Starred by 4 users

Owner:	mortonm@chromium.org
CC:	jhorwich@chromium.org sarth...@chromium.org skau@chromium.org dlunev@chromium.org ereth@chromium.org achuith@chromium.org heejunglee@google.com
Status:	Fixed *(Closed)*
Components:	OS>Security
Modified:	May 14, 2021
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Chrome
Pri:	0
Type:	Bug-Security

Security_Impact-Stable
Security_Severity-High
allpublic
reward-inprocess
Target-88
M-88
reward-45000
LTR-Merged-86
LTS-Security-86
external_security_report

BlockedOn:	Issue 1168541 ~~Issue 1168544~~ ~~Issue 1168545~~ Issue 1168549 ~~Issue 1168550~~ Issue 1168551 ~~Issue 1168552~~ Issue 1168553 ~~Issue 1168555~~ View details

Issue 1166932: Security: ChromeOS root privilege escalation and android-root persistence

Reported by rory@rorym.cnamara.com on Fri, Jan 15, 2021, 1:06 AM GMT+1

Code

VULNERABILITY DETAILS
Two race conditions in cryptohome and one in crash-reporter can allow the chronos user to take ownership of any file. This can be used to disable SELinux.

The Arc mediadrm component can be exploited by a command injection via a maliciously planted directory to achieve command execution as media in Arc. With SELinux disabled, the media user can set arbitrary android properties and restart the zygote service to obtain android-root command execution.

The arc-bugreport binary can then be replaced and combined with a controlled symlink to obtain real root file write via BackupArcBugReport. This can be used to obtain uid=0 command execution.

Arc-sdcard can then be restarted, and caused to perform a mount through a symlink to an arbitrary directory. This can be used to mount over /etc/init and obtain full root command execution.

The send_metrics_on_resume script contains a partial argument injection vulnerability which can be used to re-disable SELinux on boot, allowing for persistent android-root command execution by re-exploiting mediadrm.

VERSION
Google Chrome: 87.0.4280.109 (Official Build) (64-bit)
Revision: 04469fd2154e3506101aac1bf7f9d2565cfc74ae-refs/branch-heads/4280@{#1852}
Platform: 13505.73.0 (Official Build) stable-channel eve
Firmware Version: Google_Eve.9584.201.0

REPRODUCTION CASE
Metrics consent must be enabled ("Help improve Chrome OS features and performance" in settings). Whilst possible to do as part of the exploit script, this step is complicated so has been left out for simplicity.

If the termina cros-component has been downloaded (ie, termina has been started before), the exploit script will execute about 3x faster. Termina itself is not used as part of the exploit (See sidebar below). The script does not try to download termina itself and will fall back to a slower method if necessary.

Execute the attached privesc.sh script from a chronos shell:

```
sh <(cat privesc.sh)
```

After approximately two minutes, a root shell should be provided. After a reboot, it can be observed that the process 'sleep 1337' is executing as android-root. This is the result of the persistence (there's no ssh daemon in the container so this is just a proof of concept, there are no restrictions on what can be executed in the android container).

DETAILS
=== cryptohome chronos-access chgrp ===
When mounting a cryptohome, the following (abridged) steps are performed:
1. mount /home/user/<hash>, /home/root/<hash>, /home/chronos/u-<hash>
2. Call MountHelper::MigrateDirectory[1] to move files from MyFiles/Downloads to Downloads
3. Bind mount Download to MyFiles/Downloads
4. Create, set permissions and bind mount various /home/root/<hash>/ daemon-store directories

Given that the cryptohome hash can be calculated in advance, it is possible to spin on the visibility of /home/chronos/u-<hash>/Downloads (after step 1) and modify the directory contents before step 2. A symbolic link can be dropped into MyFiles/Downloads pointing to drivefs before step 2. Drivefs is running as chronos, and therefore can be stopped with SIGSTOP. This causes all directory accesses to hang.

Assuming the race is won, the start of the MigrateDirectory function can be hung waiting to access the now-symlinked file. While cryptohomed is hung, both MyFiles/Downloads and Downloads can be replaced with symlinks to source and destination directories. If a file exists in the new source directory which matches the name of the symbolic link which causes the hang (controllable), cryptohomed will attempt to rename(2) the file to the destination directory[2]. Since cryptohome directly calls rename(2), cross device renames are not possible, but this constitutes a mount-constrained arbitrary file/directory rename.

Drivefs can then be sent SIGCONT which will cause cryptohomed to resume and process the rename.

This is used to copy a controlled 'chaps' directory from /home/.shadow/<hash>/mount/user to /home/.shadow/<hash>/mount/root.
/home/.shadow/<hash>/mount/user is the same directory as /home/chronos/u-<hash> and therefore writable, but /home/.shadow/<hash>/mount is used as both user and root are on the same mount allowing the call to rename(2) to succeed.

This is exploited such that /home/.shadow/<hash>/mount/root/chaps contains a targeted set of symbolic links. Cryptohomed will continue to step 3 and fail, since MyFiles/Downloads at this point is a symlink and mounting to symlinks is disabled by the LSM.

The mount process for the same user can then be restarted, not attempting to exploit the race condition. The symbolic links in Downloads and MyFiles/Downloads will be deleted and recreated as real directories, allowing the mount to proceed to step 4.

During initialization, Mount::CheckChapsDirectory is called which will iterate through /home/root/<hash>/chaps/ and set the permissions [3]. Since the chaps directory was moved over during the previous mount attempt and was fully controlled by chronos, the planted symlinks are traversed by the calls to Platform::SetOwnership and Platform::SetPermissions.

This results in arbitrary file group ownership being set to chronos-access. Permissions are also set to 0640 (readonly for chronos).

[1]
https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/cryptohome/storage/mount_helper.cc#566
[2]
https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/cryptohome/storage/mount_helper.cc#580
[3] https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/cryptohome/storage/mount.cc#468

=== crash-reporter chmod 660 ===
When processing a user crash, crash-reporter will execute core2md to create a minidump file from the provided coredump [1]. Core2md opens the minidump .dmp in /home/user/<hash>/crash, and upon completion crash-reporter will set the permissions to 660. Both processes use the crash directory relative to a safely opened proc directory fd, but the directory is still normally accessible by chronos.

Chronos can trigger a process to crash, predict the filename (A combination of date, time and pid), wait for the file to be created and swap it for a symlink while core2md runs. Once core2md is finished, crash-reporter will chmod the symlink target [2].

Combined with the above, chronos can get write permissions on arbitrary files. This is used against /sys/fs/selinux/enforce, /dev/loop-control and /dev/loop*

Sidebar: Termina
The next cryptohome race requires write access to loop devices. When /dev/loop-control is writable, single /dev/loopX devices can be created (using losetup -f).

By making lots of copies of termina (actually, symlinks in the cros-components directory), ImageLoader can be requested to mount lots of images. The images aren't modified so they mount without issue. If a large number of copies are created, mounted all at once, then unmounted, the kernel will be forced to create the same number of /dev/loopX devices, which aren't deleted when they're no longer used.

Of the two above vulnerabilities, the first is relatively slow to run but can be used to chgrp as many files as we like in one shot. So if termina is present, the exploit script will pre-populate a lot of loop devices and chgrp before they're needed in one shot. This saves substantial time compared to running the initial chgrp every time we fail the race in the next race exploit.

[1] https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/crash-reporter/user_collector.cc#287
[2] https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/crash-reporter/user_collector.cc#317

=== Arbitrary code execution ===
By exploiting a type confusion in ghostscript [1], it's possible to execute arbitrary shellcode. This is used to create a memfd to allow for the execution of an elf binary via /proc/pid/fd/[memfd], used a few times throughout this exploit. Note that this is not the same vulnerability as I used in 1077531.

[1] https://git.ghostscript.com/?p=ghostpdl.git;a=blob;f=psi/interp.c;h=730ddf16d06b70946d8e36dd31cad6c211891a62;hb=HEAD#l673

=== Cryptohome chown chronos ===
Ephemeral mounts are based on loop devices. Using losetup -f we can predict what loop device the next ephemeral mount will end up on.

During creation, cryptohome will create the directories root and user in the loop device ext4 filesystem. Since the above vulnerabilities allow for write access to the underlying loop device, it's possible to spin on the creation of /user in the new filesystem and replace it with a symbolic link as soon as it's created. This is done with the debugfs library using the above code execution vulnerability.

The result is that during this mount, an arbitrary directory can be chowned to chronos. Unfortunately, since instrumenting cryptohome with strace affects the result of the race, I'm not 100% sure what the actual line that causes the vulnerability is, but I've identified at least two candidates [1][2].

This is used to chown /home/root/<hash> (for the current user).

(I realise as I write this that this isn't actually needed anymore, as I think the next step could be achieved with the arbitrary rename from the first vulnerability above. Enjoy the free vulnerability :))

[1]
https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/cryptohome/storage/mount_helper.cc#254
[2]
https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/cryptohome/storage/mount_helper.cc#283

=== mediadrm command injection ===
The vendor.move_data_sh service in the Arc container executes a shell script at /system/bin/move_widevine_data.sh which is vulnerable to parameter injection due to unsafe handling of directories inside /data/mediadrm/

I can't find representative sources for the relevant files online, so the files can be seen at the below paths in the android rootfs
Service: /vendor/etc/init/android.hardware.drm@1.1-service.widevine.rc
Executes: /system/bin/move_widevine_data.sh

The relevant parts of move_widevine_data.sh:

```
...
DEST_PATH="/data/vendor/mediadrm"
...
SRC_PATH="/data/mediadrm"
...
 for i in "$SRC_PATH/IDM"*; do
   dest_path=$DEST_PATH/"${i#$SRC_PATH/}"
...
     find $dest_path -print0 | while IFS= read -r -d '' file
```

...

/data in the Arc container is bind mounted from /home/root/<hash>/android-data. With ownership of /home/root/<hash> from the previous step, it is possible to place directories inside the SRC_PATH directory used in the script above. The directory name is passed unsafely to the find command. If a directory named 'IDM d -exec sh -c sleep${IFS}1337 {} +' is placed inside the /home/root/<hash>/android-data/mediadrm/ directory, it will exploit the find command due to control over the $dest_path variable, resulting in command execution as the 'media' user.

Chronos can create the appropriate exploit directory, and call SessionManager.StopArcInstance to trigger a restart of Arc and therefore exploitation.

Note to successfully exploit this vulnerability, SELinux must be disabled, which is achieved using the file write capabilities of the first and second vulnerabilities above to allow the use of the setenforce binary.

=== android-root privilege escalation ===
With SELinux disabled, the command execution obtained via the step above can change it's SELinux domain and call setprop (setprop still cares what the source domain is even if SELinux is disabled, but domain transitions are not controlled when it is disabled).

By setting the property 'dalvik.vm.extra-opts' and setting ctl.restart to 'zygote', the zygote service can be restarted and the arguments to the underlying dalvikvm binary controlled. /data is both writable and executable inside the Arc container, so a shell script can be written inside /data and passed to dalvikvm, such as -Xpatchoat:/data/payload.sh.

Upon restart of the zygote service, the patchoat binary is executed, which is now controlled via the passed option. This results in command execution as android-root. The real patchoat binary is called and the Arc container does not crash.

=== host root file write ===
When calling the BackupArcBugReport method on the debugd dbus interface, debugd will enter the Arc namespaces (including the mount namespace), execute /system/bin/arc-bugreport [1], and write the contents to a file in /run/daemon-store/debugd/<hash>.

