

[U-Boot] [PATCH 3/5] CVE-2019-13104: ext4: check for underflow in ext4fs_read_file

Paul Emge [paulemge at forallsecure.com](mailto:paulemge@forallsecure.com)

Mon Jul 8 23:37:05 UTC 2019

- Previous message: [\[U-Boot\] \[PATCH 2/5\] CVE-2019-13105: ext4: fix double-free in ext4_cache_read](#)
- Next message: [\[U-Boot\] \[PATCH 3/5\] CVE-2019-13104: ext4: check for underflow in ext4fs_read_file](#)
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

in ext4fs_read_file, it is possible for a broken/malicious file system to cause a memcpy of a negative number of bytes, which overflows all memory. This patch fixes the issue by checking for a negative length.

Signed-off-by: Paul Emge <[paulemge at forallsecure.com](mailto:paulemge@forallsecure.com)>

```

---
 fs/ext4/ext4fs.c | 8 +++++--
 1 file changed, 5 insertions(+), 3 deletions(-)

diff --git a/fs/ext4/ext4fs.c b/fs/ext4/ext4fs.c
index 85dc122f30..e2b740cac4 100644
--- a/fs/ext4/ext4fs.c
+++ b/fs/ext4/ext4fs.c
@@ -66,13 +66,15 @@ int ext4fs_read_file(struct ext2fs_node *node, loff_t pos,
    ext_cache_init(&cache);
-   if (blocksize <= 0)
-       return -1;
-
    /* Adjust len so it we can't read past the end of the file. */
    if (len + pos > filesize)
        len = (filesize - pos);
+   if (blocksize <= 0 || len <= 0) {
+       ext_cache_fini(&cache);
+       return -1;
+   }
    blockcnt = lldiv(((len + pos) + blocksize - 1), blocksize);
    for (i = lldiv(pos, blocksize); i < blockcnt; i++) {
--
2.20.1

```

- Previous message: [\[U-Boot\] \[PATCH 2/5\] CVE-2019-13105: ext4: fix double-free in ext4_cache_read](#)
- Next message: [\[U-Boot\] \[PATCH 3/5\] CVE-2019-13104: ext4: check for underflow in ext4fs_read_file](#)
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

[More information about the U-Boot mailing list](#)