IssueTracker

Skip Navigation

M

Create

Assigned to me

Starred by me

CC'd to me

Collaborating

Reported by me

To be verified

Bookmark groups

Saved searches

Hotlists

Create hotlist

Create bookmark group

Browse components

Issue 324336238

... >...5705 > Public Vulnerability Reports 324622856 324336238

TPM2 Out-Of-Bounds Write

+1 1 Hotlists (4) Mark as Duplicate

Comments (37)
Dependencies
Duplicates (0)
Blocking (0)
Resources (32)

Verified
Vulnerability P2

Add Hotlist

STATUS UPDATE
No update yet.

DESCRIPTIONbu...@google.com created issue on behalf of av...@gmail.com#1
Feb 7, 2024 08:45PM

Report description
TPM2 Out-Of-Bounds Write

Bug location
Which product or website have you found a vulnerability in?
ChromeOS
Which URL (or repository) have you found the vulnerability in?
src/third_party/tpm2/

The problem
Please describe the technical details of the vulnerability
Recently, soon after I found the previous bug, I have found an out-of-bounds write vulnerability in the TPM2 Reference Library. This allows attackers to change the rma auth code during the Challenge-Response process. Because the cr50 allows you to send additional tpm commands during the rma auth code, you can change it, allowing yourself to disable write-protect allowing for the bypass of operating system verification. This allows attackers that have root access to the device beforehand to do more persistent actions, and fully take over the device.
Attached is the proof-of-concept that will trigger the out-of-bounds, and the video of it being triggered on the simulator. Fortunately, this doesn't trigger any pAsserts on the way, so this is a valid out-of-bounds write. I also determined that pAsserts would trigger tpm failure mode,

but not crash on the tpm2-simulator (and the tpm2-simulator only) so it was easy to deduct that this is a real out of bounds.
My fix would be to constrain the amount of bytes that can be read from the NVRAM spaces, to prevent the buffer from overflowing while reading the bytes. Please note that what is copied will be completely controlled. The potential fix is listed below, please take a look at it and consider the patch. Unfortunately, the zip file contains many files that are important to the build process of the exploit. Therefore I will place the zip file, and only show the contents of the relevant parts of the exploit (src/main.cpp,src/esys_context.h, and src/tss_context.h)
A quick run-down of the path that leads to the out-of-bounds write in the TPM2 Ref library:

- The embedder (cr50/tpm2-simulator) calls into the TPM2 Reference Layer via ExecuteCommand.
  https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/platform/cr50/common/tpm_registers.c;drc=f20631b8a50fb9f7dfe6bdf55786df2244c3f664;l=1124
- ExecuteCommand forwards into CommandDispatcher. At this point, the TPM Header and related auth sessions are validated by that point.
  https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/third_party/tpm2/CommandDispatcher.c;drc=f4428141132ec85eb255a819fc5bdaea2303f6af;l=503
- CommandDispatcher forwards into Exec_NV_Read, this is where the custom path starts
  https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/third_party/tpm2/CommandDispatcher.c;drc=f4428141132ec85eb255a819fc5bdaea2303f6af;l=503
- Exec_NV_Read will then unmarshal the contents of what will be read. This is the offset and size. Notice that the NV_Read_Out output buffer, is allocated on the stack, and isn't resizable. The resulting buffer is only sized 1024 which is very important for the next step.
  https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/third_party/tpm2/Marshal_NV_Read.c;drc=c3730f67070589a710864eeccdc79260d4c06756;l=62
- Exec_NV_Read forwards into TPM2_NV_Read, which does actual size checks. This checks if the size requested is less than the amount of space available to read from NVRAM.
  https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/third_party/tpm2/NV_Read.c;drc=5679752bf24c21135884e987c4077e2f71848971;l=44
  Once the check shown above is established to be true, the copying is done in NvGetIndexData which no longer does any bounds checks. (Notice that the data that is read is controllable from NV_Write and we have an exemplar of us controlling the data copied)
- From here, we determine that we can now write out-of-bounds (overwrite rma auth code), and enable factory mode.
  Some of the exploit specifics is overwriting the pointers stored on the stack on Exec_NV_Read that will allow us to write a known value to an arbitrary location (rma auth code address). This allows us to enable factory mode, with a value we know about. We can probe this value, by running it legitimately, and then use this data to respond to the challenge via the oob write.

Here are the steps of how to reproduce this vulnerability: Requirements: Have installed docker (This is for testing on a vm).