Using the above chronos directory chown, it is possible to take ownership of this output directory, and using android-root it is possible to control what exists at /system/bin/arc-bugreport by mounting over it.

The file contents can be controlled by the malicious script at arc-bugreport, and the output file can be controlled by writing an appropriately named symlink in /run/daemon-store/debugd/<hash>.

This is used to write a uid_map for a chronos process in a new user namespace, giving this process uid=0 code execution (with no capabilities).

[1] https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/debugd/src/log_tool.cc#88
[2] https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/debugd/src/log_tool.cc#915

=== host root command execution ===
arc-sdcard is an upstart job which mounts the arc esdfs to /run/arc/sdcard/{default,read,write}/emulated. This job can be triggered via dbus with the above uid=0 code execution.

When performing the mount [1], the Mount function calls Realpath on the parameter to resolve it [2]. With the uid=0 code execution it is possible to move /run/arc out of the way, and re-create the required directory structure, replacing /run/arc/sdcard/default/emulated with a symbolic link to /etc/init. When the upstart job is triggered, this symlink will be resolved and the esdfs file system will be mounted over /etc/init. This directory can then be written to by the uid=0 code execution, and reloaded via initctl reload-configuration. A malicious upstart service can be installed and triggered to gain full root command execution with all capabilities.

This is used to start an ssh daemon.

[1] https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/arc/setup/arc_setup.cc#955
[2] https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/arc/setup/arc_setup_util.cc#206

=== android-root persistence ===
The send_metrics_on_resume shell script is executed on boot. The send_time_in_suspend function unsafely reads the contents of /var/spool/power_manager/root/suspend-to-ram-time [1]. The shell script is run under /bin/sh so this does not result in full command execution, however various local variables can be controlled. The time_in_suspend_name variable is passed unquoted to /usr/bin/metrics_client. By re-defining the local IFS variable, it is possible to set time_in_suspend_name such that when metrics_client is executed the variable expands to 3 arguments [2].

This is used to add the additional arguments -W /sys/fs/selinux/enforce to the metrics_client execution. This results in metrics_client attempting to write the metrics protobuf to the selinux configuration file. The third metrics_client argument expanded from time_in_suspend_name acts as padding to control the length of the resultant protobuf. The protobuf starts with a packed length field. Therefore, with enough padding the resulting protobuf output can be controlled to start with '\x30\x00\x00\x00'... When written to the selinux/enforce file, the kernel will read up to the null byte and interpret it as a number, in this case 0.

This will cause metrics_client to disable selinux directly, on boot. When the user logs in again, their Arc will auto-start, which will re-exploit the mediadrm script, which will also re-exploit to achieve android-root command execution. This can be observed on subsequent boots by looking in the output of ps to see 'sleep 1337' being executed as android-root each boot.

To ensure that suspend-to-ram-time is not overwritten as it normally is when the machine goes to sleep, it is set to immutable using the root command execution and chattr +i.

[1]
https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/power_manager/tools/send_metrics_on_resume#98
[2]
https://chromium.googlesource.com/chromiumos/platform2/+/refs/heads/main/power_manager/tools/send_metrics_on_resume#104

