CVE-2020-7221: mariadb: possible local mysql to root user exploit in mysql_install_db script setting permissions of /usr/lib64/mysql/plugin/auth_pam_tool_dir/auth_pam_tool

From: Matthias Gerstner <mgerstner () suse de>
Date: Tue, 4 Feb 2020 11:26:04 +0100

Hello list,

in the course of a review of a newly added setuid-root binary
(auth_pam_tool) in recent mariadb releases I discovered a local mysql
user to root privilege escalation.

The issue stems from the mysql_install_db script where the following
lines are found in mariadb releases ranging from 10.4.7 up and including
to 10.4.11:

```
    if test -n "$user"
    then
      chown $user "$pamtooldir/auth_pam_tool_dir" && \
      chmod 0700 "$pamtooldir/auth_pam_tool_dir"
      if test $? -ne 0
      then
          echo "Cannot change ownership of the '$pamtooldir/auth_pam_tool_dir' directory"
          echo " to the '$user' user. Check that you have the necessary permissions and try again."
          exit 1
      fi
      if test -z "$srcdir"
      then
        chown 0 "$pamtooldir/auth_pam_tool_dir/auth_pam_tool" && \
        chmod 04755 "$pamtooldir/auth_pam_tool_dir/auth_pam_tool"
        if test $? -ne 0
        then
            echo "Couldn't set an owner to '$pamtooldir/auth_pam_tool_dir/auth_pam_tool'."
            echo " It must be root, the PAM authentication plugin doesn't work otherwise.."
            echo
        fi
      fi
      args="$args --user=$user"
    fi
```

In a typical MariaDB installation where $user is set to the mysql user
this will perform the following sequence of commands as root:

```
    chown mysql /usr/lib64/mysql/plugin/auth_pam_tool_dir
    chmod 0700 /usr/lib64/mysql/plugin/auth_pam_tool_dir
    chown 0 /usr/lib64/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
    chmod 04755 /usr/lib64/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
```

These steps are executed unconditionally no matter what the current
owner and mode of the auth_pam_tool_dir are. If the mysql account is
compromised then an attacker can prepare a symlink attack or simply
place an arbitrary binary in auth_pam_tool_dir/auth_pam_tool which will
gain setuid-root privileges once mysql_install_db is run. This way the
mysql user can gain full root privileges easily.

The mysql_install_db script can be invoked automatically, depending on
the actual integration into a Linux distribution, e.g. during RPM
installation time or during systemd service start time. It can also be
invoked interactively by an Administrator (it is placed in /usr/bin).

Upstream decided to fix [1] this issue by only executing the commands in
question when the `--rpm` command line parameter is *not* passed. Thus
in typical package manager integrations the vulnerability hopefully
doesn't show any more by default. It will still occur when
Administrators interactively run the command without the `--rpm` switch.
The rationale behind this is support for users that extract tarballs
manually (probably without correctly preserving permissions) to install
MariaDB.

For Deb/RPM packaging MariaDB continues to suggest to use the following
dir and file modes [2], [3]:

```
mysql:root  0700 /usr/lib/mysql/plugin/auth_pam_tool_dir
 root:root 04755 /usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
```

I personally suggest the following directory mode instead:

```
root:mysql  0750 /usr/lib/mysql/plugin/auth_pam_tool_dir
```

This way the hardening is still intact (i.e. the setuid-root binary is
not publically available to users in the system, but only to members of
the mysql group) while the dangerous situation of a setuid-root binary
residing in a directory owned by an unprivileged user is avoided. The
latter situation can easily lead to race conditions e.g. when programs
try to replace the "auth_pam_tool" binary with a new version.

I also recommend a patch of the mysql_install_db script towards this
directory mode, to make the default behaviour of the script more secure.

Cheers

Matthias

Timeline

2020-01-14: I privately reported the issue at security () mariadb org.
2020-01-14: Upstream replied and confirmed the issue. They asked me to
  wait until the next release of MariaDB before publication of the
  issue.
2020-01-16: I attempted a deeper technical discussion with upstream
  about an appropriate fix, but it died down. I shared a CVE for use
  with this issue with upstream.
2020-01-28: MariaDB 10.4.12 got released, containing an attempted fix
  for the issue. I was not informed about the publication by upstream.

References
----

[1]: https://github.com/MariaDB/server/commit/9d18b6246755472c8324bf3e20e234e08ac45618
[2]: https://github.com/MariaDB/server/blob/mariadb-10.4.12/debian/rules#L151
[3]: https://github.com/MariaDB/server/blob/mariadb-10.4.12/plugin/auth_pam/CMakeLists.txt#L20

[4]: https://bugzilla.suse.com/show_bug.cgi?id=1160868

--
Matthias Gerstner <matthias.gerstner () suse 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

Attachment: signature.asc
Description:
