Date: Wed, 10 Feb 2021 10:28:42 +0100
From: Matthias Gerstner <mgerstner@...e.de>
To: oss-security@...ts.openwall.com
Subject: Replay-Sorcery: CVE-2021-26936: Multiple security issues in with
 setuid-root program in versions 0.4.0 through 0.5.0

Hello,

we received a review request [1] for ReplaySorcery [2] for inclusion in the
openSUSE Linux distribution. ReplaySorcery allows to record short videos of
screen content, triggered via a key combination. Since version 0.4.0 released
on 2020-12-19 through to the current version 0.5.0 the replay-sorcery program
is by default installed with setuid-root and (unnecessarily) setgid-root bits
and is thus running with root privileges. The motivation for this was to
improve screen capture performance via vaapi, which requires `CAP_SYS_ADMIN`
privileges [3].

I reviewed the security of ReplaySorcery in the setuid-root context. The
outcome of the review is that the replay-sorcery program is not fit to run as
setuid-root in the currently released versions. The program does not take any
of the many precautions that are necessary to avoid security issues in
setuid-root programs. The issues start with things like failure to establish
safe environment variables and end with careless file system accesses with
elevated rights. There happens no user privilege management at all i.e. the
program runs with full root privileges all of the time.

Following are a couple of specific issues I could find right away
(probably not a complete list):

a) The $HOME environment variable is interpreted by the program. Thus an
   unprivileged user can cause the configuration files of other users to be
   used, or output videos to be created in arbitrary (home) directories.
b) The $DISPLAY environment variable is interpreted, which could allow in
   theory to record videos from other users' X displays. Together with setting
   $XAUTHORITY to another user's Xauthority file this nearly allows to do that.
   Only that fact that libX11 is doing an `access()` check on the Xauthority
   file first comes to the rescue (`access()` takes the real user ID into
   account). For other graphic systems like Wayland or kms the outcome
   might be different, I did not extensively test that.
c) When reading config files in ~/.config/replay-sorcery.conf symlinks are
   followed. This allows for arbitrary file existence tests, opening of
   arbitrary special files (with potential side effects in the kernel) and also
   parsing files not normally accessible to the calling unprivileged user. The
   parsing will typically fail but could leak information from the file in some
   circumstances (e.g. through logging, when the target format matches the
   configuration file syntax in some ways).
d) When writing video output files into the user's home directory (by default
   ~/Videos/ReplaySorcery_%F_%H-%M-%S.mp4) then symlinks will be followed.
   Either the Videos folder or the target filename itself can be symlinks
   (apart from being able to setting $HOME to arbitrarily change the home
   directory). Even when the timestamp with second granularity is used it
   is pretty simple to pre-create a range of symlinks resulting in
   arbitrary file overwrite, resulting in local denial-of-service.
e) By configuring a user specific `outputFile` in ~/.config/replay-sorcery.conf
   like

      outputFile = /etc/ld.so.conf.d/mylib.conf

   the video will be created in the path in `/etc/ld.so.conf.d.d`.
   When setting `umask 0` before running the replay-sorcery program then
   this file will receive mode 0666 and owner root:root. Thus it can be
   edited by anybody. This can allow for a full local root exploit via
   various vectors depending on the target directory.
   If the target path already exists then it will only be overwritten but
   the mode will remain the same. This still allows for a denial-of-service.

I reported these issues to the upstream developer on 2021-01-29. We discussed
various approaches to fix the issues. By now two upstream commits [4], [5]
greatly improve the situation by dropping effective capabilities to the
unprivileged user and only obtain root privileges for calling into ffmpeg
library functions when the vaapi acceleration is necessary. I could not find
any obvious security issues with this new approach but it still feels uneasy
calling into the ffmpeg library in a setuid-root context. Also the
replay-sorcery code does not yet take precautions to clear the environment and
set a safe umask value. I urged the upstream developer to do that as well.

As a workaround for these security issues ReplaySorcery can be built with the
CMake setting `-DRS_SETID=OFF` to prevent installation with setuid-root and
setgid-root bits. The only drawback will be the missing vaapi acceleration
in certain configurations.

[1]: https://bugzilla.suse.com/show_bug.cgi?id=1181321
[2]: https://github.com/matanui159/ReplaySorcery
[3]: https://trac.ffmpeg.org/wiki/Hardware/VAAPI#ScreenCapture
[4]: https://github.com/matanui159/ReplaySorcery/commit/d6580072582a31c72fdf70fdc80431eddeb3ddc6
[5]: https://github.com/matanui159/ReplaySorcery/commit/557e8e80ab7934bfe8521f96c237ea62b961e74e

Cheers

Matthias

--
Matthias Gerstner <matthias.gerstner@...e.de>
Dipl.-Wirtsch.-Inf. (FH), Security Engineer
https://www.suse.com/security
Phone: +49 911 740 53 290
GPG Key ID: 0x14C405C971923553

SUSE Software Solutions Germany GmbH
HRB 36809, AG Nürnberg
Geschäftsführer: Felix Imendörffer

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