Mirantis Technical Bulletin 2019-008

November 25, 2019

Open vSwitch daemon memory leak

ISSUE

Open vSwitch prior to v2.8.1 includes several memory leaks. The Open vSwitch daemon, aka `ovs-vswitchd`, can consume a lot of RAM causing a node overload. [0]

AFFECTS

This issue affects all OpenStack Pike and Queens environments deployed using MCP of any version prior to the MCP 2019.2.3 maintenance update.

SECURITY IMPACT

No security impact.

ESTIMATED TIME REQUIRED TO APPLY THE FIX

The time to apply the update is approximately 10 minutes per node with the Open vSwitch packages installed.

The estimated time calculations are based on the size of the default MCP configuration. The time estimation does not take into account neither preparation steps nor workload migration.

HOW TO DETERMINE IF YOU ARE AFFECTED

1. Log in to the Salt Master node.
2. Determine all nodes that require an Open vSwitch update:

```
$ sudo salt -C 'I@neutron:gateway or I@neutron:compute' pkg.version
python-openvswitch openvswitch-common openvswitch-switch
```

Example system response:

```
cmp01.myclustername.local:
    openvswitch-common:
        2.8.0-4~u16.04+mcp1
    openvswitch-switch:
        2.8.0-4~u16.04+mcp1
    python-openvswitch:
        2.8.0-4~u16.04+mcp1
gtw01.myclustername.local:
    openvswitch-common:
        2.8.0-4~u16.04+mcp1
    openvswitch-switch:
        2.8.0-4~u16.04+mcp1
    python-openvswitch:
        2.8.0-4~u16.04+mcp1
```

If the system response displays the `2.9.5-1~u16.04+mcp` or newer version of Open vSwitch, your deployment is not affected. Otherwise, apply the fix as described further in the bulletin.

The example system response above displays the `2.8.0-4~u16.04+mcp1` version meaning that the upgrade is required.

Before proceeding, store the `openvswitch-switch`, `openvswitch-common`, and `python-openvswitch` packages versions from the output for the verification of the revert procedure.

STEPS TO CORRECT

Prepare the deployment

WARNING
Plan a maintenance window according to your cloud size taking into account the potential networking downtime as well as additional time required for the workload migration.

1. Log in to the Salt Master node and start a screen session:

```
$ screen
```

2. Download packages with the fix:

```
$ wget https://artifactory.mirantis.com/fixes/openvswitch_2.9.5.tar.gz
$ tar -xvf openvswitch_2.9.5.tar.gz
```

3. Copy the required Open vSwitch `.deb` packages to the target nodes. The packages are located in the `pike` or `queens` directory inside the archive with the fix. For example:

CAUTION: When using the `scp` command, verify that the destination folder has correct permissions for the SSH user.

```
$ cd pike OR cd queens

$ sudo salt -C 'I@neutron:gateway or I@neutron:compute' cmd.run "mkdir
-p /tmp/ovs-2.9/"

$ sudo salt -C 'I@neutron:gateway or I@neutron:compute' cmd.run "chown
<your_ssh_user> /tmp/ovs-2.9/"

$ for i in $(sudo salt -C 'I@neutron:gateway or I@neutron:compute'
test.ping --out=txt|cut -d':' -f1); do scp -o "StrictHostKeyChecking
no" *.deb $i:/tmp/ovs-2.9/; done
```

4. Verify that the required OpenvSwitch `.deb` packages exist on the target nodes:

```
$ sudo salt -C 'I@neutron:gateway or I@neutron:compute' cmd.run "ls -l
/tmp/ovs-2.9/"
```

Update the gateway nodes

1. Back up `neutron.conf` on controller nodes:

```
$ sudo salt -C "I@neutron:server" cmd.run "cp /etc/neutron/neutron.conf
/etc/neutron/neutron.conf.bkp"
```

2. On controller node, set `allow_automatic_dhcp_failover` and `allow_automatic_l3agent_failover` to `False` and restart `neutron-server`:

```
$ sudo salt -C "I@neutron:server" cmd.run "sed -i
'/allow_automatic_dhcp_failover/d' /etc/neutron/neutron.conf"

$ sudo salt -C "I@neutron:server" cmd.run "sed -i
'/allow_automatic_l3agent_failover/d' /etc/neutron/neutron.conf"

$ sudo salt -C "I@neutron:server" cmd.run "sed -i '/\[DEFAULT\]/a
allow_automatic_dhcp_failover=false' /etc/neutron/neutron.conf"

$ sudo salt -C "I@neutron:server" cmd.run "sed -i '/\[DEFAULT\]/a
allow_automatic_l3agent_failover=false' /etc/neutron/neutron.conf"
```

   This will disable rescheduling of resources between gateway nodes while performing the update.

