Mirantis Technical Bulletin 2019-006

July 29, 2019

SACK Panic and Other TCP Denial of Service Issues

ISSUE

The issue specifically relates to the Maximum Segment Size (MSS) and TCP Selective Acknowledgement (SACK) capabilities. The most serious vulnerability, known as *SACK Panic,* allows a remotely triggered kernel panic on Linux kernels. Therefore, Mirantis highly recommends completing the steps described in this bulletin to prevent the nodes deadlock on your MCP-based OpenStack environments or MOS environments.

The CVE entries for the described vulnerabilities include:

- CVE-2019-11477 for a remote denial of service (system crash) known as SACK Panic [1]
- CVE-2019-11478 for a remote denial of service (resource exhaustion) [2]
- CVE-2019-11479 for a remote denial of service (resource exhaustion) [3]

CVE-2019-11477 is the issue of the highest severity level. A remote attacker can leverage this vulnerability to immediately crash a system due to an integer overflow when processing TCP SACKs. The issue affects all current Ubuntu releases.

CVE-2019-11478 is the issue of the high severity level. The issue can have a significant impact on CPU performance when processing TCP SACKs. The issue affects the systems running the kernels of the versions 4.14 and older. These include the Ubuntu 16.04 LTS base kernel.

CVE-2019-11479 can impact CPU performance while a TCP stack is handling a malicious session that was opened using a very small MSS value. The vulnerability has less impact on a system compared to the other two vulnerabilities mentioned above. The issue affects all current Ubuntu releases and will be addressed in a set of future Ubuntu kernel updates.

AFFECTS

The issue affects any Ubuntu-based node running at the publication date of this bulletin including all OpenStack and Kubernetes nodes in the MCP-based deployments as well as all OpenStack nodes in MOS-based deployments.

The current bulletin contains the mitigation procedures that apply to the OpenStack-based nodes only. The mitigation plan for the described vulnerabilities for the Kubernetes-based compute nodes differs and is not the subject of this technical bulletin.

SECURITY IMPACT

The hacker attacks using the described vulnerabilities can totally disable the target node or slow down its performance up to the denial-of-service state. Though, an unexpected kernel panic can lead to data corruption, no data leaks are expected.

MCP: STEPS TO PREVENT

CVE-2019-11477 and CVE-2019-11478

The CVE-2019-11477 and CVE-2019-11478 vulnerabilities have been mitigated in the latest Linux kernels that include linux-image-4.4.0-151-generic 4.4.0-151.178 and linux-image-4.15.0-52-generic 4.15.0-52.56~16.04.1. The procedure below enables you to mitigate the risk from the issues without upgrading the kernel.

> **NOTE**
> Only nondestructive actions are performed during this stage.

1. Log in to the Salt Master node.
2. Verify the kernel version on all nodes:

```
salt '*' cmd.run 'uname -r'
```

   If the base kernel version is higher than 4.4.0-151.178 or the HWE kernel version is higher than 4.15.0-52.56~16.04.1, proceed to step 5.

3. Open your project Git repository with the Reclass model on the cluster level.
4. In the `infra/init.yml` file, define the `net.ipv4.tcp_sack: 0` parameter:

```
parameters:
  ...
  linux:
    ...
    system:
      kernel:
        sysctl:
          net.ipv4.tcp_sack: 0
```

5. Refresh pillars and sync salt data:

```
salt '*' saltutil.refresh_pillar
salt '*' saltutil.sync_all
```

6. Apply the `kernel` state to disable TCP SACK support:

```
salt '*' state.apply linux.system.kernel
```

7. Verify that the fix has been applied:

```
salt '*' cmd.run 'sysctl net.ipv4.tcp_sack'
```

   The system response should show the **0** value.

CVE-2019-11479

You can mitigate the CVE-2019-11479 vulnerability through the IPTables rules.

WARNING
Before proceeding, analyze the instruction below taking into account the networking configuration of each affected node. The procedure includes the creation of the firewall rule that drops all TCP packets with `Minimum Segment Size (MSS)` lower than `500`. Verify that the MCP cluster connection does not rely on `MSS` lower than `500`.

NOTE
The procedure below does not create the firewall rule for the Salt Master node to ensure that you can revert the changes if required. Though, you can apply the same steps to the Salt Master node later as appropriate.

1. Log in to the Salt Master node.
2. Verify that `net.ipv4.tcp_mtu_probing` is disabled:

```
salt '*' cmd.run 'sysctl net.ipv4.tcp_mtu_probing'
```

   If the system response contains `1`, run:

```
salt '*' cmd.run 'sysctl -w net.ipv4.tcp_mtu_probing=0'
```

   Otherwise, proceed to the next step.
3. Create the firewall rule that drops TCP packets with `MSS` lower than `500`:

   WARNING
   We recommend that you apply the rule to one node at first and validate if the node and cluster are operational.

```
salt '*' cmd.run 'sudo iptables -A INPUT -p tcp -m tcpmss --mss 1:500 -j
DROP'
```

4. Open your project Git repository with the Reclass model on the cluster level.
5. In the `infra/init.yml` file, add the firewall rule to the `if-pre-up` scripts. The firewall rule will restore after rebooting. Therefore, make the rule persistent. For example:

```
parameters:
...
  linux:
    ...
    system:
    ...
      file:
        /etc/network/if-pre-up.d/iptables:
          user: root
          mode: "0711"
          contents: |
            #!/bin/bash
            /sbin/iptables -C INPUT -p tcp -m tcpmss --mss 1:500 -j DROP
> /dev/null 2>&1 || {
            /sbin/iptables -A INPUT -p tcp -m tcpmss --mss 1:500 -j DROP
            }
```

6. Apply the changes:

```
salt '*' saltutil.refresh_pillar
salt '*' saltutil.sync_all
salt '*' state.apply linux.system.file
```

7. Verify that the file has been created:

```
salt '*' cmd.run "cat /etc/network/if-pre-up.d/iptables"
```

During the next restart of the networking service, the newly added rule will be recreated if needed.

MCP: STEPS TO REVERT THE PATCH

CVE-2019-11477 and CVE-2019-11478

1. Log in to the Salt Master node.
2. Open your project Git repository with the Reclass model on the cluster level.
3. In the `infra/init.yml` file, define `net.ipv4.tcp_sack: 1`:

NOTE
If you just remove the pillar, it will not clean up the configuration file and system flag.

```
parameters:
  ...
  linux:
    ...
    system:
      kernel:
        sysctl:
          net.ipv4.tcp_sack: 1
```

4. Refresh pillars and sync salt data:

```
salt '*' saltutil.refresh_pillar
salt '*' saltutil.sync_all
```

5. Re-enable the TCP SACK support:

```
salt '*' state.apply linux.system.kernel
```

6. Verify that the fix has been reverted:

```
salt '*' cmd.run 'sysctl net.ipv4.tcp_sack'
```

The system response should show the `1` value.

CVE-2019-11479

1. Log in to the Salt Master node.
2. Delete the firewall rule:

```
salt -C '* and not cfg01*' cmd.run 'sudo iptables -D INPUT -p tcp -m
tcpmss --mss 1:500 -j DROP'
```

3. Remove the `if-pre-up` configuration file with the firewall rule from the cluster model. For details, see *Steps To Prevent - MCP-based clusters > CVE-2019-11479 > Step 5*.

4. Manually remove the `if-pre-up` config files from the target nodes:

```
salt '*' cmd.run "rm /etc/network/if-pre-up.d/iptables"
```

5. If required, restore `net.ipv4.tcp_mtu_probing`:

```
salt -C '* and not cfg01*' cmd.run 'sysctl -w net.ipv4.tcp_mtu_probing=1'
```

MOS: STEPS TO PREVENT

As the update of the kernel 4.4 for Ubuntu 14.04 is not expected, we recommend mitigating the issue by disabling the TCP SACK.

CVE-2019-11477 and CVE-2019-11478

1. Log in to the Fuel Master node
2. Disable the TCP SACK support by running the sample command below. The execution of this command sets `net.ipv4.tcp_sack=0` for each node in the `ready` state. Modify the command according to your cluster configuration.

```
fuel node | grep ready | awk '{print $9}' | xargs -tI{} ssh {} "sed -i
'/net.ipv4.tcp_sack/d' /etc/sysctl.conf && echo 'net.ipv4.tcp_sack=0' >>
/etc/sysctl.conf && sysctl -p"
```

CVE-2019-11479

WARNING

Before proceeding, analyze the instruction below taking into account the networking configuration of each affected node. The procedure includes the creation of the firewall rule that drops all TCP packets with `Minimum Segment Size (MSS)` lower than `500`. Verify that the MOS cluster connection does not rely on `MSS` lower than `500`.

1. Log in to the Fuel Master node.
2. Ensure that `net.ipv4.tcp_mtu_probing` is disabled:

```
fuel node | grep ready | awk '{print $9}' | xargs -tI{} ssh {} "sysctl
net.ipv4.tcp_mtu_probing"
```

   If the system response contains `1`, run:

```
fuel node | grep ready | awk '{print $9}' | xargs -tI{} ssh {} "sysctl -w
net.ipv4.tcp_mtu_probing=0"
```

   Otherwise, proceed to the next step.

3. On each cluster node, create the firewall rule that drops all packets with `MSS` below `500`:

WARNING
We recommend that you apply the rule to one node at first and validate if the node and cluster are operational.

```
ssh node-X
iptables -A INPUT -p tcp -m tcpmss --mss 1:500 -j DROP
```

4. Verify that the node and the whole cluster are operational.
5. After you apply the firewall rule to all target nodes, save it to make it persistent for the rebooting nodes:

```
fuel node | grep ready | awk '{print $9}' | xargs -tI{} ssh {}
"iptables-save > /etc/iptables/rules.v4"
```

MOS: STEPS TO REVERT THE PATCH

CVE-2019-11477 and CVE-2019-11478

1. Log in to the Fuel Master node.
2. Restore the TCP SACK support:

```
fuel node | grep ready | awk '{print $9}' | xargs -tI{} ssh {} "sed -i
'/net.ipv4.tcp_sack/d' /etc/sysctl.conf && sysctl -w
net.ipv4.tcp_sack=1"
```

CVE-2019-11479

1. Log in to the Salt Master node.
2. Delete the firewall rule:

```
fuel node | grep ready | awk '{print $9}' | xargs -tI{} ssh {} "sudo
iptables -D INPUT -p tcp -m tcpmss --mss 1:500 -j DROP"
```

3. Update the persistent iptables:

```
fuel node | grep ready | awk '{print $9}' | xargs -tI{} ssh {}
"iptables-save > /etc/iptables/rules.v4"
```

4. If required, restore `net.ipv4.tcp_mtu_probing`:

```
fuel node | grep ready | awk '{print $9}' | xargs -tI{} ssh {} "sysctl -w
net.ipv4.tcp_mtu_probing=1"
```

REFERENCES

[0] https://wiki.ubuntu.com/SecurityTeam/KnowledgeBase/SACKPanic

[1] https://people.canonical.com/~ubuntu-security/cve/2019/CVE-2019-11477.html

[2] https://people.canonical.com/~ubuntu-security/cve/2019/CVE-2019-11478.html

[3] https://people.canonical.com/~ubuntu-security/cve/2019/CVE-2019-11478.html