

[New issue](#)[Jump to bottom](#)

segmentation fault when execute_command and the stack overflow caused by parameters #40

Open firmianay opened this issue on Jul 5 · 5 comments**Labels** security

firmianay commented on Jul 5

hi, great project!

I think it's better to limit the size of `res`, otherwise it may cause the program to crash, such as performing `"cat /dev/random | od -x"`, which maybe unlikely in reality.

```
char *execute_command(char *command) {
    FILE *fp;
    char *res = calloc(4096, sizeof(char));
    char buf[1024];

    fp = popen(command, "r");
    if (fp == NULL) {
        perror("Failed to run command");
        return NULL;
    }

    while (fgets(buf, sizeof(buf), fp) != NULL) {
        strcat(res, buf);
    }
    // printf("RESULT OF COMMAND: %s\n", res);

    pclose(fp);
    return res;
}
```

0xjet commented on Jul 5

Collaborator

You're right. There is surely more instances like this one. Tagging this as a security bug to be fixed at some point. Thanks!

 Oxjet added the **security** label on Jul 5

firmianay commented on Jul 5

Author

Well, there are other security issues. There is no limit to the length of program parameters, which may cause overflow.

src/client/client.c

```
void main(int argc, char* argv[]){
...
    int opt;
    char dest_address[32];
    char path_arg[512];

    while ((opt = getopt(argc, argv, ":S:c:e:u:a:p:s:h")) != -1) {
        switch (opt) {
            case 'S':
...
                strcpy(dest_address, optarg);
```



  firmianay changed the title ~~segmentation fault when execute_command~~ segmentation fault when execute_command and the stack overflow caused by parameters on Jul 5

Ifex370 commented on Jul 18

Is this the reason I get

Illegal instruction (core dumped) - when I run ./simple_timer. and a
segmentation fault (core dumped) - when I run ./simple_open?

I have not been able to carry out a PoC due to the above errors.

  h3xduck mentioned this issue on Jul 18

Library injection path error: Segfault simple_timer and simple_open #44

 Open

h3xduck commented on Jul 18 • edited ▼

Owner

@Ifex370 I am moving your issue to a different thread ([#44](#)) since it is not related to this security-related issue

firmianay commented on Aug 3 • edited ▼

Author

<https://nvd.nist.gov/vuln/detail/CVE-2022-35505>

<https://nvd.nist.gov/vuln/detail/CVE-2022-35506>

Discoverer: Chao Yang@Li Auto

Assignees

No one assigned

Labels

security

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants

