Instantly share code, notes, and snippets.

scmdcs / **CVE-2025-44951**

Created 8 months ago

Code Revisions 1 Stars 1

CVE-2025-44951

CVE-2025-44951

```
[CVE ID]
CVE-2025-44951

----------------------------------------

[Description]
A missing length check in `ogs_pfcp_dev_add` function from PFCP
library, used by both smf and upf in open5gs 2.7.2 and earlier, allows
a local attacker to cause a Buffer Overflow by changing the
`session.dev` field with a value with length greater than 32.

----------------------------------------

[Vulnerability Type]
Buffer Overflow

----------------------------------------

[Vendor of Product]
open5gs

----------------------------------------

[Affected Product Code Base]
open5gs - 2.7.2

----------------------------------------

[Affected Component]
smf, upf, open5gs-smfd, open5gs-upfd

----------------------------------------

```

```
[Attack Type]
Local

----------------------------------------

[CVE Impact]
Buffer overflow

----------------------------------------

[Attack Vectors]
the attacker must be able to change the configuration file of the affected network functio

----------------------------------------

[Reference]
https://github.com/open5gs/open5gs/issues/3775

----------------------------------------

[Has vendor confirmed or acknowledged the vulnerability?]
true

----------------------------------------

[Discoverer]
Leonardo Sagratella, Lorenzo Cannella
```

writeup.md

CVE-2025-44951 and CVE-2025-44952

Vulnerable products

Open5GS SMF, UPF v2.7.2

Steps to reproduce

We used Flawfinder for static code analysis and discovered two potential buffer overflows in the PFCP library. The vulnerable function, `ogs_pfcp_subnet_add`, is used by SMF, UPF, SGWU, and SGWC components, making all of them potentially vulnerable. We verified this vulnerability only with SMF and UPF.

The overflow occurs due to missing length checks for the `dnn` and `dev` fields when copying them from configuration files. With sufficiently large input, it's possible to overflow these buffers. One possible fix is to use a more secure function such as `snprintf`, `strcpy_s`, or `strlcpy`.

To trigger the overflow:

- You need a string longer than 32 characters to overflow the `dev` field
- You need a string longer than 101 characters to overflow the `dnn` field

For easier testing, we used a `dnn` string with a length of 368 characters to overflow the `num_of_range` field as well, making it easier to demonstrate. This is a portion of the configuration we used:

```
smf:
   ...
    session:
        - subnet: 10.45.0.0/16
          gateway: 10.45.0.1
          dnn:
internetinternetinternetinternetinternetinternetinternetinternetinternetinter
          dev: ogstunogstunogstunogstunogstunogstun
```

To verify that the buffer overflow actually occurs, we modified the source code. Here is the diff file:

```
diff --git a/lib/pfcp/context.c b/lib/pfcp/context.c
index 5f5c3c050..da9b7ca60 100644
--- a/lib/pfcp/context.c
+++ b/lib/pfcp/context.c
@@ -2347,7 +2347,9 @@ ogs_pfcp_dev_t *ogs_pfcp_dev_add(const char *ifname)
     ogs_assert(dev);
     memset(dev, 0, sizeof *dev);

+    printf("pre overflow: ifname %s | fd %d\n", dev->ifname, dev->fd);
     strcpy(dev->ifname, ifname);
+    printf("post overflow: ifname %s | fd %d\n", dev->ifname, dev->fd);

     ogs_list_add(&self.dev_list, dev);

@@ -2452,8 +2454,11 @@ ogs_pfcp_subnet_t *ogs_pfcp_subnet_add(
         ogs_assert(rv == OGS_OK);
     }

-    if (dnn)
```

```
+    if (dnn) {
+        printf("pre overflow: dnn %s | family %d\n", subnet->dnn, subnet-
>family);
         strcpy(subnet->dnn, dnn);
+        printf("post overflow: dnn %s | family %d\n", subnet->dnn, subnet-
>family);
+    }

     ogs_pool_init(&subnet->pool, ogs_app()->pool.sess);
```