PKCS#11 TA out-of-bounds read and memory disclosure

High **jenswi-linaro** published **GHSA-8cqw-mg7v-c9p9** 16 hours ago

Package

PKCS#11 TA (OP-TEE)

Affected versions

>= 3.13.0

Patched versions

4.11 and later

Description

Summary

Case 1: Missing checks in `entry_get_attribute_value()` in `ta/pkcs11/src/object.c` can lead to out-of-bounds read from the PKCS#11 TA heap or a crash.

Case 2: The PKCS#11 TA function `PKCS11_CMD_GET_ATTRIBUTE_VALUE` or `entry_get_attribute_value()` can, with a bad template parameter, be tricked into:

1. Reading at most 7 bytes beyond the end of the template buffer
2. Writing beyond the end of the template buffer with the content of an attribute value of a PKCS#11 object.

Details case 1

Close to the end of `entry_get_attribute_value(), the updated template is copied out with:

```
        /* Move updated template to out buffer */
        TEE_MemMove(out->memref.buffer, template, out->memref.size);
```

However, the size of the template isn't checked before copying, so data beyond the `template` buffer is read if `out->memref.size` is large enough.

Details case 2

1. The for loop in entry_get_attribute_value() doesn't check that an attribute head (cli_head) is fully within the template buffer before copying it.
2. The length field in the copied attribute head (cli_head.size) is not checked for overflow. When it does overflow, it leads to the content of an attribute being written into the TA heap past the allocated temporary template buffer.

Reachability

The function is reached via the TA function `PKCS11_CMD_GET_ATTRIBUTE_VALUE`.

Impact

Leakage of data from the TA heap. For instance, previously supplied secret keys that haven't been cleared from the heap yet.

Corrupting the TA heap when writing beyond the temporary template buffer.

The TA can also crash if reading or writing beyond the mapped memory.

Patches

```
From 8372182598f61b344ba52bd66806cc745c2a1137 Mon Sep 17 00:00:00 2001
From: Etienne Carriere <etienne.carriere@st.com>
Date: Wed, 21 Jan 2026 13:55:33 +0100
Subject: [PATCH 1/3] ta: pkcs11: check output buffer size on get attribute
 value

Check client output buffer input size and update its output
size on PKCS11_CMD_GET_ATTRIBUTE_VALUE command.

Fixes: 783c1515c2f9 ("ta: pkcs11: Add support for getting object size and attribute valu
Signed-off-by: Etienne Carriere <etienne.carriere@st.com>
Reviewed-by: Jens Wiklander <jens.wiklander@linaro.org>
---
 ta/pkcs11/src/object.c | 10 ++++++++++
 1 file changed, 10 insertions(+)

diff --git a/ta/pkcs11/src/object.c b/ta/pkcs11/src/object.c
index c9a95e1b274d..ba3be7a7133a 100644
--- a/ta/pkcs11/src/object.c
+++ b/ta/pkcs11/src/object.c
@@ -800,6 +800,15 @@ enum pkcs11_rc entry_get_attribute_value(struct pkcs11_client *clie
        goto out;
    }
```

```
+   /*
+    * We will update the template with relevant data, without resizing it.
+    * Upon completion, it will be copied to client output buffer.
+    */
+   if (out->memref.size < sizeof(*template) + template->attrs_size) {
+       rc = PKCS11_CKR_ARGUMENTS_BAD;
+       goto out;
+   }
+
    /* Iterate over attributes and set their values */
    /*
     * 1. If the specified attribute (i.e., the attribute specified by the
@@ -912,6 +921,7 @@ enum pkcs11_rc entry_get_attribute_value(struct pkcs11_client *clien
        rc = PKCS11_CKR_BUFFER_TOO_SMALL;

    /* Move updated template to out buffer */
+   out->memref.size = sizeof(*template) + template->attrs_size;
    TEE_MemMove(out->memref.buffer, template, out->memref.size);

    DMSG("PKCS11 session %"PRIu32": get attributes %#"PRIx32,
--
2.43.0
```

```
From 8533b17204156d2269f8e7a6307c07fd812fe01e Mon Sep 17 00:00:00 2001
From: Etienne Carriere <etienne.carriere@st.com>
Date: Wed, 21 Jan 2026 13:58:09 +0100
Subject: [PATCH 2/3] ta: pkcs11: check template consistency on get attribute
 value

Check client template holds consistent attribute area sizes
value on PKCS11_CMD_GET_ATTRIBUTE_SIZE.

Fixes: 783c1515c2f9 ("ta: pkcs11: Add support for getting object size and attribute valu
Signed-off-by: Etienne Carriere <etienne.carriere@st.com>
Reviewed-by: Jens Wiklander <jens.wiklander@linaro.org>
---
 ta/pkcs11/src/object.c | 13 ++++++++++-
 1 file changed, 12 insertions(+), 1 deletion(-)

diff --git a/ta/pkcs11/src/object.c b/ta/pkcs11/src/object.c
index ba3be7a7133a..470eeb2479cc 100644
--- a/ta/pkcs11/src/object.c
+++ b/ta/pkcs11/src/object.c
@@ -840,12 +840,23 @@ enum pkcs11_rc entry_get_attribute_value(struct pkcs11_client *cli
    for (; cur < end; cur += len) {
        struct pkcs11_attribute_head *cli_ref = (void *)cur;
        struct pkcs11_attribute_head cli_head = { };
+       uintptr_t cli_end = 0;
        void *data_ptr = NULL;

+       if ((char *)(cli_ref + 1) > end) {
+           rc = PKCS11_CKR_ARGUMENTS_BAD;
+           goto out;
+       }
```

```
+
        /* Make copy of header so that is aligned properly. */
        TEE_MemMove(&cli_head, cli_ref, sizeof(cli_head));

-       len = sizeof(*cli_ref) + cli_head.size;
+       if (ADD_OVERFLOW(sizeof(*cli_ref), cli_head.size, &len) ||
+           ADD_OVERFLOW((uintptr_t)cur, len, &cli_end) ||
+           (char *)cli_end > end) {
+           rc = PKCS11_CKR_ARGUMENTS_BAD;
+           goto out;
+       }

        /* Treat hidden attributes as missing attributes */
        if (attribute_is_hidden(&cli_head)) {
--
2.43.0
```

```
From 6eb41cb89c3e8da38c4746f31ffaa64190bcaa63 Mon Sep 17 00:00:00 2001
From: Etienne Carriere <etienne.carriere@st.com>
Date: Wed, 21 Jan 2026 14:03:26 +0100
Subject: [PATCH 3/3] ta: pkcs11: fix attribute output size if too small on get
 attribute value

Correct the size field output value for attributes fetched with
PKCS11_CMD_GET_ATTRIBUTE_VALUE where a too short buffer was provided.
As per the PKCS#11 specification, in such case, the related attributes
size field should be filled with CK_UNAVAILABLE_INFORMATION and the
function to return an non-true-error code like CKR_BUFFER_TOO_SMALL.
The implementation complied for the return value but was loading the
required attribute data value size instead in CK_UNAVAILABLE_INFORMATION
in the attribute size field.

Fixes: 783c1515c2f9 ("ta: pkcs11: Add support for getting object size and attribute valu
Signed-off-by: Etienne Carriere <etienne.carriere@st.com>
Reviewed-by: Jens Wiklander <jens.wiklander@linaro.org>
---
 ta/pkcs11/src/object.c | 5 ++++-
 1 file changed, 4 insertions(+), 1 deletion(-)

diff --git a/ta/pkcs11/src/object.c b/ta/pkcs11/src/object.c
index 470eeb2479cc..ed2ce2a956fb 100644
--- a/ta/pkcs11/src/object.c
+++ b/ta/pkcs11/src/object.c
@@ -900,8 +900,11 @@ enum pkcs11_rc entry_get_attribute_value(struct pkcs11_client *clie
            attr_type_invalid = 1;
            break;
        case PKCS11_CKR_BUFFER_TOO_SMALL:
-           if (data_ptr)
+           if (data_ptr) {
+               cli_head.size =
+                   PKCS11_CK_UNAVAILABLE_INFORMATION;
                buffer_too_small = 1;
+           }
            break;
        default:
```

```
            rc = PKCS11_CKR_GENERAL_ERROR;
--
2.43.0
```

Workarounds

N/A

Reported by

Integrity : Arm / Trusted Firmware

Timeline

2025-12-23 Report received
2026-01-08 Confirmed security issue and severity
2026-02-04: Create a fix and review it
2026-02-26: Propose disclosure date and share with stakeholders
2026-02-26: Request CVE
2026-04-23: Publish fix
2026-04-23: Publish advisory

Severity

High 8.7 / 10

CVSS v3 base metrics

Attack vector	Local
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	Low

Learn more about base metrics

CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L

CVE ID

CVE-2026-33317

Weaknesses

- CWE-125
- CWE-787