3. Restart the `neutron-server` service on the controller nodes:

```
$ sudo salt -b 1 --batch-wait=30 -C "I@neutron:server" service.restart
neutron-server
```

4. Determine the gateway nodes to be updated:

```
$ sudo salt -C "I@neutron:gateway" test.ping --out=txt | cut -d':' -f1
```

WARNING
Execute the steps 5-10 only on one gateway node at a time.

5. Stop Neutron agents on a gateway node:

```
$ sudo salt "%NODE_HOSTNAME%" service.stop neutron-dhcp-agent
$ sudo salt "%NODE_HOSTNAME%" service.stop neutron-l3-agent
$ sudo salt "%NODE_HOSTNAME%" service.stop neutron-metadata-agent
```

```
$ sudo salt "%NODE_HOSTNAME%" service.stop neutron-openvswitch-agent
```

6. Update the Open vSwitch packages:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "apt update && apt install -y -f
/tmp/ovs-2.9/*.deb"
```

7. Restart the `openvswitch-switch` service:

```
$ sudo salt "%NODE_HOSTNAME%" service.restart openvswitch-switch
```

8. Verify that the package has been updated:

```
$ sudo salt "%NODE_HOSTNAME%" pkg.version python-openvswitch
openvswitch-common openvswitch-switch
```

   The packages versions should match or be higher than the versions of the packages located in the archive.

9. Start the `neutron-openvswitch` agent:

   a. Execute the following commands:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "logrotate
/etc/logrotate.d/neutron-common"

$ sudo salt "%NODE_HOSTNAME%" service.start
neutron-openvswitch-agent
```

   b. Wait for the following log message: `Configuration for devices up [] and devices down [] completed.`

```
$ sleep 30 #seconds after starting openvswitch agent
$ sudo salt "%NODE_HOSTNAME%" cmd.run "grep -E 'Configuration for
devices up \[.*\] and devices down \[.*\] completed.'
/var/log/neutron/neutron-openvswitch-agent.log"
```

10. Start the other Neutron agents:

```
$ sudo salt "%NODE_HOSTNAME%" service.start neutron-dhcp-agent
$ sudo salt "%NODE_HOSTNAME%" service.start neutron-l3-agent
$ sudo salt "%NODE_HOSTNAME%" service.start neutron-metadata-agent
```

11. Repeat steps 5-10 for all gateway nodes.
12. Restore `neutron.conf` on the controller nodes:

```
$ sudo salt -C "I@neutron:server" cmd.run "mv
/etc/neutron/neutron.conf.bkp /etc/neutron/neutron.conf"

$ sudo salt -C "I@neutron:server" cmd.run "chown neutron:neutron
/etc/neutron/neutron.conf"
```

13. Restart the `neutron-server` service on the controller nodes:

```
$ sudo salt -b 1 --batch-wait=30 -C "I@neutron:server" service.restart
neutron-server
```

Update the compute nodes

WARNING
To avoid a brief data-plane downtime for the VMs on the node being updated, perform the live-migration of these VMs from the `cmp` node before the upgrade. If migration is not available, you may experience a networking downtime for workloads.

WARNING
Perform the upgrade of the Open vSwitch packages on small batches of compute nodes, preferably one by one.