SCRIPT WALKTHROUGH
1. Exploit ghostscript to create the memfd, copy over the payload elf file and SIGSTOP ghostscript so we can keep using the memfd
2. Drop sshd configuration files for later
3. Use the first cryptohome exploit to get write access to /dev/loop-control and /sys/fs/selinux/enforce
4. Kick off the crash-reporter chmod 660 races for both files, in the background
5. Exploit the second cryptohome exploit to get ownership over /home/root/<hash> for the current user
5a. 5 will require /dev/loopN devices to be writable. If termina is downloaded, call ImageLoader.LoadComponentAtPath 8 times, followed by UnmountComponent 8 times. This will create 8 'free' loop devices. Pass these all to the first cryptohome exploit to chgrp at once, and kick off all of the crash-reporter chmod 660 for each, also in the background
5b. If termina isn't downloaded, the /dev/loopN chgrp chronos-access and chmod 660 exploits will be done one at a time
6. Use setenforce to disable SELinux. This writes 0 to /sys/fs/selinux/enforce which should now be writable by chronos-access
7. Now /home/root/<hash> is owned by chronos from step 5, move android-data out of the way and re-create the directory structure with the mediadrm payload in /data/mediadrm. Also write the mediadrm and later android-root payload shell scripts to /data
8. Restart arc to trigger the mediadrm exploit
9. Mediadrm will be exploited, run /data/payload.sh which will call setprop to set dalvik.vm.extra-opts and restart zygote
10. Zygote will restart and call the now-malicious patchoat, which will mount a shell script over arc-bugreport, kick off a long sleep for communication, and run the real patchoat command
11. After 8 the main shell script will have been waiting for the sleep process to show up, indicating that arc-bugreport is now replaced
12. Take ownership of /run/daemon-store/debugd/<hash> for the current user. Place a symlink pointing to a target process' /proc/pid/uid_map. (This is actually in the elf binary as it will need to call setuid(geteuid()))
13. Trigger BackupArcBugReport from debugd. This will run arc-bugreport which is now a shellscript which will pick up the desired file contents from a file in /run/chrome (an easily accessible shared mount)
14. The target process should now have a uid_map giving it effective uid=0 on the host.
15. Move /run/arc out of the way, recreate the /run/arc/sdcard/{read,write,default}/emulated directory structure, and set /run/arc/sdcard/default/emulated to a symlink pointing at /etc/init
16. Trigger arc-sdcard over dbus, as we are effective uid=0

17. Write a custom upstart service to the newly mounted /etc/init
18. Trigger upstart to reload its configuration. Also allowed due to effective uid=0
19. Trigger the new upstart service
20. Sshd should now be running as root
21. Via ssh (as root), write the persistence payload to /var/spool/power_manager/root/suspend-to-ram-time, and set the file immutable
22. Via ssh (as root), repair /home/root/<hash>, chowned in step 5, to stop the persistence being lost
23. Present the user with a root shell

janwriteup.tar.bz2
529 KB Download

Comment 1 by sheriffbot on Fri, Jan 15, 2021, 1:11 AM GMT+1 Project Member

Labels: external_security_report

Comment 2 by xinghuilu@chromium.org on Fri, Jan 15, 2021, 8:11 PM GMT+1 Project Member

Labels: OS-Chrome

Comment 3 by jorgelo@chromium.org on Tue, Jan 19, 2021, 10:25 PM GMT+1 Project Member

Owner: mortonm@chromium.org
Labels: Pri-0
Components: OS>Security

Comment 4 by mortonm@chromium.org on Wed, Jan 20, 2021, 3:51 PM GMT+1 Project Member

Blockedon: 1168541

Comment 5 by mortonm@chromium.org on Wed, Jan 20, 2021, 3:52 PM GMT+1 Project Member

Blockedon: 1168544

Comment 6 by mortonm@chromium.org on Wed, Jan 20, 2021, 3:55 PM GMT+1 Project Member

Blockedon: 1168545

Comment 7 by mortonm@chromium.org on Wed, Jan 20, 2021, 4:04 PM GMT+1 Project Member

Blockedon: 1168549

Comment 8 by mortonm@chromium.org on Wed, Jan 20, 2021, 4:06 PM GMT+1 Project Member

Blockedon: 1168550

Comment 9 by mortonm@chromium.org on Wed, Jan 20, 2021, 4:08 PM GMT+1 Project Member

Blockedon: 1168551

Comment 10 by mortonm@chromium.org on Wed, Jan 20, 2021, 4:09 PM GMT+1 Project Member

Blockedon: 1168552

Comment 11 by mortonm@chromium.org on Wed, Jan 20, 2021, 4:10 PM GMT+1 Project Member

