

open5gs bug report4

☆ 0 stars    🍴 0 forks

☆ Star

🔔 Notifications

<> Code

🕒 Issues

🔗 Pull requests

🎬 Actions

📁 Projects

🛡 Security

📈 Insights

🔑 main ▾

Go to file



ToughRunner Update README.md ...

on Oct 10 ⌚ 4

[View code](#)

☰ README.md

# Open5gs - A memory leak in PFCP protocol processing crashes SMF causing DoS

Recently, we discovered a logic vulnerability that may cause Open5gs SMF to crash during a code audit of Open5gs Ver2.4.11. The specific causes of the vulnerability are as follows:

## Vulnerability description

When processing PFCP packet, a memory leak in SMF `src/smf/pfcp-path.c` from open5gs causing a DoS vulnerability.

## SMF pfcp-path

Function `pfcp_rcv_cb` from `src/smf/pfcp-path.c` will be called when receiving pfcp connection.

```
src/smf/pfcp-path.c
```

```
static void pfcp_rcv_cb(short when, ogs_socket_t fd, void *data)
{
```

...

`pfc_p_node` will be allocated by calling `ogs_pfc_p_node_add`.

`src/smf/pfc_p-path.c`

```
node = ogs_pfc_p_node_find(&ogs_pfc_p_self()->pfc_p_peer_list, &from);
if (!node) {
    node = ogs_pfc_p_node_add(&ogs_pfc_p_self()->pfc_p_peer_list, &from);
    ogs_assert(node);

    node->sock = data;
    pfc_p_node_fsm_init(node, false);
}
...
```

`pfc_p_node` is allocated from `ogs_pfc_p_node_pool` and appended to `pfc_p_peer_list` in `ogs_pfc_p_node_add`.

`lib/pfc_p/context.c`

```
ogs_pfc_p_node_t *ogs_pfc_p_node_new(ogs_sockaddr_t *sa_list)
{
    ogs_pfc_p_node_t *node = NULL;

    ogs_assert(sa_list);

    ogs_pool_alloc(&ogs_pfc_p_node_pool, &node);
    ogs_assert(node);
    memset(node, 0, sizeof(ogs_pfc_p_node_t));

    node->sa_list = sa_list;

    ogs_list_init(&node->local_list);
    ogs_list_init(&node->remote_list);

    ogs_list_init(&node->gtpu_resource_list);

    return node;
}

ogs_pfc_p_node_t *ogs_pfc_p_node_add(
    ogs_list_t *list, ogs_sockaddr_t *addr)
{
    ogs_pfc_p_node_t *node = NULL;
```

```

ogs_sockaddr_t *new = NULL;

ogs_assert(list);
ogs_assert(addr);

ogs_assert(OGS_OK == ogs_copyaddrinfo(&new, addr));
node = ogs_pfcpc_node_new(new);

ogs_assert(node);
memcpy(&node->addr, new, sizeof node->addr);

ogs_list_add(list, node);

return node;
}

```

Instead of freeing the nodes after using or encountering an error, these nodes are freed only after the termination of SMF by calling function `ogs_pfcpc_context_final`.

So making more than 64 pfcpc connections will crash the SMF causing DoS.

## ogs\_pfcpc\_node\_pool

The size of `ogs_pfcpc_node_pool` is defined as 64.

`lib/app/ogs-context.c`

```

#define MAX_NUM_OF_UE          1024    /* Num of UEs */
#define MAX_NUM_OF_PEER        64      /* Num of Peer */

self.max.ue = MAX_NUM_OF_UE;
self.max.peer = MAX_NUM_OF_PEER;

static void recalculate_pool_size(void)
{
    ...
    self.pool.nf = self.max.peer;
    ...
}

```

`lib/pfcpc/context.c`

```

ogs_pool_init(&ogs_pfcpc_node_pool, ogs_app()->pool.nf);

```

# POC

The vulnerability can be triggered simply by sending more than 64 invalid pfcp packets through different sockets.

```
smf | 09/16 08:41:19.713: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:48304 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:19.713: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:20.713: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:35267 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:20.713: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:21.635: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:50210 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:21.635: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:22.779: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:47470 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:22.779: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:23.693: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:55092 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:23.693: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:24.733: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:52768 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:24.733: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:25.862: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:55457 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:25.862: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
osmohlr | 09/16 08:41:25.862: DLINP *****
osmohlr | 09/16 08:41:25.862: DLINP *****
smf | 09/16 08:41:26.929: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:45686 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:26.929: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:27.775: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:48187 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:27.775: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:28.876: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:60195 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:28.876: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
scscf | 5(40) DEBUG: ims_dialog [dlg_handlers.c:1923]: print_all_dlgcs(): ***** 5(40) DEBUG: ims_dialog [dlg_handlers.c:1924]: print_all_dlgcs(): printing 4096 dialogs
scscf | 5(40) DEBUG: ims_dialog [dlg_handlers.c:1934]: print_all_dlgcs(): ***** 5(40) DEBUG: ims_auth [authorize.c:187]: reg_wait_timer(): Looking for expired/useless at 2740544
scscf | 5(40) DEBUG: ims_auth [authorize.c:232]: reg_wait_timer(): [DONE] looking for expired/useless at 2740544
smf | 09/16 08:41:29.937: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:51956 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:29.937: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:31.006: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:41237 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:31.006: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:32.108: [pfcp] INFO: ogs_pfcp_connect() [172.22.0.1]:47664 (../lib/pfcp/path.c:61)
smf | 09/16 08:41:32.108: [smf] WARNING: cannot handle PFCP message type[1] (../src/smf/pfcp-sm.c:140)
smf | 09/16 08:41:33.043: [pfcp] FATAL: ogs_pfcp_node_new: Assertion 'node' failed. (../lib/pfcp/context.c:642)
smf | 09/16 08:41:33.043: [core] FATAL: backtrace() returned 9 addresses (../lib/core/ogs-abort.c:37)
smf | /openSgs/install/lib/x86_64-linux-gnu/libogs_pfcp.so.2(ogs_pfcp_node_new+0x1b3) [0x7f4cd6b5783c]
smf | /openSgs-smfd(0x644b5) [0x564587f944b5] [0x7f4cd6b57b98]
smf | /openSgs/install/lib/x86_64-linux-gnu/libogscore.so.2(+0x24c43) [0x7f4cd7555c43]
smf | /openSgs-smfd(0x6f5b) [0x564587f95f5b]
smf | /openSgs/install/lib/x86_64-linux-gnu/libogscore.so.2(+0x117e5) [0x7f4cd75427e5]
smf | /lib/x86_64-linux-gnu/libpthread.so.0(+0x8609) [0x7f4cd6a15609]
smf | /lib/x86_64-linux-gnu/libc.so.6(clone+0x43) [0x7f4cd6930133]
smf | /openSgs-init.sh: line 96: 47 Aborted (core dumped) /openSgs-smfd
pcrf | 09/16 08:41:33.260: [diam] ERROR: pid:PSM/smf.epc.mnc001.mcc001.3gppnetwork.org in fd_psm_change_state@p_psm.c:284: 'STATE_OPEN' -> 'STATE_CLOSED' 'smf.epc.mnc001.mcc001.3gppnetwork.org'
pcrf | ((null):0)
smf exited with code 134
```

## Update

We have reported this vulnerability to the vendor through email at 19 Sep 2022, but this bug has not been fixed yet.

## Acknowledgment

Credit to @ToughRunner,@HenryzhaoH,@leonW7 from Shanghai Jiao Tong University.

### Releases

No releases published

### Packages

No packages published