1. Determine the compute nodes to be updated:

```
$ sudo salt -C "I@neutron:compute" test.ping --out=txt | cut -d':' -f1
```

2. Migrate workloads if possible and required (see the OpenStack operator guide [1] for details).
3. Determine the Neutron agents to be stopped:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "service neutron-*-agent status |
grep \*"
```

   The Neutron agent list depends on the cluster configuration and can contain only one `neutron-openvswitch-agent` service.

Example system response:

```
$ sudo salt "cmp001.myclustername.local" cmd.run "service
neutron-*-agent status | grep \*"
cmp001.myclustername.local:
      * neutron-openvswitch-agent.service - Openstack Neutron Open
vSwitch Plugin Agent
      * neutron-metadata-agent.service - OpenStack Neutron Metadata
Agent
      * neutron-l3-agent.service - OpenStack Neutron L3 agent
```

The above example system response demonstrates that the following Neutron agents are running on the environment:

- neutron-openvswitch-agent
- neutron-metadata-agent
- neutron-l3-agent

4. Stop Neutron agents on a compute node:

```
$ sudo salt "%NODE_HOSTNAME%" service.stop %NEUTRON_AGENT_NAME%
```

5. Update the Open vSwitch packages:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "apt update && apt install -y -f
/tmp/ovs-2.9/*.deb"
```

6. Restart the `openvswitch-switch` service on the compute node:

```
$ sudo salt "%NODE_HOSTNAME%" service.restart openvswitch-switch
```

7. (Optional) If workloads has not been migrated from the node:

    a. Create a temporary script that will clean up ports:

```
$ cat <<\EOF >>/tmp/fix-ports.sh
#!/bin/bash
for i in $(ip -o link | grep " qvo" | cut -d@ -f1 | awk '{print
$2}' | sort | uniq ) ; do if ovs-vsctl show | grep $i ; then echo
found; else echo 'removing port' "$i"; ovs-vsctl '--if-exists'
'del-port' 'br-int' "$i"; ip link del "$i" ;fi; done | tee
'/tmp/remove-ports.txt'
EOF
```

b. Upload the script to the compute node:

```
$ sudo salt-cp "%NODE_HOSTNAME%" /tmp/fix-ports.sh
/tmp/fix-ports.sh
```

b. Run the script:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run '/bin/bash
/tmp/fix-ports.sh' | tee removing-broken-ports-and-fix.txt
```

8. Restart the `nova-compute` service that will recreate the OVS ports:

```
$ sudo salt "%NODE_HOSTNAME%" service.restart nova-compute
```

9. Start the `neutron-openvswitch` agent:

a. Execute the following commands:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "logrotate
/etc/logrotate.d/neutron-common"

$ sudo salt "%NODE_HOSTNAME%" service.start
neutron-openvswitch-agent
```

b. Wait for the following log message: `Configuration for devices up [] and devices down [] completed.`

```
$ sleep 30 #seconds after starting openvswitch agent

$ sudo salt "%NODE_HOSTNAME%" cmd.run "grep -E 'Configuration for
devices up \[.*\] and devices down \[.*\] completed.'
/var/log/neutron/neutron-openvswitch-agent.log"
```

10. If required, start the rest of the Neutron agents on the compute node:

```
$ sudo salt "%NODE_HOSTNAME%" service.start %NEUTRON_AGENT_NAME%
```

11. Verify that the packages has been updated:

```
$ sudo salt "%NODE_HOSTNAME%" pkg.version python-openvswitch
openvswitch-common openvswitch-switch
```

The packages versions should match or be higher than the versions of the packages located in the archive.

12. Repeat steps 2-11 for each compute node.

STEPS TO VERIFY THE PATCH

1. Log in to the Salt Master node and verify logs from the Open vSwitch service:

```
$ sudo salt -C 'I@neutron:gateway or I@neutron:compute' cmd.run
"systemctl status openvswitch-switch"
```

2. Verify the Neutron agents status:

```
$ sudo salt "ctl01*" cmd.run ". ./keystonercv3; neutron agent-list"
```

3. Create a network.
4. Create a subnet.
5. Boot a test instance on the network and attach a floating IP to it.
6. Create a security rule and verify that it applies correctly.
7. Create a router and verify that it works correctly.

