~~Bug 1160868 - VUL-0: CVE-2020-7221: mariadb: auth_pam_tool runtime permission setting allows privilege escalation from mysql user to root~~

Status:	RESOLVED FIXED
Classification:	Novell Products
Product:	SUSE Security Incidents
Component:	Incidents
Version:	unspecified
Hardware:	Other Other
Priority:	P3 - Medium **Severity**: Normal
Target Milestone:	---
Assigned To:	Kristyna Streitova
QA Contact:	Security Team bot
URL:	https://smash.suse.de/issue/250855/
Whiteboard:	
Keywords:	
Depends on:	
Blocks:	~~1160285~~

Show dependency tree / graph

- Create test case
- Clone This Bug

Reported:	2020-01-14 09:42 UTC by Matthias Gerstner
Modified:	2020-02-21 16:05 UTC (History)
CC List:	4 users (show)
See Also:	
Found By:	---
Services Priority:	
Business Priority:	
Blocker:	---

Attachments

disarm mysql_install_db default behaviour (1.81 KB, patch) Details | Diff
2020-02-04 10:56 UTC, Matthias Gerstner

Add an attachment (proposed patch, testcase, etc.) View All

Note
You need to log in before you can comment on or make changes to this bug.

Matthias Gerstner 2020-01-14 09:42:24 UTC Description

I've recently sent a security report to MariaDBs security contact about a local mysql to root user privilege escalation. This is related to the auth_pam_tool setuid binary which I audited in ~~bug 1160285~~. The report goes as follows:

```
> Hello MariaDB security team,
>
> in the course of a security audit of the new "auth_pam_tool" setuid root
> binary in MariaDB I have found a potential security issue. I have been
> looking into MariaDB version 10.4.10 for this purpose.
>
> The issue is a possible local mysql to root user privilege escalation.
> It stems from the mysql_install_db script where the following lines are
> found:
>
> ```
> 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' direct
>   echo " to the '$user' user. Check that you have the necessary permissions a
>   exit 1
>   fi
>   if test -z "$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_tc
>   echo " It must be root, the PAM authentication plugin doesn't work otherw
>   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 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.
>
> My recommendations to make this secure are as follows:
>
> - let the directory auth_pam_tool_dir be owned by root:root, mode 755.
>
```

> - let the setuid-binary auth_pam_tool be owned by root:mysql, mode 750.
>
> - don't actively modify file system permissions during DB setup, only
>   inspect the permissions and warn or fail if they're not correct. Let
>   instead the OS packaging take care of correct permissions. Or use
>   systemd-tmpfiles which is able to adjust permissions in a safe manner.
>   mariadb already ships a sytemd-tmpfiles configuration file that could
>   be used for this purpose.
>
> This way only members of the mysql group can access the setuid-root
> binary and the attack surface is lowered.

Comment 1

This finding here also show sthat the permissions originally requested in audit ~~bug 1160285~~ conflict with the current upstream implementation. But since the upstream implementation is buggy we should wait for them to fix it and then decide what to do in our packaging.

---

**Matthias Gerstner** 2020-01-14 09:51:10 UTC Comment 2

Please note that this report here is still confidential and we can't publish any information about it before we've agreed with upstream about a publication date.

---

**Matthias Gerstner** 2020-01-14 10:05:54 UTC Comment 3

Info for reactive security: We didn't ship this setuid-binary yet in any product so we're actually not affected. But we should still track this since we found it and manage the embargo, and we need to avoid packaging this in its current form.

---

**Matthias Gerstner** 2020-01-16 12:53:22 UTC Comment 4

Upstream replied by now. They acknowledge the issue but the way of fixing it is a bit disputed. At least they will make a change that causes this chown logic to be disabled if mysql_install_db is called with the --rpm parameter. The logic is originally intended for people who use tarballs and mess up permissions.

There is no fixed CRD, they want us to keep this private until their next release (scheduled for the end of this month) is out. They will ping me when this happens.

---

**Matthias Gerstner** 2020-01-20 08:49:48 UTC Comment 5

Upstream sees no urgent need for a CVE and left it to mee to request one. So I requested one from Mitre.

---

**Matthias Gerstner** 2020-01-20 09:00:04 UTC Comment 6

This upstream fix [1] will prevent the chmod/chown logic to trigger in the RPM packaging context. We should either apply this patch or wait for the next MariaDB release which is supposed to be finished by end of this month.

[1]: https://github.com/MariaDB/server/commit/9aa1d516f135b299

---

**Kristyna Streitova** 2020-01-21 15:49:02 UTC Comment 7

I think we can wait till the next MariaDB release if they really finish it by the end of this month.

---

**Matthias Gerstner** 2020-02-03 11:40:18 UTC Comment 8

It seems the communication with upstream didn't work out so well. There's a 10.4.12 release [1] available now that changes something in this area [2]. They didn't even include the CVE number I pulled for them.

Basically this means that this bug here can be published an a posting on oss-sec is probably in order.

[1]: https://mariadb.com/kb/en/mariadb-10412-changelog/
[2]: https://github.com/MariaDB/server/commit/9d18b62467

---

**Matthias Gerstner** 2020-02-04 10:30:28 UTC Comment 11

I just posted an email describing this vulnerability to the oss-sec mailing list.

The "fix" in upstream release 10.4.12 looks incomplete to me. They now leave the permissions of auth_pam_tool_dir and auth_pam_tool alone when mysql_install_db is invoked with the `--rpm` switch. However they still propagate the directory ownership and mode "mysql:root 0700" for packagers. Also when mysql_install_db is invoked without `--rpm` then the issue still occurs, which can easily happen when an Administrator runs it manually.

I agreed with kstreitova that we can accept this upstream release 10.4.12 when our packaging uses ownership and mode "root:mysql 07500" instead. Now that I'm thinking about it I also like to add another patch for the mysql_install_db script to sanitize the default behaviour of the script. I will share the patch soon.

---

**Matthias Gerstner** 2020-02-04 10:50:11 UTC Comment 12

oss-sec posting can be found here: https://seclists.org/oss-sec/2020/q1/55

---

**Matthias Gerstner** 2020-02-04 10:56:07 UTC Comment 13

Created attachment 829037 [details]

disarm mysql_install_db default behaviour

---

**Matthias Gerstner** 2020-02-04 11:01:58 UTC Comment 14

The patch in attachment 829037 [details] will remove the dangerous behaviour of mysql_install_db. I want it to be added in our packaging. We don't expect our customers to extract tarballs manually in the system, thus we won't ever benefit from those dangerous chown/chmod calls. Instead my patch changes the code to just warn the caller if the mode is not the expected one.

@kstreitova: I've submitted sr#769946 towards your home project to add this patch and also fixes the name of the other patch.

---

**Matthias Gerstner** 2020-02-06 08:43:17 UTC Comment 16

From the security side this bug is finished. Since we never shipped this vulnerability we don't need to track anything for older codestreams. The patches I asked for are present in Factory and have been submitted to SLE-15-SP2. So when you think you're done you may close the bug.

---

**Kristyna Streitova** 2020-02-07 18:02:12 UTC Comment 18

| Codestream | Request |
|---|---|
| openSUSE:Factory | #772117 |
| SLE15SP2 | #210871 |

I'm closing it as fixed.

---