Red Hat

Tainted kernel tech preview

Table 1. Revision History

Version	Date	Who	Changes
1.0	Jun 27, 2023	David Rabkin	Initial Version

Table 2. Terminology

Acronym	Meaning
BIOS	Basic Input/Output System
CNF	Containerized Network Function
GA	General Availability
kTLS	Kernel Transport Layer Security
Metal3	Bare metal host provisioning for Kubernetes
RHCOS	Red Hat Enterprise Linux CoreOS operating system
RHEL	Red Hat Enterprise Linux operating system
OCP	OpenShift Container Platform

History

A major part of Telco partner certification is CNF Certification Test. These test cases are specifically designed to verify whether the deployment of the Telco partner's solution on Red Hat OpenShift follows the best practices. By conducting these tests, we ensure that the implementation aligns with the recommended standards and guidelines for optimal performance and compatibility.

Among the 88 test cases in the suite, one of them is named `platform-alteration-tainted-node-kernel`. It verifies that the nodes hosting CNFs do not utilize tainted kernels. For more detailed information regarding tainted kernels in the Linux kernel, you can refer to the documentation.

During the certification process of OCP versions 4.10 and 4.11, our team encountered difficulties related to tainted kernels. The majority of failures were attributed to incomplete kernel updates or outdated BIOS firmware versions. Resolving the issue was possible by performing reboots and updating the BIOS firmware. I recommend referring to a well-written article dedicated to this particular issue.

Story

The `platform-alteration-tainted-node-kernel` test in CNF Certification Tests v4.2.4 consistently fails, accompanied by the following error message:

```
Taints mask=65536 - Decoded taints: [auxiliary taint, defined for and used by
distros (tainted bit 16)]
```

The test examines the taint status of the kernel on each node, indicating whether the kernel is tainted or not:

```
Pod IP: 192.168.26.51
If you don't see a command prompt, try pressing enter.
sh-4.4# chroot /host
sh-4.4# cat /proc/sys/kernel/tainted
65536
```

A zero value indicates that the kernel is in a satisfactory state:

```
Pod IP: 192.168.26.52
If you don't see a command prompt, try pressing enter.
sh-4.4# chroot /host
sh-4.4# cat /proc/sys/kernel/tainted
0
```

The tainted kernel test continues to fail on one of the master nodes. The job reports the failure on `master-1`, and when I reboot that particular master with the tainted kernel, the failure then occurs on the next master, which is `master-2`. This failure pattern follows a migration sequence of `0->1->2->0` after each reboot.

OCP 4.12 uses RHCOS based on 8.6. From here:

```
With the TAINT_TECH_PREVIEW mask value being removed upstream and within RHEL 9
the TAINT_AUX bit is used instead of TAINT_TECH_PREVIEW within 8.6 and later
kernels.
```

The node with the taint reports that it is running a technical preview module:

```
sh-4.4# dmesg|grep TECH
[17980.575402] TECH PREVIEW: kTLS may not be fully supported.
sh-4.4# uname -a
Linux master-0.r207-6node.r207.lab.eng.cert.redhat.com
4.18.0-372.49.1.el8_6.x86_64 #1 SMP Thu Mar 9 21:11:55 EST 2023 x86_64 x86_64
x86_64 GNU/Linux
```

There is an existing bug (Red Hat internal) that is approximately six months old, which attributes the issue to Metal3. The bug has been resolved with the final resolution stating that there is no solution for version 4.12, and the taint is expected to be resolved when transitioning to version 4.13. The issue in Metal3 is caused by using RHEL 9-based images to build the container, which use some system calls that end up loading `kTLS`. Further details regarding this matter can be found in the Slack thread (Red Hat internal).

Problem

The certification team faces a challenge in explaining why the GA version of OCP 4.12 utilizes `kTLS`, which is still in the Technical Preview phase. This situation raises concerns for security experts, as it could be perceived as a potential security vulnerability. The tests, performed by our partners on their CNFs, further emphasize the need to address this issue and provide a satisfactory explanation.

Resolution

One potential solution is to revert Metal3 back to RHCOS8, which would eliminate the use of `kTLS` in the GA version as it is intended. More information about this resolution can be found in the accompanying comment (Red Hat internal).