STEPS TO REVERT THE PATCH

Prepare the deployment

1. Log in to the Salt Master node.
2. Verify that the candidate version of the Open vSwitch packages matches the version that was written down during How to determine if you are affected -> Step 2.

```
$ sudo salt -C 'I@neutron:gateway or I@neutron:compute' cmd.run
"apt-cache policy openvswitch*"
```

3. On the controller nodes, back up `neutron.conf`:

```
$ sudo salt -C "I@neutron:server" cmd.run "cp /etc/neutron/neutron.conf
/etc/neutron/neutron.conf.bkp"
```

4. On the controller nodes, set `allow_automatic_dhcp_failover` and `allow_automatic_l3agent_failover` to `False` and restart `neutron-server`:

```
$ sudo salt -C "I@neutron:server" cmd.run "sed -i
'/allow_automatic_dhcp_failover/d' /etc/neutron/neutron.conf"

$ sudo salt -C "I@neutron:server" cmd.run "sed -i
'/allow_automatic_l3agent_failover/d' /etc/neutron/neutron.conf"

$ sudo salt -C "I@neutron:server" cmd.run "sed -i '/\[DEFAULT\]/a
allow_automatic_dhcp_failover=false' /etc/neutron/neutron.conf"

$ sudo salt -C "I@neutron:server" cmd.run "sed -i '/\[DEFAULT\]/a
allow_automatic_l3agent_failover=false' /etc/neutron/neutron.conf"
```

   This will disable rescheduling of resources between gateway nodes while performing the rollback.

5. Restart the `neutron-server` service on the controller nodes:

```
$ sudo salt -b 1 --batch-wait=30 -C "I@neutron:server" service.restart
neutron-server
```

Roll back the gateway nodes

1. Determine the gateway nodes to downgrade:

```
$ sudo salt -C "I@neutron:gateway" test.ping --out=txt | cut -d':' -f1
```

WARNING
Perform the downgrade steps 2-7 only on one gateway node at a time.

2. Stop Neutron agents on a gateway node:

```
$ sudo salt "%NODE_HOSTNAME%" service.stop neutron-dhcp-agent
$ sudo salt "%NODE_HOSTNAME%" service.stop neutron-l3-agent
$ sudo salt "%NODE_HOSTNAME%" service.stop neutron-metadata-agent
$ sudo salt "%NODE_HOSTNAME%" service.stop neutron-openvswitch-agent
```

3. Re-install the Open vSwitch packages using the version identified in How To Determine If You Are Affected > Step2:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "apt-get install -y --reinstall
--allow-downgrades openvswitch-switch=<version>
openvswitch-common=<version> python-openvswitch=<version>"
```

For example:

```
$ sudo salt "gtw01.myclustername.local" cmd.run "apt-get install -y
--reinstall --allow-downgrades openvswitch-switch=2.8.0-4~u16.04+mcp1
openvswitch-common=2.8.0-4~u16.04+mcp1
python-openvswitch=2.8.0-4~u16.04+mcp1"
```

4. Restart the `openvswitch-switch` service:

```
$ sudo salt "%NODE_HOSTNAME%" service.restart openvswitch-switch
```

5. Verify that the package has been re-installed:

```
$ sudo salt "%NODE_HOSTNAME%" pkg.version python-openvswitch
openvswitch-common openvswitch-switch
```

The packages versions should match the packages versions identified in How To Determine If You Are Affected > Step2

6. Start the `neutron-openvswitch` agent:

    a. Execute the following commands:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "logrotate
/etc/logrotate.d/neutron-common"

$ sudo salt "%NODE_HOSTNAME%" service.start
neutron-openvswitch-agent
```

    b. Wait for the following log message: `Configuration for devices up [] and devices down [] completed.`

```
$ sleep 10 #seconds after starting openvswitch agent

$ sudo salt "%NODE_HOSTNAME
%" cmd.run "grep -E 'Configuration for devices up \[.*\] and
devices down \[.*\] completed.'
/var/log/neutron/neutron-openvswitch-agent.log"
```

7. Start the rest of the Neutron agents:

```
$ sudo salt "%NODE_HOSTNAME%" service.start neutron-dhcp-agent
$ sudo salt "%NODE_HOSTNAME%" service.start neutron-l3-agent
$ sudo salt "%NODE_HOSTNAME%" service.start neutron-metadata-agent
```