- Setup a chromiumos vm by downloading the vm from this command https://commondatastorage.googleapis.com/chromiumos-image-archive/amd64-generic-public/R121-15691.0.0/chromiumos_test_image.tar.xz
- Extract the image, and run it with `qemu-system-x86_64 -enable-kvm -display gtk,gl=on,show-cursor=on -hda <path to image> -device virtio-vga-gl -net nic,model=virtio-net-pci -net user,hostfwd=tcp::2222-:22 -usb -device usb-ehci -device usb-tablet` Now you should have a qemu session running chromiumos with ssh forwarding. Run `scp -P 2222 <download_path>/rmasmoke_shim.sh root@localhost:/usr/local/bin/rmasmoke` Run `make deploy-cros` to run it on the vm. This command requires docker to be installed, so have it prepared. This command will build the entire thing in a docker container, and then export it as a chroot, so it can be copied to the system. Most libraries are statically linked, and this will compile fine if you have docker.
- Ssh to your vm and run deploy_rmasmoke
- Run `rmasmoke pwd` and the pid of tpm2-simulator will change. This demonstrates that the process crashes. Also check /var/log/upstart.log. Note that the password destination (argument 1) doesn't matter This is the destination path of the owner password obtained from protobuf. It will obtain the owner password from the system. On a real cr50, this will reboot the system.

I also made a small shim builder tool in there to reproduce on the cr50 board, and you can use the rma shim for the corresponding board to run the program on a real cr50 board.
An exploit will be attached later that will disable wp.
Please feel free to ask me any questions.

Please briefly explain who can exploit the vulnerability, and what they gain when doing so

An attacker with root access can exploit this vulnerability to gain persistence.

The cause

What version of ChromeOS have you found the security issue in?

stable

Manufacturer and model of the device running ChromeOS

Cr50 Boards

Choose the type of vulnerability

Chrome OS- Exploit Persistence

Provide a demonstration of the security bug on ChromeOS, including all steps needed to reproduce the issue. Be as detailed as possible

- Setup a chromiumos vm by downloading the vm from this command https://commondatastorage.googleapis.com/chromiumos-image-archive/amd64-generic-public/R121-15691.0.0/chromiumos_test_image.tar.xz
- Extract the image, and run it with `qemu-system-x86_64 -enable-kvm -display gtk,gl=on,show-cursor=on -hda <path to image> -device virtio-vga-gl -net nic,model=virtio-net-pci -net user,hostfwd=tcp::2222-:22 -usb -device usb-ehci -device usb-tablet`
  Now you should have a qemu session running chromiumos with ssh forwarding. Run

```
scp -P 2222 <download_path>/rmasmoke_shim.sh
root@localhost:/usr/local/bin/rmasmoke
```

- Run `make deploy-cros` to run it on the vm. This command requires docker to be installed, so have it prepared. This command will build the entire thing in a docker container, and then export it as a chroot, so it can be copied to the system. Most libraries are statically linked, and this will compile fine if you have docker.
- Ssh to your vm and run deploy_rmasmoke
- Run `rmasmoke pwd` and the pid of tpm2-simulator will change. This demonstrates that the process crashes. Also check /var/log/upstart.log. Note that the password destination (argument 1) doesn't matter This is the destination path of the owner password obtained from protobuf. It will obtain the owner password from the system.

Does anyone else know about this vulnerability?

No, this vulnerability is private

Do you plan to disclose this bug publicly?

No

How would you like to be publicly acknowledged for your report?

Avadhut Mahamuni

RMASmoke-main.zip

6.5 MB Download

main.cpp

8.3 KB Download

esys_context.h

24 KB Download

tss_context.h

1.3 KB Download

RMASmoke-main.zip

6.5 MB Download

upstart.log

465 B Download

Issue summary

✓ Similar issues ✓ Mentioned issues (1) ✓ Links (25)

Hide all

Similar issues

Title	Dupes	+1s	Created	
ShelfModel::Move function lacks index check, resulting in heap-use-after-free and heap-buf	0	1	Jan 11, 2024	⋮

Mentioned issues (1)

Links (25)

“ .dnammoCetucexE aiv reyaL ecnerefeR 2MPT eht otni sllac (rotalumis-2mpt/05rc) reddebme ehThttps://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/platform/cr50/common/tpm_registers.c;drc=f20631b8a50fb9f7dfe6bdf55786df2244c3f664;l=1124 ”
bu...@ #1

“ https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/third_party/tpm2/CommandDispatcher.c;drc=f4428141132ec85eb255a819fc5bdaea2303f6af;l=503 ”
bu...@ #1

“ https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/third_party/tpm2/Marshal_NV_Read.c;drc=c3730f67070589a710864eeccdc79260d4c06756;l=62 ”
bu...@ #1

