New issue

Jump to bottom

Null Pointer Dereference / Crash #30

Closed martinclauss opened this issue on May 21, 2020 · 1 comment

martinclauss commented on May 21, 2020

Hi!

I found a bug that crashes the forked child pre-authenticated. The details are as follows:

Triggering the Bug

```
root@5829efcb25f7:/opt/uftpd# uftpd -v
2.11
root@5829efcb25f7:/opt/uftpd# uftpd -n /tmp

root@5829efcb25f7:/opt# nc 127.0.0.1 21
220 uftpd (2.11) ready.
CWD /../../../../../../../etc/passwd

root@5829efcb25f7:/opt/uftpd# dmesg
...
[40553.625757] uftpd[75111]: segfault at 0 ip 00007f6806a484e5 sp 00007ffda5d8c1c8 error 4 in libc-2.31.so[7f68068e2000+178000]
[40553.625771] Code: 00 00 0f 1f 00 31 c0 c5 f8 77 c3 66 2e 0f 1f 84 00 00 00 00 00 f3 0f 1e fa 89 f9 48 89 fa c5 f9 ef c0 83 e1 3f 83 f9 20 77 2b <c5> fd 74 0f c5 fd d7 c1 85 c0 0f 8
```

output of GDB session attached to `uftpd -n /tmp` :

```
[Attaching after process 2761 fork to child process 2776]
[New inferior 2 (process 2776)]
[Detaching after fork from parent process 2761]
[Inferior 1 (process 2761) detached]

Thread 2.1 "uftpd" received signal SIGSEGV, Segmentation fault.
[Switching to process 2776]
__strlen_avx2 () at ../sysdeps/x86_64/multiarch/strlen-avx2.S:65
65      ../sysdeps/x86_64/multiarch/strlen-avx2.S: No such file or directory.
(gdb) x/i $Pc
Value can't be converted to integer.
(gdb) x/i $pc
=> 0x7f3898d354e5 <__strlen_avx2+21>:   vpcmpeqb (%rdi),%ymm0,%ymm1
(gdb) disass
Dump of assembler code for function __strlen_avx2:
   0x00007f3898d354d0 <+0>:     endbr64
   0x00007f3898d354d4 <+4>:     mov    %edi,%ecx
   0x00007f3898d354d6 <+6>:     mov    %rdi,%rdx
   0x00007f3898d354d9 <+9>:     vpxor  %xmm0,%xmm0,%xmm0
   0x00007f3898d354dd <+13>:    and    $0x3f,%ecx
   0x00007f3898d354e0 <+16>:    cmp    $0x20,%ecx
   0x00007f3898d354e3 <+19>:    ja     0x7f3898d35510 <__strlen_avx2+64>
=> 0x00007f3898d354e5 <+21>:    vpcmpeqb (%rdi),%ymm0,%ymm1
```

registers

```
(gdb) i r
rax            0x556208658f59      93879536029529
rbx            0x556208bd0550      93879541761360
rcx            0x0                 0
rdx            0x0                 0
rsi            0x7ffcf35cb960      140724391426400
rdi            0x0                 0
...
```

`rdi` is 0

the stack trace is as follows:

```
(gdb) where
#0  __strlen_avx2 () at ../sysdeps/x86_64/multiarch/strlen-avx2.S:65
#1  0x0000556208652768 in handle_CWD (ctrl=0x556208bd0550, path=<optimized out>) at ftpcmd.c:407
#2  0x0000556208654c37 in read_client_command (w=<optimized out>, arg=0x556208bd0550, events=<optimized out>) at ftpcmd.c:1586
#3  0x00007f3898daad37 in uev_run (ctx=0x556208bd0520, flags=flags@entry=0) at uev.c:415
#4  0x0000556208656a96 in ftp_command (ctrl=0x556208bd0550) at ftpcmd.c:1610
#5  ftp_session (ctx=<optimized out>, sd=<optimized out>) at ftpcmd.c:1652
#6  0x00007f3898daad37 in uev_run (ctx=0x7ffcf35ccf60, flags=0) at uev.c:415
#7  0x0000556208650597 in serve_files (ctx=0x7ffcf35ccf60) at uftpd.c:247
#8  main (argc=<optimized out>, argv=<optimized out>) at uftpd.c:405
```

the relevant source code in `ftpcmd.c`

```
static void handle_CWD(ctrl_t *ctrl, char *path)
{
        struct stat st;
        char *dir;

        if (!path)
                goto done;
```

```
        /*
         * Some FTP clients, most notably Chrome, use CWD to check if an
         * entry is a file or directory.
         */
        dir = compose_abspath(ctrl, path);
        if (!dir || stat(dir, &st) || !S_ISDIR(st.st_mode)) {
                DBG("chrooted:%d, ctrl->cwd: %s, home:%s, dir:%s, len:%zd, dirlen:%zd",
                    chrooted, ctrl->cwd, home, dir, strlen(home), strlen(dir));
                send_msg(ctrl->sd, "550 No such directory.\r\n");
                return;
        }
        // ...
```

`if (!dir ...` so `dir` might be `NULL` but will be used in the `DBG()` macro with a call to `strlen(dir)`. As can be seen from the stack trace `dir == $rdi` and is `NULL (0)` at this point which causes the crash. To force `dir == NULL` we can look at `compose_abspath` in `common.c` which calls `compose_path` here `ptr = compose_path(ctrl, path);` and `return ptr;` returns the pointer at the end of the function:

```
char *compose_abspath(ctrl_t *ctrl, char *path)
{
        char *ptr;
        char cwd[sizeof(ctrl->cwd)];

        if (path && path[0] == '/') {
                strlcpy(cwd, ctrl->cwd, sizeof(cwd));
                memset(ctrl->cwd, 0, sizeof(ctrl->cwd));
        }

        ptr = compose_path(ctrl, path);

        if (path && path[0] == '/')
                strlcpy(ctrl->cwd, cwd, sizeof(ctrl->cwd));

        return ptr;
}
```

So we have to look at `compose_path` in `common.h`. The trace that leads to `dir == NULL` is the following:

```
time
|  char dir[PATH_MAX] = { 0 };
|  strlcpy(dir, ctrl->cwd, sizeof(dir));
|  DBG("Compose path from cwd: %s, arg: %s", ctrl->cwd, path ?: "");
|  if (!path || !strlen(path))
|        if (path[0] != '/') {
|        strlcat(dir, path, sizeof(dir));
|  while ((ptr = strstr(dir, "//")))
|  if (!chrooted) {
|  if (!stat(dir, &st) && S_ISDIR(st.st_mode)) {
|        name = basename(path);
|        ptr = dirname(dir);
|        memset(rpath, 0, sizeof(rpath));
|        if (!realpath(ptr, rpath)) {
|               INFO("Failed realpath(%s): %m", ptr);
v               return NULL;
```

Best
Martin

troglobit closed this as completed in 5c3b201 on May 22, 2020

troglobit commented on May 22, 2020 — Owner

Nice catch! Fix pushed to master in 5c3b201, with a few more cleanups of this class of error handling preceding that commit.

1

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

2 participants