Improper Validation of Array Index in the cleanup_shm_refs function

High **jbech-linaro** published **GHSA-65w8-6mrg-52g7** 4 days ago

Package **OP-TEE** (OP-TEE)	
Affected versions <= 3.18.0	Patched versions 3.19.0

Description

Amazon Web Services found an Improper Validation of Array Index vulnerability [1] in OP-TEE OS. The function `cleanup_shm_refs()` is called by both `entry_invoke_command()` and `entry_open_session()` . The commands `OPTEE_MSG_CMD_OPEN_SESSION` and `OPTEE_MSG_CMD_INVOKE_COMMAND` can be executed from the normal world via an OP-TEE SMC. This function is not validating the `num_params` argument [2], which is only limited to `OPTEE_MSG_MAX_NUM_PARAMS` (127) in the function `get_cmd_buffer()` . Therefore, an attacker in the normal world can craft an SMC call that will cause out-of-bounds reading in `cleanup_shm_refs` and potentially freeing of fake-objects in the function `mobj_put()` .

In short, a normal-world attacker with permission to execute SMC instructions may exploit this flaw. We believe this problem permits local privilege escalation from the normal world to the secure world.

Trigger the problem
The following SMC instruction will trigger the bug:

```
Register   | Value
----------------------
x0         |   OPTEE_SMC_CALL_WITH_ARG
x1+x2      |   point to shared memory with the following struct:

struct optee_msg_arg {
    uint32_t cmd = OPTEE_MSG_CMD_INVOKE_COMMAND;
    uint32_t func;
    uint32_t session;
    uint32_t cancel_id;
    uint32_t pad;
    uint32_t ret;
    uint32_t ret_origin;
    uint32_t num_params = 127;
    struct optee_msg_param params[];
}
```

When triggering the problem with `num_params = 127` , one of the first things the `entry_invoke_command()` function does is to copy the parameters received from normal world by calling `copy_in_params()` which checks the following:

```
if(num_params > TEE_NUM_PARAMS)
   return TEE_ERROR_BAD_PARAMETERS;
```

Because `TEE_NUM_PARAMS` is defined as value 4, we end up in the function `cleanup_shm_refs()` which does not check the `num_params` input as mentioned in the introduction.

Details and mitigation
Once the bug is triggered, the for-loop in `cleanup_shm_refs()` will read out-of-bounds values from the `saved_attr` array,
which is found on the stack of the calling function. If the value on the stack is one of [`OPTEE_MSG_ATTR_TYPE_TMEM_INPUT` , `OPTEE_MSG_ATTR_TYPE_TMEM_OUTPUT` , `OPTEE_MSG_ATTR_TYPE_TMEM_INOUT`] or in addition [`OPTEE_MSG_ATTR_TYPE_RMEM_INPUT` , `OPTEE_MSG_ATTR_TYPE_RMEM_OUTPUT` , `OPTEE_MSG_ATTR_TYPE_RMEM_INOUT`] if `CFG_CORE_DYN_SHM` (dynamic shared memory) has been defined, then `mobj_put ()` will be called with an out-of-bounds index into the `param->u` array (which can be found on the calling stack as well), effectively forming a dangling pointer vulnerability. In order to fix this issue, the function `cleanup_shm_refs()` should limit the loop counter `n` to `MIN(TEE_NUM_PARAMS, num_params)` .

Severity rationale
Currently set to "high" based on the CVSSv3 scoring below [3].

Patches

optee_os.git

- core: tee_entry: fix array out of bounds check in cleanup_shm_refs()

Workarounds

N/A

References

[1] CWE-129: Improper Validation of Array Index
[2] cleanup_shm_refs() function uses num_params without validation.
[3] CVSSv3 calculator

OP-TEE ID

OP-TEE-2022-0002

Reported by

Amazon Web Services (Asaf Modelevsky [**@asafmod**]).

For more information

For more information regarding the security incident process in OP-TEE, please read the information that can be found when going to the "Security" page at https://www.trustedfirmware.org.

Timeline

2022-08-30: Initial report sent to TrustedFirmware.
2022-08-30: Confirmed that report has been received.
2022-08-30: OP-TEE maintainers internal assessment.
2022-08-31: Fix proposed internally.
2022-10-06: Informing Trusted Stakeholders.
2022-11-29: Providing the advisory to the wider public.

Severity

High

CVE ID

CVE-2022-46152

Weaknesses

CWE-129

Credits

asafmod