“ .MARVN morf daer ot elbaliava ecaps fo tnuoma eht naht ssel si detseuqer ezis eht fi skcehc sihT .skcehc ezis lautca seod hcihw ,daeR_VN_2MPT otni sdrawrof daeR_VN_cexEhttps://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/third_party/tpm2/NV_Read.c;drc=5679752bf24c21135884e987c4077e2f71848971;l=44 ”
bu...@ #1

“ dnammoc siht morf mv eht gnidaolnwod yb mv somuimorhc a puteShttps://commondatastorage.googleapis.com/chromiumos-image-archive/amd64-generic-public/R121-15691.0.0/chromiumos_test_image.tar.xz ”
bu...@ #1

See all related links

COMMENTS

All comments

Oldest first

---

ro...@google.com <ro...@google.com> #2Feb 7, 2024 08:51PM

Assigned to ap...@google.com.

Thank you for your report Avadhut!

---

av...@gmail.com <av...@gmail.com> #3Feb 7, 2024 09:03PM

Thank you for your support as well. I also have attached a video demonstrating the vulnerability being triggered on the TPM2.

deleted
0 B

---

av...@gmail.com <av...@gmail.com> #4Feb 7, 2024 09:09PM

The contents of `deploy_rmasmoke` is also shown below.

```
#!/bin/bash

pushd /mnt/stateful_partition
mkdir rmasmoke_root
tar -C rmasmoke_root -xvf rmasmoke_root.tar.xz
popd
```

Copy this to `/usr/local/bin/deploy_rmasmoke` within the ChromeOS root which extracts the chroot onto the device. This isn't necessary for shims as the root is already extracted (referring to `build_shim.sh`).

---

ap...@google.com <ap...@google.com> #5Feb 8, 2024 03:45PM

Reassigned to su...@google.com.

Summarizing the vulnerability:

- An attacker can craft an nvmem space that is longer than 1024 bytes, then request reading more than 1024 bytes. The request will pass the checks, and then overwrite the stack past the `NV_Read_Out out` location with whatever was crafted in that nvmem space. This allows overwriting the return stack and calling any existing cr50 routine on the return path.
- The long nvmem space data shall be crafted separately for each cr50 version, but there's no address randomization to prevent calling a known code snippet.
- The issue is not specific to RMA auth value or WP.

Supporting details:

- `NV_Read_Out` structure is essentially `TPM2B_MAX_NV_BUFFER`, which contains `BYTE buffer[MAX_NV_BUFFER_SIZE]` as the target buffer, and `MAX_NV_BUFFER_SIZE` is 1024.
- The only check made by `TPM2_NV_Read` is that `(in->size + in->offset) <= nvIndex.publicArea.dataSize`, and it is never checked that `in->size <= MAX_NV_BUFFER_SIZE`. The issue is present in TPM reference code 1.16 and was fixed in 1.38. I don't see an associated CVE or errata, though.
- `NV_Read_In_Unmarshal` does just `UINT16_Unmarshal` for the size parameter, w/o restricting to `MAX_NV_BUFFER_SIZE`. `TPM2B_MAX_NV_BUFFER_Unmarshal` that checks for `MAX_NV_BUFFER_SIZE` is never used for NV_Read path (though used for Write and similar paths).

Fix

The fix is adding "`if (in->size > MAX_NV_BUFFER_SIZE) return /* error */;`" to the places where we fill TPM2B_MAX_NV_BUFFER from nvmem contents.

Specifically the places are

- `NV_Read_Out` used in `TPM2_NV_Read`: requries adding the size check.

```
 TPM_RC

TPM2_NV_Read(
  NV_Read_In       *in,              // IN: input parameter list
  NV_Read_Out      *out              // OUT: output parameter list
  )
{
...
  result = NvReadAccessChecks(in->authHandle, in->nvIndex);
  if(result != TPM_RC_SUCCESS)
     return result;

+   if(in->size > MAX_NV_BUFFER_SIZE)
+      return TPM_RC_VALUE + TPM_RC_P + TPM_RC_1;

  // Too much data
```

```
    if((in->size + in->offset) > nvIndex.publicArea.dataSize)
        return TPM_RC_NV_RANGE;
...
}
```

