RPMDB Corruption

128 Technology has observed situations where the RPM (Red Hat Package Manager) database becomes corrupted, and results in asset failures visible in `show assets` and its various subcommands.

When the RPMDB is corrupted, there will be asset errors that accumulate over time until manual intervention is taken. By default, each managed asset will attempt to retrieve upgradable software versions every ten minutes. When the RPMDB is corrupt, these attempts fail and cause an "asset error."

The impact of this error is nominal, but it will prevent the system from being upgraded to a newer software version and may affect other interactions that leverage SaltStack (in the event that there are custom salt states applied to your assets).

Symptoms

Impacted systems will potentially see a large number of alarms, clearly visible in the output of `show assets summary`:

```
admin@node1.bernstein# show assets summary
Mon 2020-03-16 09:14:17 EDT

================================
 Summary of Assets
================================
 total:                    1661
 disconnected:             20
 running:                  1630
 connected:                11

 ------------------
 Versions Installed
 ------------------
 4.1.5-3.el7.centos:       1620
 4.1.7-1.el7.centos:       8
 4.2.1-1.el7:              2

 ------
 Errors
 ------
 LR201905001872:           206
```

Here we can see the asset `LR201905001872` has 206 errors.

From the Linux shell on the host, run `sudo systemctl status salt-minion`:

```
● salt-minion.service - The Salt Minion
   Loaded: loaded (/usr/lib/systemd/system/salt-minion.service; enabled; vendor
preset: disabled)
  Drop-In: /usr/lib/systemd/system/salt-minion.service.d
           └─dnsTimeout.conf, minionWatchdog.conf, restartAlways.conf
   Active: active (running) since Sun 2020-03-15 19:25:38 UTC; 23h ago
     Docs: man:salt-minion(1)
           file:///usr/share/doc/salt/html/contents.html
           https://docs.saltstack.com/en/latest/contents.html
 Main PID: 29314 (salt-minion)
    Tasks: 14
   Memory: 96.5M
   CGroup: /system.slice/salt-minion.service
           ├─29314 /usr/bin/python /usr/bin/salt-minion
           ├─29319 /usr/bin/python /usr/bin/salt-minion
           └─29339 /usr/bin/python /usr/bin/salt-minion

Mar 16 19:04:02 LR201905001872 salt-minion[29314]: error: cannot open Packages
database in /var/lib/rpm
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: [ERROR   ] Error occurred
installing package(s). Additional info follows:
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: errors:
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: - Running scope as unit run-
21047.scope.
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: error: rpmdb: BDB0113
Thread/process 5966/140205829543744 failed: BDB1507 Thread died in Berkeley DB
library
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: error: db5 error(-30973) from
dbenv->failchk: BDB0087 DB_RUNRECOVERY: Fatal error, run database recovery
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: error: cannot open Packages
index using db5 -  (-30973)
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: error: cannot open Packages
database in /var/lib/rpm
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: CRITICAL:yum.main:
Mar 16 19:04:02 LR201905001872 salt-minion[29314]: Error: rpmdb open failed
```

There are clear indicators of issues with the RPMDB.

Repairing the RPMDB

Fortunately, rebuilding the RPMDB is quick and requires no downtime.

First, move the existing (corrupt) RPMDB to `/tmp`, where it will be deleted upon reboot (after thirty days):

```
sudo mv /var/lib/rpm/__db* /tmp
```

Next, clean the Yum and DNF caches. Yum is used by the 128T software to retrieve upgrade packages, and DNF is used to upgrade/rollback software versions. They both use the RPMDB.

```
sudo yum clean all
sudo dnf clean all
```

Last, restart the salt-minion:

```
sudo systemctl restart salt-minion
```

Confirming the Restoration

On your conductor, once the asset has reconnected it should transition to the `running` state and no longer show any errors.