Blockedon: 1168553

Comment 12 by mortonm@chromium.org on Wed, Jan 20, 2021, 4:48 PM GMT+1 Project Member

Blockedon: 1168555

BlockedOn: 1168555

Comment 13 by mortonm@chromium.org on Wed, Jan 20, 2021, 6:06 PM GMT+1 Project Member

Cc: jhorwich@chromium.org

Comment 14 by jhorwich@chromium.org on Wed, Jan 20, 2021, 7:07 PM GMT+1 Project Member

Cc: ereth@chromium.org

Comment 15 by sheriffbot on Wed, Jan 20, 2021, 8:37 PM GMT+1 Project Member

Status: Assigned (was: Unconfirmed)

Comment 16 by mortonm@chromium.org on Wed, Jan 20, 2021, 8:41 PM GMT+1 Project Member

Cc: skau@chromium.org

Comment 17 by kerrnel@chromium.org on Wed, Jan 20, 2021, 8:58 PM GMT+1 Project Member

Cc: dlunev@chromium.org sarth...@chromium.org

Comment 18 by rory@rorym.cnamara.com on Wed, Jan 20, 2021, 9:10 PM GMT+1

Could I be CCd on the BlockedOn bugs please?

Comment 19 by jorgelo@chromium.org on Wed, Jan 20, 2021, 9:16 PM GMT+1 Project Member

Let's try to get this merged to 88 at the earliest.

Micah, can you set the Security_ labels too?

Comment 20 by mortonm@chromium.org on Wed, Jan 20, 2021, 9:17 PM GMT+1 Project Member

done

Comment 21 by mortonm@chromium.org on Wed, Jan 20, 2021, 9:47 PM GMT+1 Project Member

Labels: Security_Impact-Stable Security_Severity-High

Comment 22 by sheriffbot on Thu, Jan 21, 2021, 6:48 PM GMT+1 Project Member

Labels: Target-88 M-88

Setting milestone and target because of Security_Impact=Stable and high severity.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 23 by sheriffbot on Thu, Jan 21, 2021, 7:28 PM GMT+1 Project Member

Labels: -Pri-0 Pri-1

Setting Pri-1 to match security severity High. If this is incorrect, please reset the priority. Sheriffbot won't make this change again.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 24 by sheriffbot on Fri, Jan 29, 2021, 5:42 PM GMT+1 Project Member

Pri-0 bugs are critical regressions or serious emergencies, and this bug has not been updated in three days. Could you please provide an update, or adjust the priority to a more appropriate level if applicable?

If a fix is in active development, please set the status to Started.

Thanks for your time! To disable nags, add the Disable-Nags label.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 25 by jorgelo@chromium.org on Fri, Jan 29, 2021, 5:47 PM GMT+1 Project Member

Micah, this probably needs an update.

Comment 26 by jorgelo@chromium.org on Wed, Feb 3, 2021, 8:05 PM GMT+1 Project Member

Can we mark this as fixed now that some of the blocking bugs have been fixed? Has this been coordinated with the release folks?

Comment 27 by mortonm@chromium.org on Wed, Feb 3, 2021, 8:35 PM GMT+1 Project Member

Status: Fixed (was: Assigned)

The mediadrm command injection and debugd file write fixes have been merged to M88 and M89, so we can mark this as fixed and continue working on the other sub-bugs.

Comment 28 by achuith@chromium.org on Thu, Feb 4, 2021, 5:33 PM GMT+1 Project Member

Cc: achuith@chromium.org

Comment 29 by sheriffbot on Thu, Feb 4, 2021, 6:44 PM GMT+1 Project Member

Labels: reward-topanel

Comment 30 by sheriffbot on Thu, Feb 4, 2021, 7:58 PM GMT+1 Project Member

Labels: -Restrict-View-SecurityTeam Restrict-View-SecurityNotify

Comment 31 by sheriffbot on Thu, Feb 4, 2021, 8:24 PM GMT+1 Project Member

Labels: Merge-Request-89

This is sufficiently serious that it should be merged to beta. But I can't see a Chromium repo commit here, so you will need to investigate what - if anything - needs to be merged to M89. Is there a fix in some other repo which should be merged? Or, perhaps this ticket is a duplicate of some other ticket which has the real fix: please track that down and ensure it is merged appropriately.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 32 by sheriffbot on Thu, Feb 4, 2021, 8:26 PM GMT+1 Project Member