- `nvContents` in `TPMS_NV_CERTIFY_INFO` used in `TPM2_NV_Certify`: requires adding the size check.

```
 TPM_RC

TPM2_NV_Certify(
  NV_Certify_In     *in,              // IN: input parameter list
  NV_Certify_Out    *out                // OUT: output parameter list
  )
{
...
  result = NvReadAccessChecks(in->authHandle, in->nvIndex);
  if(result != TPM_RC_SUCCESS)
      return result;

+   if(in->size > MAX_NV_BUFFER_SIZE)
+       return TPM_RC_VALUE + TPM_RC_P + TPM_RC_3;

  // See if the range to be certified is out of the bounds of the defined
  // Index
  if((in->size + in->offset) > nvIndex.publicArea.dataSize)
      return TPM_RC_NV_RANGE;
...
}
```

- `NV_Extend_In` used in `Exec_NV_Extend`: there `NV_Extend_In_Unmarshal` already calls `TPM2B_MAX_NV_BUFFER_Unmarshal` that does the right checks; no fixes needed.
- `NV_Write_In` used in `Exec_NV_Write`: there `NV_Write_In_Unmarshal` already calls `TPM2B_MAX_NV_BUFFER_Unmarshal` that does the right checks; no fixes needed.

An additional defense in depth might be passing max size to `NvGetIndexData`.

---

av...@gmail.com <av...@gmail.com> #6Feb 8, 2024 04:08PM

I have a patch file for this as well. Please consider the patch file below:

tpm2.patch

474 B ViewDownload

---

jw...@google.com <jw...@google.com> #7Feb 8, 2024 04:11PM

Wow, this is bad. :/ I assume that due to the lack of ASLR it would probably be possible to turn this into a full RCE (making it jump through some hoops into shell code e.g. from the attacker-controlled NV space) with a bit more effort? Do we just fix this or do we also need to evaluate what existing secrets may have been stolen and need to be renewed (if possible) in case someone had already been using this in the wild?
For the fix: should we also add a check to DefineSpace to disallow creating NV spaces larger than 1024? It doesn't make sense to allow the user to define a space so large that we couldn't support reading it back anyway. (FWIW the TPM spec says about TPM2_NV_DefineSpace: "If the NV Index is an ordinary Index and publicInfo→dataSize is larger than supported by the TPM implementation then the TPM shall return TPM_RC_SIZE.")

---

ap...@google.com <ap...@google.com> #8Feb 8, 2024 04:16PM

> should we also add a check to DefineSpace to disallow creating NV spaces larger than 1024?

not really, and we do use spaces larger than that.

> we couldn't support reading it back anyway

it can be read, one just needs multiple commands (with {offset = 0, size=1024} then {offset = 1024, size=1024}, etc). there is an additional check in later specs that WRITEALL spaces cannot be larger than this MAX_NV_BUFFER_SIZE parameter (since they must be written in one go, which would become impossible). I haven't checked if we have that check already.

---

ap...@google.com <ap...@google.com> #9Feb 8, 2024 04:19PM

> "If the NV Index is an ordinary Index and publicInfo→dataSize is larger than supported by the TPM implementation then the TPM shall return TPM_RC_SIZE."

That's a different parameter: MAX_NV_INDEX_SIZE.

av...@gmail.com <av...@gmail.com> #10Feb 8, 2024 04:19PM

Yes there is no aslr, or stack canaries. This could be turned into a full rce, on ARM, it is difficult to control the RIP equivalent (Program Counter), but you can modify other parts of it with an argument pointer being set after the resulting call. One of these is the vNVRAM spaces which store raw function pointers that point to rodata functions [1] and execute such functions after being ran [2] , and they can be changed to other stuff (such as factory mode) which allows for further code execution on the AP Firmware.
[1]
https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/platform/cr50/board/cr50/tpm2/virtual_nvmem.c;drc=83ddf363efb34712a6c152eccbb345cd01390d27;l=349
[2]
https://source.chromium.org/chromiumos/chromiumos/codesearch/+/main:src/platform/cr50/board/cr50/tpm2/virtual_nvmem.c;drc=83ddf363efb34712a6c152eccbb345cd01390d27;l=250

---

av...@gmail.com <av...@gmail.com> #11Feb 8, 2024 04:20PM

#8 I'm also wondering if due to that, this could be ran from chronos through tpm_manager_client, a well known entrypoint into the tpm_manager service from chronos. This may bypass the need to gain root.

---

ap...@google.com <ap...@google.com> #12Feb 8, 2024 04:23PM

Re comment #6:
I have a patch file for this as well. Please consider the patch file below
yes, that's a reasonable 1st patch, but NV_Certify shall also be fixed, and there's a later recommendation that RC_NV_RANGE is not the best error code (since hard to distinguish size vs offset vs other errors), so RC_VALUE is recommended (and RC_NV_RANGE in the 2nd check technically also better be modified, but we won't do it in this fix).
So, the right patch is in comment #5.

