~~Bug 1181050 - (CVE-2021-32000) VUL-0: CVE-2021-32000: clone-master-clean-up: potentially dangerous file system operations in clone-master-clean-up.sh~~

- Create test case
- Clone This Bug

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:	Security Team bot
QA Contact:	Security Team bot
URL:	https://smash.suse.de/issue/275734/
Whiteboard:	CVSSv3.1:SUSE:CVE-2021-32000:5.0:(AV:...
Keywords:	
Depends on:	
Blocks:	
	Show dependency tree / graph

Reported:	2021-01-18 09:58 UTC by Matthias Gerstner
Modified:	2022-11-29 13:39 UTC (History)
CC List:	7 users (show)
See Also:	
Found By:	---
Services Priority:	
Business Priority:	
Blocker:	---

Attachments

supportconfig from virtualmachine (825.45 KB, application/x-xz-compressed-tar) Details
2022-09-09 13:07 UTC, Ednilson Miura

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 2021-01-18 09:58:36 UTC   Description

~~~
Looking at openSUSE:Factory/clone-master-clean-up/clone-master-clean-up.sh I
stumbled upon potentially dangerous file system operations.

line 49:

```
rm -rf /etc/ssh/ssh_host*key* /root/.ssh/* /home/*/.ssh/* /home/*/.*_history &>
/dev/null
```

In any unprivileged user with a home directory places a symlink like:

    /home/evil/.ssh -> /

then the whole file system would be deleted.

Some other locations are owned by unprivileged accounts like:

line 56:

```
rm -rf
/var/spool/postfix/{active,corrupt,deferred,hold,maildrop,saved,bounce,defer,flush,ir
```

These directories are owned by the 'postfix' user. If compromised this could
also be used as an attacker vector.

Also some log files or log directories are owned by non-privileged accounts,
so line 62:

```
find /var/log -type f -exec truncate -s 0 {} \;
```

could also possibly follow symlinks, when race conditions are exploited.

I realize that a partially compromised system being used as a master clone
image is a problem in the first place. But maybe some of these operations can
be made more security aware anyway.
~~~

Egbert Eich 2021-01-18 10:43:58 UTC   Comment 1

```
Before we play a guessing game again, any suggestion how to do this securely?
```

Egbert Eich 2021-01-18 11:45:29 UTC   Comment 2

````
(In reply to Matthias Gerstner from comment #0)

> Looking at openSUSE:Factory/clone-master-clean-up/clone-master-clean-up.sh I
> stumbled upon potentially dangerous file system operations.
>
> line 49:
>
> ```
````

```
> rm -rf /etc/ssh/ssh_host*key* /root/.ssh/* /home/*/.ssh/* /home/*/.*_history
> &> /dev/null
> ```
>
> In any unprivileged user with a home directory places a symlink like:
>
>   /home/evil/.ssh -> /
>
> then the whole file system would be deleted.
```

Removing anything under /home is probably a bad idea anyway:
A system with active homes should probably not be cloned. If it is done anyway it is probably not good to remove people's ssh keys.
The only valid use case would be to clone a freshly installed system on which the installation has created a single user. This user would not have any ssh keys set.
Maybe we should check for a populated /home and warn the user.

Egbert Eich 2021-01-18 12:25:47 UTC Comment 3

(In reply to Matthias Gerstner from comment #0)

```
> rm -rf
> /var/spool/postfix/{active,corrupt,deferred,hold,maildrop,saved,bounce,defer,
> flush,incoming,trace}/*
```

How about:

```
for i in
/var/spool/postfix/{active,corrupt,deferred,hold,maildrop,saved,bounce,defer,flush,ir
do
   # descend following symlink and check if it was symlink, if not, recursively
delete entries in this directory. 'rm -rf' doesn't follow symlinks.
   cd -P $i
   [ "$i" = "${pwd}" ] && find -maxdepth 1 ! -name . -print0 | xargs -0 rm -rf
done
```

Matthias Gerstner 2021-01-19 10:58:07 UTC Comment 4

(In reply to Egbert Eich from comment #3)

```
> (In reply to Matthias Gerstner from comment #0)
>
> > rm -rf
> > /var/spool/postfix/{active,corrupt,deferred,hold,maildrop,saved,bounce,defer,
> > flush,incoming,trace}/*
>
> How about:
>
> for i in
> /var/spool/postfix/{active,corrupt,deferred,hold,maildrop,saved,bounce,defer,
> flush,incoming,trace}; do
>    # descend following symlink and check if it was symlink, if not,
> recursively delete entries in this directory. 'rm -rf' doesn't follow
> symlinks.
>    cd -P $i
>    [ "$i" = "${pwd}" ] && find -maxdepth 1 ! -name . -print0 | xargs -0 rm
> -rf
> done
```

This goes in the right direction. Maybe replacing the `find` pipeline by a `rm -rf *` would be even better. `rm` should be safe against symlink race conditions in its recursion logic. It only is a problem if a command line argument is already a symlink like `rm somelink/*`.