8. Repeat steps 2-7 for all gateway nodes.

9. Restore `neutron.conf` on the controller nodes:

```
$ sudo salt -C "I@neutron:server" cmd.run "mv
/etc/neutron/neutron.conf.bkp /etc/neutron/neutron.conf"

$ sudo salt -C "I@neutron:server" cmd.run "chown neutron:neutron
/etc/neutron/neutron.conf"
```

10. Restart the `neutron-server` service on the controller nodes:

```
$ sudo salt -b 1 --batch-wait=30 -C "I@neutron:server" service.restart
neutron-server
```

Roll back the compute nodes

WARNING
To avoid a brief data-plane downtime for the VMs on the node being downgraded, perform the live-migration of these VMs from the `cmp` node before the downgrade. If migration is not available, you may experience a networking downtime for workloads.

WARNING
Perform the downgrade of the Open vSwitch packages on small batches of compute nodes, preferably one by one.

1. Determine the compute nodes to be downgraded:

```
$ sudo salt -C "I@neutron:compute" test.ping --out=txt | cut -d':' -f1
```

2. Migrate workloads if possible and required (see OpenStack operator guide [1] for details).
3. Determine the Neutron agents to be stopped:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "service neutron-*-agent status |
grep \*"
```

4. Stop Neutron agents on a compute node:

```
$ sudo salt "%NODE_HOSTNAME%" service.stop %NEUTRON_AGENT_NAME%
```

5. Re-install the Open vSwitch packages using the version identified in How To Determine If You Are Affected > Step2:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "apt-get install -y --reinstall
--allow-downgrades openvswitch-switch=<version>
openvswitch-common=<version> python-openvswitch=<version>"
```

6. Restart the `openvswitch-switch` service on the node:

```
$ sudo salt "%NODE_HOSTNAME%" service.restart openvswitch-switch
```

7. (Optional) If workloads have not been migrated from the node:

a. Create a temporary script that will clean up ports:

```
$ cat <<\EOF >>/tmp/fix-ports.sh
#!/bin/bash
for i in $(ip -o link | grep " qvo" | cut -d@ -f1 | awk '{print
$2}' | sort | uniq ) ; do if ovs-vsctl show | grep $i ; then echo
found; else echo 'removing port' "$i"; ovs-vsctl '--if-exists'
'del-port' 'br-int' "$i"; ip link del "$i" ;fi; done | tee
'/tmp/remove-ports.txt'
EOF
```

b. Upload the script to the compute node:

```
$ sudo salt-cp "%NODE_HOSTNAME%" /tmp/fix-ports.sh
/tmp/fix-ports.sh
```

c. Run the script:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run '/bin/bash
/tmp/fix-ports.sh' | tee removing-broken-ports-and-fix.txt
```

8. Restart the `nova-compute` service to recreate OVS ports:

```
$ sudo salt "%NODE_HOSTNAME%" service.restart nova-compute
```

9. Start the `neutron-openvswitch` agent:

a. Execute the following commands:

```
$ sudo salt "%NODE_HOSTNAME%" cmd.run "logrotate
/etc/logrotate.d/neutron-common"

$ sudo salt "%NODE_HOSTNAME%" service.start
neutron-openvswitch-agent
```

b. Wait for the following log message: `Configuration for devices up [] and devices down [] completed.`

```
$ sleep 10 #seconds after starting openvswitch agent
$ sudo salt "%NODE_HOSTNAME
```

```
%" cmd.run "grep -E 'Configuration for devices up \[.*\] and
devices down \[.*\] completed.'
/var/log/neutron/neutron-openvswitch-agent.log"
```

10. If required, start the rest of Neutron agents on the compute node:

```
$ sudo salt "%NODE_HOSTNAME%" service.start %NEUTRON_AGENT_NAME%
```

11. Verify that the package has been re-installed:

```
$ sudo salt "%NODE_HOSTNAME%" pkg.version python-openvswitch
openvswitch-common openvswitch-switch
```

    The packages versions should match the packages versions determined in How To Determine If You Are Affected > Step2.

12. Repeat steps 2-11 for all compute nodes.
13. Perform the steps to verify the patch.

REFERENCES

[0] https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2017-14970

[1] https://docs.openstack.org/nova/pike/admin/live-migration-usage.html