---

jw...@google.com <jw...@google.com> #13Feb 8, 2024 04:30PM

> should we also add a check to DefineSpace to disallow creating NV spaces larger than 1024?

not really, and we do use spaces larger than that.

Hmmm... okay, if we need it then we need it, I guess. I don't see anything in the TPM2_NV_Read specification that says "if the `size` parameter is too large you're allowed to return this error and then that tells the caller that they'll have to chunk the request in smaller sizes", though, so it seems that creating those restrictions is not really spec compliant. But I guess we don't have a better option.

---

ap...@google.com <ap...@google.com> #14Feb 8, 2024 04:30PM

> this could be ran from chronos through tpm_manager_client

not through tpm_manager and similar utilities based on libtrunks's TpmUtility classes. there we do have code that ensures that the buffers sent to NV_Read are no larger than 1024. sending would require utilities (or direct access to /dev/tpm[rm]*) that can send raw commands.

---

jw...@google.com <jw...@google.com> #15Feb 8, 2024 04:32PM

Also, if this bug came from the TPM reference implementation, did we check whether Ti50 is also affected? (It does still use the reference code, right? Or did we also rewrite that in Rust?)

---

ap...@google.com <ap...@google.com> #16Feb 8, 2024 04:44PM

ti50 is not affected. reference implementation fixed in 1.38

---

ro...@google.com <ro...@google.com> #17Feb 8, 2024 08:52PM

We are keeping the priority at p2 to make sure we go through beta before we push out to stable and lts.

---

ap...@google.com <ap...@google.com> #18Feb 8, 2024 09:00PM

Project: chromiumos/third_party/tpm2
Branch: main

commit 3ce9ff6661d4738d6c66ad0e0b4d6d8ba37d71d2
Author: Vadim Sukhomlinov <sukhomlinov@google.com>
Date: Thu Feb 08 14:17:45 2024

tpm2: update checks in TPM2_NV_Read(), TPM2_NV_Certify(), TPM2_NV_Write

Implement same checks for incoming data size as in latest TPM2 version.

BUG=b:324336238
TEST=make cr50

Change-Id: I2a1194dafb643ab9069ca2989f2d52b97edbd706
Reviewed-on: https://chromium-review.googlesource.com/c/chromiumos/third_party/tpm2/+/5280421
Commit-Queue: Vadim Sukhomlinov <sukhomlinov@google.com>
Commit-Queue: Andrey Pronin <apronin@chromium.org>
Reviewed-by: Vadim Sukhomlinov <sukhomlinov@google.com>
Auto-Submit: Vadim Sukhomlinov <sukhomlinov@google.com>
Reviewed-by: Andrey Pronin <apronin@chromium.org>
Code-Coverage: Vadim Sukhomlinov <sukhomlinov@google.com>
Tested-by: Vadim Sukhomlinov <sukhomlinov@google.com>

M NV_Certify.c
M NV_DefineSpace.c
M NV_Read.c
M NV_Write.c
M TPM_Types.h

https://chromium-review.googlesource.com/5280421

---

av...@gmail.com <av...@gmail.com> #19Feb 8, 2024 09:40PM

Comment has been deleted.

Message last modified on May 22, 2024 06:35PM

---

ap...@google.com <ap...@google.com> #20Feb 8, 2024 10:20PM

Project: chromiumos/third_party/tpm2
Branch: firmware-cr50-stab-14294.B

commit 28e6aec07dba0f4d7ce9788ab0ce3852879abc86
Author: Vadim Sukhomlinov <sukhomlinov@google.com>
Date: Thu Feb 08 14:17:45 2024

tpm2: update checks in TPM2_NV_Read(), TPM2_NV_Certify(), TPM2_NV_Write

Implement same checks for incoming data size as in latest TPM2 version.

BUG=b:324336238
TEST=make cr50

Change-Id: I2a1194dafb643ab9069ca2989f2d52b97edbd706
Reviewed-on: https://chromium-review.googlesource.com/c/chromiumos/third_party/tpm2/+/5280421
Commit-Queue: Vadim Sukhomlinov <sukhomlinov@google.com>
Commit-Queue: Andrey Pronin <apronin@chromium.org>
Reviewed-by: Vadim Sukhomlinov <sukhomlinov@google.com>
Auto-Submit: Vadim Sukhomlinov <sukhomlinov@google.com>
Reviewed-by: Andrey Pronin <apronin@chromium.org>
Code-Coverage: Vadim Sukhomlinov <sukhomlinov@google.com>
Tested-by: Vadim Sukhomlinov <sukhomlinov@google.com>
(cherry picked from commit 3ce9ff6661d4738d6c66ad0e0b4d6d8ba37d71d2)
Reviewed-on: https://chromium-review.googlesource.com/c/chromiumos/third_party/tpm2/+/5281194
Auto-Submit: Mary Ruthven <mruthven@chromium.org>
Commit-Queue: Mary Ruthven <mruthven@chromium.org>
Tested-by: Mary Ruthven <mruthven@chromium.org>

