New issue

Jump to bottom

Deallocation of control->suffix corrupts Heap Memory #216

Closed pietroborrello opened this issue on Feb 24 · 5 comments

pietroborrello commented on Feb 24

The `suffix` field in the `static rzip_control` structure is initialized to point to global memory in initialize_control

lrzip/lrzip.c
Line 1341 in `64eb4a8`

```
1341	control->suffix = ".lrz";
```

and in the lrzip main.

lrzip/main.c
Line 496 in `6a1600b`

```
496	control->suffix = optarg;
```

However the field is then treated as a heap allocated variable while freeing the `rzip_control` variable. Both in `rzip_control_free`

lrzip/rzip.c
Line 1269 in `465afe8`

```
1269	dealloc(control->suffix);
```

and when setting a new suffix

lrzip/liblrzip.c
Line 439 in `465afe8`

```
439	dealloc(lr->control->suffix);
```

Impact

Corrupting the heap state may result in an exploitable vulnerability, especially if initialized with `optarg` that points to global RW memory.

Fix

It is sufficient to initialize `control->suffix` using the return value of a `strdup` of the strings.

pete4abw commented on Feb 24 Contributor

Good grief! This has been around since v0.1 and rzip before, even before I became involved (v0.19). The initialise function should be used for setting constants or like-size variables, like compression level, etc. Setting control->suffix to equal `optarg` is probably a mistake if there will be recursion. I think the dealloc of suffix is incorrect too. It does not need to be. HOWEVER, the ability to pipe input to `lrzip` sort of makes recursion obsolete and unnecessary. strdup will work and I'll see about implementing it in `lrzip-next` . Thank you

pietroborrello commented on Feb 25 Author

Great, thank you! Will checkout `lrzip-next`

pietroborrello closed this as completed on Feb 25

ckolivas reopened this on Feb 25

ckolivas commented on Feb 25 Owner

Fixed in master.

ckolivas closed this as completed on Feb 25

pete4abw mentioned this issue on Feb 25

#216 may have other issues #217

Closed

carnil commented on Apr 16

Retrospective note: This seems to have been a CVE assigned, which is CVE-2022-28044.

utkarsh2102 commented on May 12

Hello, is there a simple reproducer for this one?

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

5 participants