

229

size_t-to-int vulnerability in exFAT leads to memory corruption via malformed USB flash drives

Share:



SUMMARY BY PLAYSTATION



Summary

A heap-based buffer overflow can be triggered by a malformed exFAT USB flash drive.

Vulnerability

The vulnerability is in Sony's exFAT implementation where there is an integer truncation from 64bit to 32bit on a size variable that is used to allocate the up-case table:

Code 381 Bytes

```
1 int UVFAT_readupcasetable(void *unused, void *fileSystem) {
2     ...
3     size_t dataLength = *(size_t *)(upcaseEntry + 24);
4     size_t size = sectorSize + dataLength - 1;
5     size = size - size % sectorSize;
6     uint8_t *data = sceFatfsCreateHeapV1(0, size);
7     ...
8     while (1) {
9         ...
10        UVFAT_ReadDevice(fileSystem, offset, sectorSize, data);
11        ...
12        data += sectorSize;
13        ...
14    }
15 }
```

Namely, `dataLength` and `size` are both 64bit wide, however the `size` argument of `sceFatfsCreateHeapV1()` is 32bit wide:

```
3 }
```

When using a big size for `dataLength`, this function will therefore only allocate a small buffer, and as a result overflow and corrupt subsequent objects on the heap when calling `UVFAT_ReadDevice()`.

For example, using `sectorSize=0x200` and `dataLength=0x100000200` we have:

Code 216 Bytes

```
1 size = (sectorSize + dataLength - 1) - (sectorSize + dataLength - 1) % sectorSi
2 <=> size = (0x200 + 0x100000200 - 1) - (0x200 + 0x100000200 - 1) % 0x200;
3 <=> size = 0x1000003FF - 0x1FF;
4 <=> size = 0x100000200;
```

When passing this size to `sceFatfsCreateHeapV1()`, the leading 1 is cut off to `0x200`.

Exploitation

This vulnerability allows us to allocate any buffer on the heap with size ≥ 512 and multiple of 512, and allows us to overflow by a multiple of 512. There are interesting objects that one could spray on the heap such as `struct usb_endpoint` which contain interesting pointers that one could corrupt.

Impact

Jailbreak the PS4/PS5 by plugging in the USB and directly getting kernel code execution.

TIMELINE



[theflow0](#) submitted a report to [PlayStation](#).

Sep 15th (about 1 year ago)

[hacker-01](#)

[PlayStation staff](#) posted a comment.

Sep 15th (about 1 year ago)

[theflow0](#)

posted a comment.

Sep 16th (about 1 year ago)

[hacker-01](#)

[PlayStation staff](#) changed the status to ● **Triaged**.

Sep 22nd (about 1 year ago)

[PlayStation](#)

rewarded [theflow0](#) with a \$10,000 bounty.

Oct 1st (about 1 year ago)

- theflow0 posted a comment. Updated Oct 1st (about 1 year ago)
- hacker-01 posted a comment. Oct 1st (about 1 year ago)
- theflow0 posted a comment. Oct 2nd (about 1 year ago)
- hacker-01 posted a comment. Oct 5th (about 1 year ago)
- theflow0 posted a comment. Oct 8th (about 1 year ago)
- theflow0 posted a comment. Dec 2nd (12 months ago)
- shoshin_cup changed the status to **Needs more info.** Dec 7th (12 months ago)
- theflow0 changed the status to **New.** Updated Dec 7th (12 months ago)
- shoshin_cup closed the report and changed the status to **Resolved.** Dec 7th (12 months ago)
- theflow0 requested to disclose this report. Dec 7th (12 months ago)
- theflow0 posted a comment. Feb 1st (10 months ago)
- theflow0 posted a comment. Mar 12th (9 months ago)
- theflow0 posted a comment. Sep 21st (2 months ago)
- hacker-01 posted a comment. Sep 21st (2 months ago)
- hacker-01 agreed to disclose this report. Sep 21st (2 months ago)
- This report has been disclosed. Sep 21st (2 months ago)