M NV_Certify.c
M NV_DefineSpace.c
M NV_Read.c
M NV_Write.c
M TPM_Types.h

https://chromium-review.googlesource.com/5281194

---

ap...@google.com <ap...@google.com> #21Feb 12, 2024 01:37PM

Project: chromiumos/third_party/tpm2
Branch: firmware-cr50-stab-mp-14300.B

commit e9a4df61cbc49527a29da2f4f2e9e6cfae95a1f4
Author: Vadim Sukhomlinov <sukhomlinov@google.com>
Date: Thu Feb 08 14:17:45 2024

  tpm2: update checks in TPM2_NV_Read(), TPM2_NV_Certify(),
TPM2_NV_Write

  Implement same checks for incoming data size as in latest TPM2 version.

  BUG=b:324336238
  TEST=make cr50

  Change-Id: I2a1194dafb643ab9069ca2989f2d52b97edbd706
  Reviewed-on: https://chromium-
review.googlesource.com/c/chromiumos/third_party/tpm2/+/5280421
  Commit-Queue: Vadim Sukhomlinov <sukhomlinov@google.com>
  Commit-Queue: Andrey Pronin <apronin@chromium.org>
  Reviewed-by: Vadim Sukhomlinov <sukhomlinov@google.com>
  Auto-Submit: Vadim Sukhomlinov <sukhomlinov@google.com>
  Reviewed-by: Andrey Pronin <apronin@chromium.org>
  Code-Coverage: Vadim Sukhomlinov <sukhomlinov@google.com>
  Tested-by: Vadim Sukhomlinov <sukhomlinov@google.com>
  (cherry picked from commit 3ce9ff6661d4738d6c66ad0e0b4d6d8ba37d71d2)
  Reviewed-on: https://chromium-
review.googlesource.com/c/chromiumos/third_party/tpm2/+/5281194
  Auto-Submit: Mary Ruthven <mruthven@chromium.org>
  Commit-Queue: Mary Ruthven <mruthven@chromium.org>
  Tested-by: Mary Ruthven <mruthven@chromium.org>
  (cherry picked from commit 28e6aec07dba0f4d7ce9788ab0ce3852879abc86)
  Reviewed-on: https://chromium-
review.googlesource.com/c/chromiumos/third_party/tpm2/+/5286991

M NV_Certify.c
M NV_DefineSpace.c
M NV_Read.c
M NV_Write.c
M TPM_Types.h

https://chromium-review.googlesource.com/5286991

---

su...@google.com <su...@google.com> #22Feb 16, 2024 07:09PM

Accepted by su...@google.com.

I suppose functionally it is fixed, so now it is related to landing in ChromeOS/Cr50.

ap...@google.com <ap...@google.com> #23Feb 27, 2024 05:02PM

Project: chromiumos/third_party/tpm2
Branch: firmware-cr50-prepvt-15608.B

commit d947b63c901d0c952f0886d6c4333134a3c30f91
Author: Vadim Sukhomlinov <sukhomlinov@google.com>
Date: Thu Feb 08 14:17:45 2024

tpm2: update checks in TPM2_NV_Read(), TPM2_NV_Certify(), TPM2_NV_Write

Implement same checks for incoming data size as in latest TPM2 version.

BUG=b:324336238
TEST=make cr50

Change-Id: I2a1194dafb643ab9069ca2989f2d52b97edbd706
Reviewed-on: https://chromium-review.googlesource.com/c/chromiumos/third_party/tpm2/+/5280421
Commit-Queue: Vadim Sukhomlinov <sukhomlinov@google.com>
Commit-Queue: Andrey Pronin <apronin@chromium.org>
Reviewed-by: Vadim Sukhomlinov <sukhomlinov@google.com>
Auto-Submit: Vadim Sukhomlinov <sukhomlinov@google.com>
Reviewed-by: Andrey Pronin <apronin@chromium.org>
Code-Coverage: Vadim Sukhomlinov <sukhomlinov@google.com>
Tested-by: Vadim Sukhomlinov <sukhomlinov@google.com>
(cherry picked from commit 3ce9ff6661d4738d6c66ad0e0b4d6d8ba37d71d2)
Reviewed-on: https://chromium-review.googlesource.com/c/chromiumos/third_party/tpm2/+/5328152
Auto-Submit: Mary Ruthven <mruthven@chromium.org>
Commit-Queue: Mary Ruthven <mruthven@chromium.org>
Tested-by: Mary Ruthven <mruthven@chromium.org>

