tuned: local root exploit in D-Bus method instance_create and other issues in tuned >= 2.23 (CVE-2024-52336, CVE-2024-52337)

Nov 26, 2024 • Matthias Gerstner

#CVE #LOCAL #D-BUS

Table of Contents

1) Introduction

Tuned is a privileged daemon for Linux that supports automatic tuning of various hardware and kernel settings during runtime. The daemon offers a comprehensive D-Bus interface protected by Polkit authentication. We regularly perform reviews of newly introduced D-Bus system services and changes to them. Tuned sees frequent additions to its D-Bus interface and this is

already the tenth review of it that we carried out since 2019. Usually the reviews are straightforward and we have no complaints, but this time was the exception.

During the review we checked the D-Bus methods matching the following Polkit actions:

```
com.redhat.tuned.instance_create        (auth_admin:auth_admin:yes)
com.redhat.tuned.instance_destroy       (auth_admin:auth_admin:yes)
net.hadess.PowerProfiles.HoldProfile    (no:no:yes)
net.hadess.PowerProfiles.ReleaseProfile (no:no:yes)
```

This report is based on tuned release v2.24.0.

2) Problems in the `instance_create` D-Bus Method

Calling the `instance_create()` D-Bus method is allowed without authentication for locally logged-in users (`yes` Polkit setting). The method call accepts various parameters, including an `options` dictionary, that are fully under attacker control.

2a) Script Options Allow Local Root Exploit (CVE-2024-52336)

The `script_pre` and `script_post` options allow to pass arbitrary scripts that will be executed by *root*. The parameters are extracted in daemon/controller.py:459, stored unmodified in a new `Instance` object and the only verification of the script path is performed in plugins/base.py:222:

```
    if not script.startswith("/"):
        log.error("Relative paths cannot be used in script_pre or script_p
                + "Use ${i:PROFILE_DIR}.")
        return False
```

So the only requirement is that an absolute path is passed. Thus, scripts under control of an unprivileged user can be passed here. This allows for a local root exploit.

Reproducer

As a locally logged-in non-privileged user execute the following D-Bus call:

```
    $ gdbus call -y -d com.redhat.tuned -o /Tuned \
        -m com.redhat.tuned.control.instance_create cpu myinstance \
```

```
        '{"script_pre": "/path/to/myscript.sh", "devices": "*"}'
```

The path */path/to/myscript.sh* needs to be replaced by a path to a user controlled executable script or program. It will be executed by tuned with root privileges.

2b) Instance Name can Contain Arbitrary Data (CVE-2024-52337)

The `instance_name` parameter of the `instance_create()` method is not sanitized. This string is later on used in logging and in the output of utilities like `tuned-adm get_instances`, or other third party programs that utilize tuned's D-Bus interface to obtain instance names.

A local attacker can include arbitrary data in the instance name and can achieve log spoofing this way. By placing newline characters into the name, seemingly independent, legitimate-looking entries can be added to the tuned log. By adding terminal control sequences the terminal emulators of administrators or other users can be influenced. The following is a Proof-of-Concept for this:

```
    $ EVIL=`echo -e "this is\nevil\033[?1047h"`
    $ gdbus call -y -d com.redhat.tuned -o /Tuned -m com.redhat.tuned.cont
```

When another user now calls `tuned-adm get_instances` then the terminal emulator will switch to the alternate screen upon output of the crafted instance name.

Affectedness

The `instance_create()` D-Bus method has been added via upstream commit cddcd233 and was first part of version tag v2.23.0. The initial version already contained support for the script option parameters and the `instance_name` parameter.

3) Problems in the PowerProfiles Interface

3a) Cookie in PowerProfiles API is Predictable

The new D-Bus methods `HoldProfile()` and `ReleaseProfile()` use a cookie to identify a profile hold. The cookie is simply a continuously increasing integer starting at zero. This means other users in the system can easily release the profile holds of other users.

3b) User Supplied Strings can Contain Arbitrary Data

The `HoldProfile()` call accepts `reason` and `app_id` strings which are used in logging and may also be returned as a dictionary via `ProfileHold.as_dict()`. These strings can again contain crafted data that could have side effects similar to the ones shown in section 2b).

Suggested Fix

A local DoS scenario using the `cookie` would only be an issue on multi-user systems, or if the Polkit settings are relaxed so that also non-local sessions can use these D-Bus methods. One way to make this more robust could be to hand out random `cookie` IDs instead, to make the attack less trivial.

4) Bugfixes

Upstream published release v2.24.1 that addresses the issues described in this report. Commit 90c24eea037 contains the cumulative fixes as follows:

- plugins are now only loaded from trusted locations (`_safe_script_path()` function).
- various user supplied strings are rejected if they contain disallowed characters (`is_valid_name()` function).
- upstream also tightened the tuned Polkit policy for `instance_create` and a number of other actions, that they also found to be problematic when accessed by local unprivileged users.

For issue 3a) no upstream fix is available at the moment. This is more of a hardening suggestion, though.

5) Timeline

2024-11-07	We reported the issues to the Red Hat security team.
2024-11-08	Red Hat security confirmed the issues and communicated the CVE assignments to us, and the publication date of 2024-11-26 was suggested.

2024-11-11	Red Hat shared the suggested patch and we reviewed it.
2024-11-26	The publication date has been reached and publication happened as planned.

6) References

- Tuned GitHub project
- Bugfix release v2.24.1
- Bugfix commit

SUSE Security Team Blog

SUSE Security Team
security@suse.de

Open Source vulnerability reports and code review results.