



With the fix applied, the payload triggers slab-out-of-bounds. As that specific one is not considered harmful, no additional CVE is assigned, but it is fixed in "wifi: cfg80211: ensure length byte is present before access".

#### # CVE-2022-42719

This vulnerability was introduced in v5.2-rc1.

With the patch for CVE-2022-41674 and the one mentioned prior applied, the same payload triggers use-after-frees, such as the following:

```
=====
BUG: KASAN: use-after-free in ieee80211_update_bss_from_elems (net/mac80211/scan.c:104)
Read of size 1 at addr ffff88800befa00a by task ksoftirqd/1/20
***
```

#### # CVE-2022-42720

This vulnerability was introduced in v5.1-rc1.

After receiving the attached frames, the kernel log looks like that:

```
***
=====
BUG: KASAN: use-after-free in cfg80211_inform_bss_frame_data (net/wireless/scan.c:2536)
Read of size 8 at addr ffff888008d04478 by task ksoftirqd/1/20
***
```

Its patch fixes a root cause for at least four UAFs and other different memory issues, including:

```
***
BUG: KASAN: use-after-free in cmp_bss+0x856/0x920
Read of size 8 at addr ffff88801459a068 by task ksoftirqd/0/14
```

```
BUG: KASAN: use-after-free in cfg80211_inform_single_bss_data+0xe08/0xea0
Read of size 8 at addr ffff888016272c40 by task ksoftirqd/0/14
```

```
BUG: KASAN: use-after-free in cfg80211_put_bss+0x261/0x270
Read of size 8 at addr ffff8880162b4248 by task ksoftirqd/0/14
```

```
general protection fault, probably for non-canonical address 0xdffffc0200000005: 0000 [#1] PREEMPT SMP
KASAN PTI
KASAN: probably user-memory-access in range [0x0000001000000028-0x000000100000002f]
```

```
general protection fault, probably for non-canonical address 0xdffffc0000000001: 0000 [#1] PREEMPT SMP
KASAN PTI
KASAN: null-ptr-deref in range [0x0000000000000008-0x000000000000000f]
```

```
general protection fault, probably for non-canonical address 0xf99995999999999a: 0000 [#1] PREEMPT SMP
KASAN PTI
KASAN: maybe wild-memory-access in range [0xffffffffcccccd0-0xffffffffcccccd7]
***
```

#### # CVE-2022-42721

This vulnerability was introduced in v5.1-rc1 and leads to an endless loop, leading to a DoS. This is the related kernel log:

```
***
watchdog: BUG: soft lockup - CPU#0 stuck for 52s! [ksoftirqd/0:14]
***
```

#### # CVE-2022-42722

For this, a P2P device is required. This is e.g. default behavior when running NetworkManager to the best of my knowledge. If there is no P2P device yet, it must be created for the reproduction:

```
***
ip 1 set wlan0 up
iw wlan0 interface add p2p0 type __p2pdev addr 02:00:00:00:00:00
iw wdev 0x2 p2p start
***
```

Running the PoC leads to a null-ptr-dereference and thus to a DoS:

```
***
general protection fault, probably for non-canonical address 0xdffffc0000000064: 0000 [#1] PREEMPT SMP
```

KASAN PTI  
KASAN: null-ptr-deref in range [0x0000000000000320-0x0000000000000327]  
```\n

For more details, please see the full logs for each CVE attached.

Best  
Sönke from SEEMOO @ TU Darmstadt

**View attachment "[CVE-2022-41674-decoded.log](#)" of type "text/x-log" (5387 bytes)**

**View attachment "[CVE-2022-42719-decoded.log](#)" of type "text/x-log" (7840 bytes)**

**View attachment "[CVE-2022-42720-decoded.log](#)" of type "text/x-log" (8413 bytes)**

**View attachment "[CVE-2022-42721-decoded.log](#)" of type "text/x-log" (7153 bytes)**

**View attachment "[CVE-2022-42722-decoded.log](#)" of type "text/x-log" (7813 bytes)**

**Download attachment "[CVE-2022-41674.pcap](#)" of type "application/vnd.tcpdump.pcap" (1110 bytes)**

**Download attachment "[CVE-2022-42720.pcap](#)" of type "application/vnd.tcpdump.pcap" (1472 bytes)**

**Download attachment "[CVE-2022-42721.pcap](#)" of type "application/vnd.tcpdump.pcap" (1472 bytes)**

**Download attachment "[CVE-2022-42722.pcap](#)" of type "application/vnd.tcpdump.pcap" (94 bytes)**

[Powered by blists](#) - [more mailing lists](#)

Please check out the [Open Source Software Security Wiki](#), which is counterpart to this [mailing list](#).

Confused about [mailing lists](#) and their use? [Read about mailing lists on Wikipedia](#) and check out these [guidelines on proper formatting of your messages](#).