M NV_Certify.c
M NV_DefineSpace.c
M NV_Read.c
M NV_Write.c
M TPM_Types.h

https://chromium-review.googlesource.com/5328152

av...@gmail.com <av...@gmail.com> #24Feb 27, 2024 05:25PM

Greetings all, I wanted to thank you all for the support to find a fix for this vulnerability. Thank you, please let me know if this can have a CVE assigned for this.

---

st...@google.com <st...@google.com> #25Mar 6, 2024 11:36AM

Hi OP, We've reserved CVE-2024-0203 for this issue.

---

ch...@google.com <ch...@google.com> #26Mar 6, 2024 02:28PM

This issue requires additional review before it can be merged to the LTS channel. Please answer the following questions to help us evaluate this merge:

1. Number of CLs needed for this fix and links to them.
2. Level of complexity (High, Medium, Low - Explain)
3. Has this been merged to a stable release? beta release?
4. Overall Recommendation (Yes, No)

---

av...@gmail.com <av...@gmail.com> #27Mar 11, 2024 07:17AM

#25 Thank you for getting a CVE for this bug, unfortunately the CVE has been used up by someone else. Can another CVE be acquired for this bug report?

---

ap...@google.com <ap...@google.com> #28Mar 12, 2024 07:03PM

Project: chromiumos/third_party/tpm2
Branch: firmware-cr50-mp-15611.B

commit 08425f1a7ea6af7325e7f266b53d6026b91038a4
Author: Vadim Sukhomlinov <sukhomlinov@google.com>
Date: Thu Feb 08 14:17:45 2024

tpm2: update checks in TPM2_NV_Read(), TPM2_NV_Certify(),

TPM2_NV_Write

Implement same checks for incoming data size as in latest TPM2 version.

BUG=b:324336238
TEST=make cr50

Change-Id: I2a1194dafb643ab9069ca2989f2d52b97edbd706
Reviewed-on: https://chromium-review.googlesource.com/c/chromiumos/third_party/tpm2/+/5280421
Commit-Queue: Vadim Sukhomlinov <sukhomlinov@google.com>
Commit-Queue: Andrey Pronin <apronin@chromium.org>
Reviewed-by: Vadim Sukhomlinov <sukhomlinov@google.com>
Auto-Submit: Vadim Sukhomlinov <sukhomlinov@google.com>
Reviewed-by: Andrey Pronin <apronin@chromium.org>
Code-Coverage: Vadim Sukhomlinov <sukhomlinov@google.com>
Tested-by: Vadim Sukhomlinov <sukhomlinov@google.com>
(cherry picked from commit 3ce9ff6661d4738d6c66ad0e0b4d6d8ba37d71d2)
Reviewed-on: https://chromium-review.googlesource.com/c/chromiumos/third_party/tpm2/+/5328152
Auto-Submit: Mary Ruthven <mruthven@chromium.org>
Commit-Queue: Mary Ruthven <mruthven@chromium.org>
Tested-by: Mary Ruthven <mruthven@chromium.org>
(cherry picked from commit d947b63c901d0c952f0886d6c4333134a3c30f91)
Reviewed-on: https://chromium-review.googlesource.com/c/chromiumos/third_party/tpm2/+/5366177

M NV_Certify.c
M NV_DefineSpace.c
M NV_Read.c
M NV_Write.c
M TPM_Types.h

https://chromium-review.googlesource.com/5366177

---

av...@gmail.com <av...@gmail.com> #29Mar 24, 2024 01:13PM

Please let me know if this vulnerability qualifies for critical severity as per [1]. This vulnerability can bypass verified boot, by allowing for the modification of RO Firmware, by disabling wp through the cr50.
[1] https://www.chromium.org/chromium-os/developer-library/guides/bugs/security-severity-guidelines/

---

ro...@google.com <ro...@google.com> #30Mar 24, 2024 01:42PM

This is not a critical severity vulnerability in that an attacker would need to successfully exploit one or more vulnerabilities to get root on the system.
To qualify for the persistence bonus
https://bughunters.google.com/about/rules/4919474699501568/chromeos-vulnerability-reward-program-rules#reward-amounts
Your would need to submit an exploit chain that compromises a Chromebook or Chromebox with device persistence in guest mode (i.e. guest-to-guest persistence with interim reboot, delivered via a web page).
That said this is excellent work and our VRP Panel will take that into consideration when making a reward decision!
We thank you for your report and encourage you to keep submitting issues to our VRP.