The most orubst approach actually would be using `sudo`, `su` or `setpriv` to change to the uid/gid that owns the directory tree.

Egbert Eich 2021-01-22 12:08:57 UTC Comment 5

Ok, thanks!

Regarding:

```
> Also some log files or log directories are owned by non-privileged accounts,
> so line 62:
>
> ```
> find /var/log -type f -exec truncate -s 0 {} \;
> ```
```

I'd suggest something like this:

```
while IFS= read -r -d $'\0' file; do
   d=$(dirname $file);
   u=$(stat --printf="%u" %d);
   f=$(basename $file);
   t=$(mktemp -d $d/tmp-XXXXXXXXX) || continue;
   mv $file $t
   if ! test -h $t/$f || ! setpriv --ruid $u truncate -s 0 $t/$f ; then
      setpriv --ruid $u rm $t/$f && setpriv --ruid $u touch $t/$f
   fi
   mv $t/$f $file
   rmdir $t
done < <(find /var/log -type f -print0)
```

Egbert Eich 2021-01-22 12:09:43 UTC Comment 6

Setting to needinfo for Matthias - see above comment.

Matthias Gerstner 2021-01-25 12:10:13 UTC Comment 7

(In reply to Egbert Eich from comment #5)

```
> I'd suggest something like this:
> while IFS= read -r -d $'\0' file; do
>    d=$(dirname $file);
>    u=$(stat --printf="%u" %d);
>    f=$(basename $file);
>    t=$(mktemp -d $d/tmp-XXXXXXXXX) || continue;
```

```
>    mv $file $t
>    if ! test -h $t/$f || ! setpriv --ruid $u truncate -s 0 $t/$f ; then
>       setpriv --ruid $u rm $t/$f && setpriv --ruid $u touch $t/$f
>    fi
>    mv $t/$f $file
>    rmdir $t
> done < <(find /var/log -type f -print0)
```

Hmm I find the approach with the temporary directory a bit confusing. Also if more deeply nested directories exist (like /var/log/nginx/some-sub-dir) then this could still have issues.

Doing this in bash is not pretty, sadly. How about that:

~~~
```
while IFS= read -r -d $'\0' dir; do
        cd -P "$dir"
        # not the expected directory (symlink involved?), skip over this
        [ "$PWD" != "$dir" ] && continue
        info=( $(stat --printf="%u %g 0%a" ".") )
        owner=${info[0]}
        group=${info[1]}
        mode=${info[2]}

        rmuid=0
        rmgid=0

        if (( "$group" != 0 && ($mode & 0020) != 0 )); then
                rmuid=nobody
                rmgid=$group
        elif (( "$owner" != 0 )); then
                rmuid=$owner
                rmgid=nobody
        fi

        setpriv --clear-groups --reuid "$rmuid" --regid "$rmgid" find -maxdepth 1 -type f -print
done < <(find /var/log -type d -print0)
```
~~~

Egbert Eich 2021-03-26 09:04:52 UTC Comment 8

@Matthias, I don't think we are getting any closer to the goal ie securely truncating log files: if I'm not totally mistaken, the suggestion in comment #7 is doing nothing (it runs a find, but what does this do except printing output?). Maybe, you've intended to run 'truncate' instead, however, I doubt that this will work on directory permissions if the file ownership/permission does not allow writing.

This addresses the symlink race condition issue only indirectly by allowing it to act only on files which have at least group modify rights for the group of the directory or owner modify rights for the owner, thus, it cannot affect anything owned by root.

The subdir mechanism suggested in comment #6 is to prevent any symlink race by virtue of the discussion in bsc#1155075.
All this seems to be overkill as a system prepared for cloning should be 'well defined' ie the sysadmin needs to be sufficiently sure that
a. no users - neither legitimate nor malicious are logged in
b. no unwanted processes and services are running (that would exploit a
   symlink race).
If the sysadmin cannot rule out these with reasonable certainty, the issue is much larger than just a few truncated files that are system relevant.

I fear that the attempt to make this script more secure will cause it to do less.

