[TrouSerS-tech] Multiple Security Issues in the TrouSerS tpm1.2 tscd Daemon

Brought to you by: ashleylai, dvelarde, gcwilson, hcl2014, and 4 others

[TrouSerS-tech] Multiple Security Issues in the TrouSerS tpm1.2 tscd Daemon

[TrouSerS-tech] Multiple Security Issues in the TrouSerS tpm1.2 tscd Daemon

From: Matthias Gerstner <mgers...@su...> - 2020-05-20 12:54:48

Attachments: signature.asc tcsd_fixes.patch

Hello,

I have discovered multiple security issues in the tcsd daemon of the TrouSerS
[1] tpm 1.2 stack.

Introduction
============

The tcsd daemon manages access to the tpm 1.2 compliant /dev/tpm0 device on
Linux systems. The daemon utilizes an unprivileged user and group account to
run as. These are called tss:tss by default.

The tcsd can be started directly as the tss user and group e.g. via systemd or
via start-stop-daemon. In this case the /dev/tpm0 device needs to be owned by
the tss user. This mode of operation is safe and is not affected by the
following findings.

If the tcsd is started with root privileges then it opens /dev/tpm0 as root
and drops privileges to the unprivileged user afterwards. In this case the tss
user can achieve privilege escalations. The following logic is performed by
the tcsd:

1) the daemon reads in the configuration in /etc/tcsd.conf after making sure
   that the config file is owned by tss:tss mode 0600 (function
   `conf_file_init()`). From this configuration file the path `system_ps_file`
   (by default /var/lib/tpm/system.data) is parsed and used for further
   operations.

2) the daemon makes sure that the directory where the `system_ps_file` is
   contained in exists (function `ps_dirs_init()`, /var/lib/tpm by default).
   The directory is created, if necessary, using `mkdir()` and mode 0700.
   Afterwards an explicit `chown()` to mode 0700 is made in case the mode of
   the directory doesn't match this mode yet.

3) in the function `ps_init_disk_cache()` the function `get_file()` is called
   which opens the `system_ps_file` using `O_RDWR|O_CREAT` and mode 0600:

   `openat(AT_FDCWD, "/var/lib/tpm/system.data", O_RDWR|O_CREAT, 0600) = 4`

4) only after these steps a privilege drop to the tss uid is performed in
    the `main()` function.

Security Issues
===============

The security issues resulting from this are as follows:

a) Since /var/lib/tpm is owned by the tss user (as per dist/Makefile.am), the
    creation of the `system.data` file in step 3) is prone to symlink attacks. The
    tss user can thereby cause the creation of new files or the corruption of
    existing files. These new files end up with mode 0600 and no `chown()` to the
    tss user is performed by the tcsd. Thus it looks like no full local root
    privilege escalation can be achieved but only DoS attacks.

b) The tcsd only drops the root uid, not the root gid in step 4). A call to
    `setgid()` is missing. Therefore the tcsd continues to run with root group
    privileges it doesn't actually require. This could allow further privilege
    escalations when combined with other, yet unknown attack vectors.

c) The configuration file /etc/tcsd.conf is _required_ by the tcsd to be
    owned by tss:tss mode 0600. Therefore the unprivileged user can change all
    daemon related settings, including the `system_ps_file` path. This means
    the `mkdir()` and `chmod()` performed in step 2) can be directed to an
    arbitrary path. This also includes the symlink attack described in a)
    for arbitrary paths.

    Further security issues could stem from this by manipulating other config
    file options. I did not look deeper into this.

d) Not directly related to the logic above. The example RPM spec file [5] in
    the TrouSerS repository is using unsafe file and directory modes for
    /var/lib/tpm and /usr/sbin/tcsd:

    ```
    # create the default location for the persistent store files
    if test -e %{_localstatedir}/tpm; then
         mkdir -p %{_localstatedir}/tpm
         /bin/chown tss:tss %{_localstatedir}/tpm
         /bin/chmod 1777 %{_localstatedir}/tpm
    fi

    # chown the daemon
    /bin/chown tss:tss %{_sbindir}/tcsd
    ```

    So here a public sticky-bit directory is setup in /var/lib/tpm. This could
    allow arbitrary users to setup the symlink attack mentioned in a). It could
    also lead to an information leak. Once the tcsd is started as root the mode
    of /var/lib/tpm will be corrected in step 1), however.

