New issue

Jump to bottom

error: Incorrect handling of function 'zzip_fread' return value #68

Closed   N3vv opened this issue on Mar 5, 2019 · 7 comments

N3vv commented on Mar 5, 2019

Hello, I found a bug of zziplib on the lastest commit `b7747bc` . It's in the function unzzip_cat_file (unzzipcat-zip.c:37) , and it is caused by incorrect handling of the return value of the function 'zzip_fread'.

Relevant code in function unzzip_cat_file in unzzipcat-zip.c:

```
static void unzzip_cat_file(ZZIP_DIR* disk, char* name, FILE* out)
{
    ZZIP_FILE* file = zzip_file_open (disk, name, 0);
    if (file)
    {
        char buffer[1024]; int len;
        while ((len = zzip_file_read (file, buffer, 1024)))
       // Incorrect handling of the return value (-1) of the function zzip_file_read causes an infinite loop.
        {
            fwrite (buffer, 1, len, out);
        }

        zzip_file_close (file);
    }
}
```

POC.zip

Using the POC file, I find that the function zzip_file_read returns -1. And it is handled incorrectly in the caller (unzzip_cat_file), which leads to an infinite loop.

```
zzip_ssize_t
zzip_file_read(ZZIP_FILE * fp, void *buf, zzip_size_t len)
{
            ……
            startlen = fp->d_stream.total_out;
            err = inflate(&fp->d_stream, Z_NO_FLUSH);

            if (err == Z_STREAM_END)
                { fp->restlen = 0; }
            else if (err == Z_OK)
                { fp->restlen -= (fp->d_stream.total_out - startlen); }
            else
                { dir->errcode = err; return -1; }  // Return -1 when there is an error.
          ……
}
```

gdraheim added a commit that referenced this issue on Jan 4, 2021

`#68 ssize_t return value of zzip_file_read is a signed value being po…` … `ac9ae39`

gdraheim added a commit that referenced this issue on Jan 4, 2021

`#68 return value of zzip_mem_disk_fread is signed` `7e78654`

gdraheim added a commit that referenced this issue on Jan 4, 2021

`#68 return value of zzip_entry_fread is signed` `d453977`

gdraheim added a commit that referenced this issue on Jan 4, 2021

`#68 return value of zzip_mem_disk_fread is signed` `0a9db9d`

gdraheim added a commit that referenced this issue on Jan 4, 2021

`#68 return value of zzip_fread is signed` `a34a96f`

gdraheim added a commit that referenced this issue on Jan 4, 2021

`#68 return value of zzip_entry_fread is signed` `fa1f78a`

gdraheim added a commit that referenced this issue on Jan 4, 2021

`#68 return value of posix read(2) is signed` `f7a6fa9`

StayPirate commented on Jun 21, 2021

@gdraheim can we consider this issue fixed?

N3vv commented on Jun 21, 2021   Author

Yes, I think it's been fixed.

StayPirate commented on Jun 21, 2021

If your PoC doesn't hang in a infinite loop anymore I think you're ok to close this issue

kirotawa commented on Jun 21, 2021

CVE-2020-18442 is assigned for this issue

tongxiaoge1001 commented on Jun 24, 2021

Has CVE-2020-18442 been fixed? I find that the patch has not been incorporated into the mainline branch.

StayPirate commented on Jun 24, 2021

All the above commits are both part of `master` and tag `v0.13.72` . Could you share the link to the patch you can't find in the master branch?

tongxiaoge1001 commented on Jun 24, 2021

> All the above commits are both part of `master` and tag `v0.13.72` . Could you share the link to the patch you can't find in the master branch?

Sorry, it was an oversight on my part. So has CVE-2020-18442 been fixed? I find that the issue is still open.

N3vv closed this as completed on Jul 3, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants