brackeen / ok-file-formats Public

Code Issues Pull requests Actions Projects Security Insights

New issue

Jump to bottom

heap-buffer-overflow in function ok_jpg_generate_huffman_table() at ok_jpg.c:403 #11

Closed **NISL-SecurityGroup** opened this issue on Mar 5, 2021 · 1 comment

NISL-SecurityGro... commented on Mar 5, 2021 • edited

Version

dev version, git clone https://github.com/brackeen/ok-file-formats.git

Environment

Ubuntu 18.04, 64bit

Testcase

```
#include <stdio.h>
#include <stdlib.h>
#include "ok_jpg.h"
#include "ok_jpg.c"

int main(int _argc, char **_argv) {
    FILE *file = fopen("_argv[1]", "rb");
    ok_jpg image = ok_jpg_read(file, OK_JPG_COLOR_FORMAT_RGBA);
    fclose(file);
    if (image.data) {
        printf("Got image! Size: %li x %li\n", (long)image.width, (long)image.height);
        free(image.data);
    }
    return 0;
}
```

Command

```
$ gcc -g -o main main.c ok_jpg.h
$ ./main heap-buffer-overflow-1.jpg
```

Result

```
Got image! Size: 0 x 0
```

Although the results of the running are correct, when I used our vulnerability detection tool to detect, I found that a heap buffer overflow occurred in line 403. Looking Description for a detailed description.

Description

When I used gdb for debugging with the following command：

```
(gdb) b 1989
(gdb) p decoder
$1 = (ok_jpg_decoder *) 0x55555575e490
(gdb) p sizeof(ok_jpg_decoder)
$2 = 52376
```

Obtaining the start address and size of the **decoder** with the help of the above command， which explaining that the valid address range of the **decoder** is in [0x55555575e490,0x55555576b128].

```
(gdb) b 403
(gdb) p &huff->code[j - 1]
$2 = (uint16_t *) 0x55555576b2d2
```

It can be analyzed from the code context that **huff** points to decoder， and the address **0x55555576b2d2** which is accessed by **huff** is not in the valid range of **[0x55555575e490,0x55555576b128]**. So heap buffer overflow occurs in function ok_jpg_generate_huffman_table() at ok_jpg.c:403.

Note: You can use ASAN for more direct verification.

Poc

Poc file is this.

brackeen added a commit that referenced this issue on Mar 6, 2021

`ok_jpg: Fix invalid DHT (#11)` a9cc171

brackeen commented on Mar 6, 2021 Owner

Thanks for the report, **@NISL-SecurityGroup**. This is now fixed.

brackeen closed this as completed on Mar 6, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants