New issue

Jump to bottom

Exploitable Stack Overflow #103

Closed **dbastone** opened this issue on Sep 3 · 6 comments · Fixed by #104

dbastone commented on Sep 3 • edited Contributor

The unnamed function at 0x80bb148 is used to copy data into a buffer and lacks a destination length check. This function is called in two places - by `process_fmt()` and `fmt_cell_combine()`. The call by `process_fmt()` is reachable using a `w3r_format` element (0x13) in a wk3 file, where user-controlled data from the file is copied into a stack variable. The call by `fmt_cell_combine()` was not investigated.

```
ushort process_fmt(byte *buf,ushort buflen,ushort param_3)
{
[...]
  char local_404 [1024];
[...]
    uVar3 = FUN_080bb148(local_404,buf + 4,buflen - 4);
                          dst       src     len
```

Both `buf` and `buflen` are controllable. The included exploit demonstrates this by overwriting the return address to point to a `jmp esp` gadget, where the payload causes the process to exit with a return value of 3 (I had originally intended to submit this to BGGP3, but missed the deadline!)

A pull request will be provided containing a proposed fix.

Base64 encoded exploit.wk3 - `AAAFAAAQBAAREwAVAAAAAAAD+/v7+/zMzMzO8yhIIQM2AMw==`
[edit: reduced exploit size from 38 to 34 bytes]

This was discovered using Ghidra and AFL++'s QEMU mode, and was inspired by this tweet.

5

dbastone mentioned this issue on Sep 3

Reimplementation of function at 0x80bb148 that prevents overflowing the destination buffer #104

Merged

taviso commented on Sep 3 — Owner

This is incredible, thank you for the thorough bug report!

I'm away from my workstation until the morning, I'll build a new release tomorrow. I think there should also be an advisory to commemorate, this has to be some sort of record 😂

taviso added a commit that referenced this issue on Sep 4

`add tests for #103` `22f6c3a`

taviso closed this as completed in #104 on Sep 4

---

taviso commented on Sep 4 — Owner

I've confirmed everything you've said. I think the `__builtin_return_address()` solution is clever, but maybe just capping it at 256 will work... at least all the tests seem to pass, and it's a bit cleaner 🤦

I guess I'll try that and see if something breaks.

Everything else looks good to me, I'll build a new release.

I think I'll send an advisory, let's see if mitre will assign a CVE lol

dbastone commented on Sep 4 — Contributor, Author

Sounds good, thank you!

taviso commented on Sep 4 — Owner

All done. Thank you so much... and a quality patch and exploit, impressive and hilarious 😆

❤️ 1

taviso commented on Sep 4 — Owner

123ADV-001: Stack Buffer Overflow in Lotus 1-2-3 R3 for UNIX/Linux

About

The 123 command is a spreadsheet application for UNIX-based systems that can be used in interactive mode to create and modify financial and scientific models.

For more information, see https://123r3.net

Advisory

A stack buffer overflow was reported in the cell format processing routines. If a victim opens an untrusted malicious worksheet, code execution could occur. This vulnerability has been resolved by adding additional bounds checks.

There have been no reports of this vulnerability being exploited in the wild.

We take your security very seriously, in fact, this is the first known vulnerability reported in Lotus 1-2-3 R3 since it's release in September 1990.

Credit

This issue was reported to the 123elf project by **@dbastone**.

Solution

A new release has been prepared to resolve this issue, we recommend affected users upgrade urgently.

https://github.com/taviso/123elf/releases

Lotus 1-2-3 releases for other platforms are affected, but are not actively maintained. Users are advised to migrate to Linux to continue receiving updates.

😄 19 ❤️ 3

poinine commented on Sep 11

:d

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Reimplementation of function at 0x80bb148 that prevents overflowing the destination buffer
dbastone/123elf

3 participants