    Passing ownership of /usr/sbin/tcsd to the tss user would allow the tss
    user to replace the tcsd binary by malicious code that will potentially be

```
   executed by the root user, leading to arbitrary code execution.

   I'm not aware of any distribution actually using this spec file or parts of
   it. Still it is a very bad example.

Mitigation and Bugfixes
=======================

It seems best to me to run the tcsd as the tss:tss user and group right away
and to not rely on the privilege drop logic implemented in the daemon itself.
All of a), b) and c) should no longer be problematic in this case. I found
that on Debian and Gentoo Linux this is already the case. To make this work a
udev rule needs to be packaged that passes ownership of /dev/tpm0 device to
the tss user. To prevent regressions when switching from the privilege drop
approach to this new approach, a possibly already existing
/var/lib/tpm/system.auth file needs to be safely chown()'ed to the tss user
during package updates.

On SUSE and Fedora Linux the tcsd is started as root via systemd, thus they
are affected by the security issues. A preliminary suggested source code fix
is attached to this mail. It makes sure that `O_NOFOLLOW` is added to step 3)
to prevent a symlink attack. It also adds a drop of the root gid to the tss
gid. And it modifies the check of /etc/tcsd.conf such that ownership root:tss
and mode 0640 are necessary. The packaging needs to be adjusted accordingly.

The correct long term fix should probably be to *only* open /dev/tpm0 as root,
immediately drop to tss:tss and only then perform the further initialization
steps. The initialization sequence in `tcsd_startup()` is currently running
completely in the root user context and seems rather complex. Maybe there are
more details to this that I don't know of yet. For this reason I didn't try a
patch in this direction yet.

Upstream Reporting
==================

I reported issues a), b) and d) privately to the documented upstream contacts
without much success (see Timeline below). The SUSE Security Team 90 days
maximum disclosure time has been reached, therefore I'm publishing this now in
an uncoordinated way. While working on a fix I additionally discovered issue
c). SUSE is tracking the issues in bsc#1164472 [6] currently.

Issues a), b) and c) deserve CVE assignments in my opinion. I can't request
CVEs myself though, because IBM upstream is a CNA themselves. Therefore
upstream is required to assign their own CVEs.

Timeline
========

2020-02-19: I reported findings a), b) and d) to ho...@li...,
            the security contact of the project according to the README file [2].
2020-02-28: I reported findings a), b) and d) to de...@li..., the
            maintainer of the project according to the AUTHORS file [3].
2020-03-16: I received a reply from de...@li..., stating that she
            will look into the findings.
2020-05-06: I reminded de...@li... that the latest disclosure time
            [4] for the findings is approaching and asked for any updates.
2020-05-20: I started working on a bugfix and mitigations, discovered the
            additional finding c) and started publishing the findings.

[1]: https://sourceforge.net/projects/trousers
[2]: https://sourceforge.net/p/trousers/trousers/ci/master/tree/README
[3]: https://sourceforge.net/p/trousers/trousers/ci/master/tree/AUTHORS
[4]: https://en.opensuse.org/openSUSE:Security_disclosure_policy
[5]: https://sourceforge.net/p/trousers/trousers/ci/master/tree/dist/trousers.spec.in
[6]: https://bugzilla.suse.com/show_bug.cgi?id=1164472

Best Regards

Matthias

--
Matthias Gerstner <matth...@su...>
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
```

View entire thread

Top Downloaded Projects

Company

About

Team

SourceForge Headquarters

225 Broadway Suite 1600

San Diego, CA 92101

+1 (858) 454-5900

Resources

Support

Site Documentation

Site Status