---

av...@gmail.com <av...@gmail.com> #31Mar 24, 2024 01:48PM

Comment has been deleted.

Message last modified on Mar 24, 2024 04:28PM

---

av...@gmail.com <av...@gmail.com> #32Mar 24, 2024 04:29PM

Thank you

---

ap...@google.com <ap...@google.com> Apr 4, 2024 01:07PM

Assigned to mr...@google.com.

---

mr...@google.com <mr...@google.com> #33Apr 4, 2024 02:37PM

The fixes are in cr50 version 0.5.230. We'll mark this fixed after that is cherry-picked into LTS.

---

mr...@google.com <mr...@google.com> Apr 5, 2024 09:59AM

Marked as fixed.

---

gm...@google.com <gm...@google.com> #34Apr 5, 2024 11:43AM

LTS merge review and approval tracked in
https://b.corp.google.com/issues/324475884

---

rb...@google.com <rb...@google.com> #35Apr 5, 2024 01:19PM

Exploitability - Explain why/why not the bug is reachable and/or exploitable For example, if a bug mentions a race, details are needed about how easy that race would be to achieve / can the attack retry infinite times to win the race, etc..
The bug is reachable through a Out-Of-Bounds write within TPM2. There is no bounds checking on the length of input so the buffer can be overflowed.
Privileges and Capabilities - Identify which process is exploited and where code execution potentially can be achieved if the attacker can break out of that process, and explain why
Cr50 board's TPM. The return stack can be overwritten to call existing cr50 routines.
Origin of fix - Is the issue already known upstream, fixed by work from a previously known or reported issue, provided by the reporter, or any other information that would be relevant toward reward eligibility
Google engineer fixed and cherry picked to LTS.
Mitigations - Detailed explanation of any relevant mitigations
Added bounds checking to prevent overwriting the buffer.

---

ro...@google.com <ro...@google.com> Apr 9, 2024 12:04PM

Verified by cr...@google.com.

---

sp...@google.com <sp...@google.com> #36Apr 9, 2024 12:55PM

** NOTE: This is an automatically generated email **

Hello,

ChromeOS Vulnerability Reward Program panel has decided to issue a reward of $10000.00 for your report. Congratulations!

Important: If you aren't already registered with Google as a supplier, p2p-vrp@google.com will reach out to you. If you have registered in the past, no need to repeat the process – you can sit back and relax, and we will process the payment soon.

If you have any payment related requests, please direct them to p2p-vrp@google.com. Please remember to include the subject of this email and the email address that the report was sent from.

Regards,

Google Security Bot

P.S. Two other things we'd like to mention:

* If you'd like us to add your name to our Leaderboard at https://bughunters.google.com/leaderboard, please create a profile at https://bughunters.google.com if you haven't already done so.
* We encourage you to sign up for our Vulnerability Research Grants program at https://www.google.com/about/appsecurity/research-grants/, where we issue monetary payments to VRP researchers, even when no vulnerabilities are found.

--
How did we do? Please fill out a short anonymous survey (https://goo.gl/IR3KRH).

av...@gmail.com <av...@gmail.com> #37Apr 9, 2024 06:27PM

Wow! Thank you so much for supporting me through the whole process.

Add comment

Comment

Issue metadata

Reporter
av...@gmail.com

Type
Vulnerability

Priority
P2

Severity
S2

Status
Fixed (Verified)

Access
Default access View

Expanded Access

Assignee
mr...@google.com

Verifier
cr...@google.com

Collaborators

ap...@google.com
mr...@google.com
su...@google.com

CC

Ch...@google.com
at...@google.com
av...@gmail.com
cf...@google.com
jw...@google.com
mr...@google.com
ro...@google.com
st...@google.com

su...@google.com

Code Changes

--

Pending Code Changes

--

Staffing

--

CLs: Merged

crrev/c/5280421

crrev/c/5281194

crrev/c/5286991

... and 2 more (show all)

CLs: Pending

--

CVE#

CVE-2024-0203

Embargo disclosure date

--

Is issue under embargo

False

Monorail dependency URL

--

Reporter credit

--

Requested charity

--

Reward amount

10000

Triage

Untriaged

Type

--

VRP Status

pending reward

Found In

--

Targeted To

--

Verified In

--

In Prod

Privacy|Terms