collect?
  - Flow log rule of ACCEPT ssh packet look like:

    table=92, priority=73,**ct_state=+new-est**,
    tcp,reg5=0x1c,dl_dst=fa:16:3e:6f:c9:47,tp_dst=22
    actions=ct(commit,zone=NXM_NX_REG6[0..15]),strip_vlan,output:28
    **CONTROLLER:65535**

  - Flow log rule of DROP traffic looks like:

    table=93, priority=53,reg5=0x1c **actions=CONTROLLER:65535**

# Introduce the framework

- Limitation of logging API in Queens release:
  - Currently, logging API only works on OVS agent + native OVSFW.
  - In future, logging api will support for Linuxbridge agent + Iptables
  - **We don’t have any plan to support iptables_hybrid.**

# Introduce the framework

- How do I consume log-data:
  - Log data would look like:

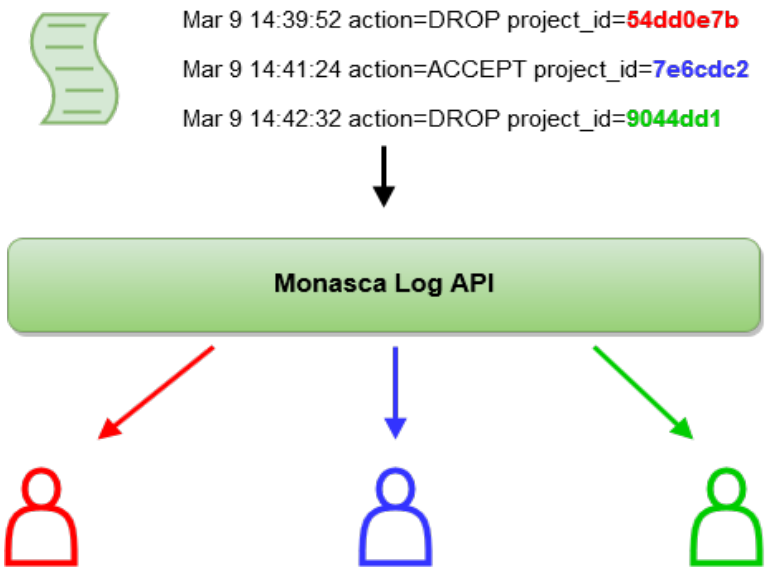
```
May 5 09:02:34 localhost greenthread.py: action=DROP
project_id=736672c700cd43e1bd321aeaf940365c
log_ids=['44a2e297-60ef-4bdd-8cad-14917cbefe9c']
vm_port=0720a67d-3e29-4231-891a-e0ac681bdc7f
pkt=ethernet(dst='fa:16:3e:6f:c9:47',ethertype=2048,src='fa:16:3e:50:aa:b5'),
ipv4(csum=43626,dst='10.0.0.11',flags=2,header_length=5,identification=55076,offset=0,
     option=None,proto=6,src='172.24.4.10',tos=0,total_length=60,ttl=63,version=4),
tcp(ack=0,bits=2,csum=14680,dst_port=22,offset=10,
    option=[TCPOptionMaximumSegmentSize(kind=2,length=4,max_seg_size=1460),
    TCPOptionSACKPermitted(kind=4,length=2),
    TCPOptionTimestamps(kind=8,length=10,ts_ecr=0,ts_val=196380390),
    TCPOptionNoOperation(kind=1,length=1),
    TCPOptionWindowScale(kind=3,length=3,shift_cnt=3)],
    seq=571457376,src_port=42838,urgent=0,window_size=14600)
```

- Date
- Packet action (ACCEPT or DROP)
- Project
- Log resource ID

- Where packet is dropped?
- Source/destination IP
- Source/destination MAC
- Source/destination port
- Protocol

# Introduce the framework

- How do I consume log-data:
  - We can consume log-data by Monasca service or other, then we can show log-data to each tenant as below:




POST
/v3.0/logs?tenant_id=54dd0e7b

# Introduce the framework

- How do I add new resource like firewall group to the framework:
  - No need add a new API.
  - Basically, we should register ‘firewall group’ resource to LoggingServiceDriverManager by provide a logging driver of firewall group (similar as:
    - https://github.com/openstack/neutron/tree/master/neutron/services/logapi/drivers/openvswitch).

# How do I setup logging API

- Enable logging service in server-side by setting in /etc/neutron/neutron.conf:
  - service_plugins=log,..
- Enable logging extension in agent-side by setting in /etc/neutron/plugins/ml2/ml2_conf.ini
  - [agent] extensions=log,…
- Reference:
  - https://docs.openstack.org/neutron/latest/admin/config-logging.html
  - https://developer.openstack.org/api-ref/network/v2/index.html#logging

# Demo

# Future plan

- In Rocky:
  - Support logging for security group based iptables on LinuxBridge agent [1]
  - SNAT log and Firewall log [2]
- After rocky:
  - Integrate with Monasca or develop a new project to detect attack pattern and alarm to user if there is some suspicion.

[1] https://review.openstack.org/#/c/445827/

[2] https://bugs.launchpad.net/neutron/+bug/1752290

# Reference

- https://www.openstack.org/videos/boston-2017/show-me-my-packet-log-neutron-packet-logging-with-monasca


THANK YOU FOR LISTENING
ANY QUESTION???
memegenerator.net