Matthias Gerstner 2021-03-29 08:52:01 UTC Comment 9

(In reply to Egbert Eich from comment #8)

> @Matthias, I don't think we are getting any closer to the goal ie securely
> truncating log files: if I'm not totally mistaken, the suggestion in comment
> #7 is doing nothing (it runs a find, but what does this do except printing
> output?).

Probably remains of my testing, the `setpriv` call should of course perform some kind of deletion/truncation.

> Maybe, you've intended to run 'truncate' instead, however, I doubt that this
> will work on directory permissions if the file ownership/permission does not
> allow writing.

Why should there be a logfile that doesn't allow writing by the user that has write permission on the containing directory? Such cases would be suspicious, security wise, I guess.

> This addresses the symlink race condition issue only indirectly by allowing
> it to act only on files which have at least group modify rights for the
> group of the directory or owner modify rights for the owner, thus, it cannot
> affect anything owned by root.

You mean when a directory has write-permission for somebody who is not root but it contains a logfile writable only by root?

> The subdir mechanism suggested in comment #6 is to prevent any symlink race
> by virtue of the discussion in bsc#1155075.

If the temporary directory would be in a safe place then this would be fine, but putting it below potentially non-root controlled directories can be an issue again, especially when more deeply nested like I mentioned in comment 7.

> All this seems to be overkill as a system prepared for cloning should be
> 'well defined' ie the sysadmin needs to be sufficiently sure that
> a. no users - neither legitimate nor malicious are logged in
> b. no unwanted processes and services are running (that would exploit a
>   symlink race).
> If the sysadmin cannot rule out these with reasonable certainty, the issue
> is much larger than just a few truncated files that are system relevant.
>
> I fear that the attempt to make this script more secure will cause it to do
> less.

Yes the complexity is unfortunate and I know it is annoying. Still something like the .ssh symlink can be prepared long before any sysadmin runs this cleanup script, so it is not just a question about whether at the time of execution of the cleanup script any evil players are around in the system. So this /home bit should be addressed in any case like you already stated in comment 2.

Symlinks in /var can in theory also be prepared ahead of time by compromised services. A discussion about whether this is realistic or not is tedious. Cloning a partially compromised system is always problematic, of course. On the other hand, when code like this accumulates then at some point we will have actual problems again. Another question is if something could go wrong by accident in the future e.g. when a service deliberately places a symlink in its /var directory.

If the precondition for this tool is that the admin is aware and the system is safe and sound then maybe it would help if the admin makes an informed decision e.g. by the tool printing a security note and a list of actions that will be executed when continuing.

Johannes Segitz 2021-07-05 11:07:30 UTC Comment 15

Please use CVE-2021-32000

Carlos López 2022-08-29 12:02:35 UTC Comment 17

According to our tracking, this was never fixed for:
- SUSE:SLE-12-SP3:Update
- SUSE:SLE-15:Update
- SUSE:SLE-15-SP1:Update

Assigning to the SLE maintainers.

Angela Briel 2022-08-31 09:03:22 UTC Comment 18

As discussed in our meeting today assigned to you, Peter.
The changes from Egbert and me are available in a PR (https://github.com/SUSE/clone-master-clean-up/pull/10) with review requested for you.
Thanks for taking over.

Angela Briel 2022-08-31 10:10:41 UTC Comment 19

After some additional discussion with Matthias Gerstner it turns out, that we can ignore the request for changes of 'find /var/log -type f'.
It seems that the newer versions of the 'find' command are secure against 'symlink' attacks.
Many thanks for the help on this special part.

Ednilson Miura 2022-09-09 13:05:02 UTC Comment 22

While testing S:M:25847:279301 (clone-master-clean-up - SLE15SP3/4), apparently there is a problem with btrfs snapshots:

SETUP:
SLE15SP3 under kvm/virt-manager.
Snapshot taken, so all modifications are reverted after first run and after updating the package running under the same ambient.

```
BEFORE update:
# clone-master-clean-up
The script will delete all SSH keys, log data, and more. Type YES and enter to
proceed.
YES
Wiping active swap devices/files (this may take a while)
Turning off swap device/file /dev/vda3 (UUID d8e520af-523b-44bb-8db8-d4d5263dddff7)
Zero-overwriting /dev/vda3...
Setting up swapspace version 1, size = 2 GiB (2147459072 bytes)
no label, UUID=d8e520af-523b-44bb-8db8-d4d5263dddff7
Removing system registration information and zypper repositories
Removing zypper anonymous ID
Removing SSH host keys, user SSH keys, authorized keys, and shell history
Removing all mails and cron-jobs
Clean up postfix
Removing all temporary files
Clearing log files and removing log archives
Clearing HANA firewall script
Removing random seeds
Clearing systemd journal
Clearing machine ID file
Removing Salt client ID
Removing osad authentication configuration file and the system ID
Removing domain name and set host name from DHCP in network config
Removing persistent network interfaces
Restoring initial system-wide network config
Enabling YaST Firstboot if necessary
Removing all pre/post btrfs snapshots from /.snapshot
Would you like to give root user a new password? Type YES to set a new password,
otherwise simply press Enter.

swap the uuid strings with dev strings in /etc/fstab
Clean up network files (except interfaces using dhcp boot protocol)
Clean up persistent network data
Clean up collectd
Clean up /root
Clean up cache, crash and coredump
Finished. The system is now sparkling clean. Feel free to shut it down and image
it.

AFTER updating package:
# clone-master-clean-up_
The script will delete root SSH keys, log data, and more.
 WARNING: This should only be used on a pristine system
 WARNING: with no populated /home directories!
 Type YES and enter to proceed.
YES
There seem to be populated /home directories on this system
 Cloning such systems is not recommended.
 Type YES if you still would like to proceed.
YES
Wiping active swap devices/files (this may take a while)
Turning off swap device/file /dev/vda3 (UUID d8e520af-523b-44bb-8db8-d4d5263dddff7)
Zero-overwriting /dev/vda3...
Setting up swapspace version 1, size = 2 GiB (2147459072 bytes)
no label, UUID=d8e520af-523b-44bb-8db8-d4d5263dddff7
Removing system registration information and zypper repositories
Removing zypper anonymous ID
Removing SSH host keys, root user SSH keys, authorized keys, and shell history
Removing all mails and cron-jobs
```

```
Clean up postfix
Removing all temporary files
Removing log archives
Clearing log files
Clearing HANA firewall script
Removing random seeds
Clearing systemd journal
Clearing machine ID file
Removing Salt client ID
Removing osad authentication configuration file and the system ID
Removing domain name and set host name from DHCP in network config
Removing persistent network interfaces
Restoring initial system-wide network config
Enabling YaST Firstboot if necessary
Removing all pre/post btrfs snapshots from /.snapshot
Snapshot '34' not found.
Sorry! An error occured on line 182, the clean up routine did not complete
successfully.
```

Ednilson Miura 2022-09-09 13:07:40 UTC Comment 23

Created attachment 861391 [details]
supportconfig from virtualmachine

Peter Varkoly 2022-09-12 09:16:53 UTC Comment 24

Please provide me the output of:

```
dbus-send --type=method_call --system --print-reply --dest=org.opensuse.Snapper
/org/opensuse/Snapper org.opensuse.Snapper.ListSnapshots string:root
2>/dev/null
```

Peter Varkoly 2022-09-12 09:19:51 UTC Comment 25

IMHO, this is a completely different bug. It has nothing to do with the original.
It might make sense to open a new bug for it.

Peter Varkoly 2022-09-22 09:07:49 UTC Comment 26

I've opened a new ticket for the snapshot removing problem:
~~https://bugzilla.suse.com/show_bug.cgi?id=1203651~~

The original issue should be fixed.

Swamp Workflow Management 2022-10-20 01:22:22 UTC Comment 29

SUSE-SU-2022:3667-1: An update that solves one vulnerability and has one errata is now available.

```
Category: security (moderate)
Bug References: 1181050,1203651
CVE References: CVE-2021-32000
JIRA References:
Sources used:
openSUSE Leap 15.4 (src):    clone-master-clean-up-1.8-150100.3.14.1
openSUSE Leap 15.3 (src):    clone-master-clean-up-1.8-150100.3.14.1
SUSE Linux Enterprise Module for Server Applications 15-SP4 (src):    clone-master-clean-up-1.8-150100.3.14.1
SUSE Linux Enterprise Module for Server Applications 15-SP3 (src):    clone-master-clean-up-1.8-150100.3.14.1
```

NOTE: This line indicates an update has been released for the listed product(s). At times this might be only a partial fix. If you have questions please reach out to maintenance coordination.

Swamp Workflow Management 2022-10-20 16:25:05 UTC Comment 30

SUSE-SU-2022:3674-1: An update that solves one vulnerability and has one errata is now available.

```
Category: security (moderate)
Bug References: 1181050,1203651
CVE References: CVE-2021-32000
JIRA References:
Sources used:
SUSE Linux Enterprise Server 12-SP5 (src):    clone-master-clean-up-1.8-4.11.1
```

NOTE: This line indicates an update has been released for the listed product(s). At times this might be only a partial fix. If you have questions please reach out to maintenance coordination.