Labels: -Merge-Request-89 Merge-Review-89 Hotlist-Merge-Review

This bug requires manual review: M89's targeted beta branch promotion date has already passed, so this requires manual review
Before a merge request will be considered, the following information is required to be added to this bug:

1. Does your merge fit within the Merge Decision Guidelines?
- Chrome: https://chromium.googlesource.com/chromium/src.git/+/master/docs/process/merge_request.md#when-to-request-a-merge
- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines
2. Links to the CLs you are requesting to merge.
3. Has the change landed and been verified on ToT?
4. Does this change need to be merged into other active release branches (M-1, M+1)?

5. Why are these changes required in this milestone after branch?
6. Is this a new feature?
7. If it is a new feature, is it behind a flag using finch?

Chrome OS Only:

8. Was the change reviewed and approved by the Eng Prod Representative? See Eng Prod ownership by component: http://go/cros-engprodcomponents

Please contact the milestone owner if you have questions.
Owners: benmason@(Android), bindusuvarna@(iOS), geohsu@(ChromeOS), pbommana@(Desktop)

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 33 by geo...@google.com on Sat, Feb 6, 2021, 12:06 AM GMT+1 Project Member

This got auto tagged by sheriffbot for M89. Is this already fixed in M89 or do we need to take action? Thanks

Comment 34 by jorgelo@chromium.org on Mon, Feb 8, 2021, 3:18 PM GMT+1 Project Member

Labels: -Hotlist-Merge-Review -Merge-Review-89

The relevant blocking bugs have already been merged.

Comment 35 by amyressler@google.com on Thu, Feb 18, 2021, 1:12 AM GMT+1 Project Member

Labels: -reward-topanel reward-unpaid reward-45000

*** Boilerplate reminders! ***
Please do NOT publicly disclose details until a fix has been released to all our users. Early public disclosure may cancel the provisional reward. Also, please be considerate about disclosure when the bug affects a core library that may be used by other products. Please do NOT share this information with third parties who are not directly involved in fixing the bug. Doing so may cancel the provisional reward. Please be honest if you have already disclosed anything publicly or to third parties. Lastly, we understand that some of you are not interested in money. We offer the option to donate your reward to an eligible charity. If you prefer this option, let us know and we will also match your donation - subject to our discretion. Any rewards that are unclaimed after 12 months will be donated to a charity of our choosing.

Please contact security-vrp@chromium.org with any questions.
****

Comment 36 by amyressler@google.com on Thu, Feb 18, 2021, 1:19 AM GMT+1 Project Member

Congratulations, Rory! The VRP Panel has decided to award you $45,000 for this report. Which I guess is known in ChromeOS currency as 1.5Rory. Nice work!

Comment 37 by rory@rorym.cnamara.com on Thu, Feb 18, 2021, 10:38 AM GMT+1

Thank you!

Comment 38 by awhalley@google.com on Fri, Feb 19, 2021, 11:35 PM GMT+1 Project Member

Labels: -reward-unpaid reward-inprocess

Comment 39 by sheriffbot on Thu, May 13, 2021, 7:52 PM GMT+2 Project Member

Labels: -Restrict-View-SecurityNotify allpublic

This bug has been closed for more than 14 weeks. Removing security view restrictions.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 40 by janagrill@google.com on Fri, May 14, 2021, 7:48 PM GMT+2 Project Member

Labels: LTS-Security-86 LTR-Merged-86

Applying LTS labels, since all the associated fixes have been backported.