Security Group In Neutron OVSDPDK

Introduction

There are 3 security group implementations in neutron ovsdpdk agent.

1.Iptable based security group

The OVS agent and Compute service use a Linux bridge between each instance (VM) and the OVS integration bridge br-int to implement security groups. This implementation is stateful but cause scalability and performance problems.

2. Openflow based security group

This implementation is stateless.
https://github.com/openstack/networking-ovs-dpdk/blob/master/networking_ovs_dpdk/agent/ovs_dpdk_firewall.py

3. Openflow + conntrack based security group
   https://docs.openstack.org/newton/networking-guide/config-ovsfwdriver.html
   http://docs.openvswitch.org/en/latest/tutorials/ovs-conntrack/

In this paper, we try to verify openflow + contrack based security group.

Setup Environment

Devstack branch: stable/queens
OVSDPDK: 2.9.0
Contrack: 1.4.3

Devstack Configuration

[[post-config|/etc/nova/nova.conf]]
[securitygroup]
firewall_driver = noop

[[post-config|/etc/neutron/plugins/ml2/ml2_conf.ini]]
[ovs]
datapath_type = netdev
vhostuser_socket_dir = /var/lib/libvirt/qemu
[securitygroup]
firewall_driver = openvswitch

Verification Step

1. Create openstack VM and verify to use vhostuser interface

```
vagrant@compute-1:~$ sudo ovs-vsctl show  | grep -A4 "Port .vhu"
        Port "vhu54e5d135-97"
            tag: 1
            Interface "vhu54e5d135-97"
                type: dpdkvhostuserclient
                options: {vhost-server-path="/var/lib/libvirt/qemu/vhu54e5d135-97"}

vagrant@compute-1:~$ virsh dumpxml 1 | grep -A7 vhost
    <interface type='vhostuser'>
      <mac address='fa:16:3e:7d:44:81'/>
      <source type='unix' path='/var/lib/libvirt/qemu/vhu54e5d135-97' mode='server'/>
      <target dev='vhu54e5d135-97'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
```

2. Verify Ingress ICMP rule

Add ingress ICMP rule and ping to vm and dump openflow rules:

```
root@control:~# ping 10.0.0.7
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=2 ttl=64 time=998 ms
64 bytes from 10.0.0.7: icmp_seq=3 ttl=64 time=0.396 ms

vagrant@compute-1:~$ sudo ovs-ofctl dump-flows br-int  | grep 'ct_state.*icmp'
```

```
 cookie=0x847ea4e162bea178, duration=227.244s, table=82, n_packets=3, n_bytes=294,
idle_age=119, priority=75,ct_state=+est-rel-rpl,icmp,reg5=0x3 actions=output:3,resubmit(,92)
 cookie=0x847ea4e162bea178, duration=227.244s, table=82, n_packets=1, n_bytes=98,
idle_age=122, priority=75,ct_state=+new-est,icmp,reg5=0x3
actions=ct(commit,zone=NXM_NX_REG6[0..15]),output:3,resubmit(,92)
```

2. Verify Ingress ssh rule

Add ssh ingress rule and ssh to vm and verify openflow rules:

```
root@control:~# ssh cirros@10.0.0.7
cirros@10.0.0.7's password:
Permission denied, please try again.
cirros@10.0.0.7's password:
$
```

```
vagrant@compute-1:~$ sudo ovs-ofctl dump-flows br-int  | grep "ct_state.*22"
 cookie=0x847ea4e162bea178, duration=388.688s, table=82, n_packets=127, n_bytes=20654,
idle_age=3, priority=77,ct_state=+est-rel-rpl,tcp,reg5=0x3,tp_dst=22
actions=output:3,resubmit(,92)
 cookie=0x847ea4e162bea178, duration=388.688s, table=82, n_packets=4, n_bytes=296,
idle_age=67, priority=77,ct_state=+new-est,tcp,reg5=0x3,tp_dst=22
actions=ct(commit,zone=NXM_NX_REG6[0..15]),output:3,resubmit(,92)
```

3. Verify egress icmp rule

Delete all egress rules and add egress icmp rule and ping from vm

```
$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: seq=0 ttl=64 time=0.786 ms
64 bytes from 10.0.0.2: seq=1 ttl=64 time=0.655 ms
```

```
vagrant@compute-1:~$ sudo ovs-ofctl dump-flows br-int  | grep 'ct_state.*icmp'
 cookie=0x847ea4e162bea178, duration=46.104s, table=72, n_packets=14, n_bytes=1372,
idle_age=20, priority=75,ct_state=+est-rel-rpl,icmp,reg5=0x3 actions=resubmit(,73)
 cookie=0x847ea4e162bea178, duration=46.104s, table=72, n_packets=2, n_bytes=196,
idle_age=21, priority=75,ct_state=+new-est,icmp,reg5=0x3 actions=resubmit(,73)
```

3. Verify egress ssh rule

Delete all egress fules and add egress ssh rule and ssh from vm

```
$ ssh vagrant@192.168.0.10 "hostname"
vagrant@192.168.0.10's password:
control

vagrant@compute-1:~$ sudo ovs-ofctl dump-flows br-int  | grep "ct_state.*22"
 cookie=0x847ea4e162bea178, duration=131.559s, table=72, n_packets=60, n_bytes=7554,
idle_age=66, priority=77,ct_state=+est-rel-rpl,tcp,reg5=0x3,tp_dst=22 actions=resubmit(,73)
 cookie=0x847ea4e162bea178, duration=131.559s, table=72, n_packets=5, n_bytes=370,
idle_age=69, priority=77,ct_state=+new-est,tcp,reg5=0x3,tp_dst=22 actions=resubmit(,73)
```