author	Dominique Martinet <asmadeus@codewreck.org>	2025-03-19 20:20:15 +0900
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2025-05-02 07:47:04 +0200
commit	468ff4a7c61fb811c596a7c44b6a5455e40fd12b (patch)	
tree	03b260eb763f24b76a4261c8e0d04437c5d72fdc	
parent	43b498a8452d613ae6339809ab4ac49abc299bd4 (diff)	
download	linux-468ff4a7c61fb811c596a7c44b6a5455e40fd12b.tar.gz	

9p/net: fix improper handling of bogus negative read/write replies

```
[ Upstream commit d0259a856afca31d699b706ed5e2adf11086c73b ]

In p9_client_write() and p9_client_read_once(), if the server
incorrectly replies with success but a negative write/read count then we
would consider written (negative) <= rsize (positive) because both
variables were signed.

Make variables unsigned to avoid this problem.

The reproducer linked below now fails with the following error instead
of a null pointer deref:
9pnet: bogus RWRITE count (4294967295 > 3)

Reported-by: Robert Morris <rtm@mit.edu>
Closes: https://lore.kernel.org/16271.1734448631@26-5-164.dynamic.csail.mit.edu
Message-ID: <20250319-9p_unsigned_rw-v3-1-71327f1503d0@codewreck.org>
Reviewed-by: Christian Schoenebeck <linux_oss@crudebyte.com>
Signed-off-by: Dominique Martinet <asmadeus@codewreck.org>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r-- net/9p/client.c 30

1 files changed, 16 insertions, 14 deletions

```
diff --git a/net/9p/client.c b/net/9p/client.c
index e876d6fea2fc44..e89a91802e038f 100644
--- a/net/9p/client.c
+++ b/net/9p/client.c
@@ -1545,7 +1545,8 @@ p9_client_read_once(struct p9_fid *fid, u64 offset, struct iov_iter *to,
        struct p9_client *clnt = fid->clnt;
        struct p9_req_t *req;
        int count = iov_iter_count(to);
-       int rsize, received, non_zc = 0;
+       u32 rsize, received;
+       bool non_zc = false;
        char *dataptr;

        *err = 0;
@@ -1568,7 +1569,7 @@ p9_client_read_once(struct p9_fid *fid, u64 offset, struct iov_iter *to,
                                       0, 11, "dqd", fid->fid,
                                       offset, rsize);
        } else {
-               non_zc = 1;
+               non_zc = true;
                req = p9_client_rpc(clnt, P9_TREAD, "dqd", fid->fid, offset,
```

```
                                     rsize);
        }
@@ -1589,11 +1590,11 @@ p9_client_read_once(struct p9_fid *fid, u64 offset, struct iov_iter *to,
                return 0;
        }
        if (rsize < received) {
-               pr_err("bogus RREAD count (%d > %d)\n", received, rsize);
+               pr_err("bogus RREAD count (%u > %u)\n", received, rsize);
                received = rsize;
        }

-       p9_debug(P9_DEBUG_9P, "<<< RREAD count %d\n", received);
+       p9_debug(P9_DEBUG_9P, "<<< RREAD count %u\n", received);

        if (non_zc) {
                int n = copy_to_iter(dataptr, received, to);
@@ -1620,9 +1621,9 @@ p9_client_write(struct p9_fid *fid, u64 offset, struct iov_iter *from, int *err)
        *err = 0;

        while (iov_iter_count(from)) {
-               int count = iov_iter_count(from);
-               int rsize = fid->iounit;
-               int written;
+               size_t count = iov_iter_count(from);
+               u32 rsize = fid->iounit;
+               u32 written;

                if (!rsize || rsize > clnt->msize - P9_IOHDRSZ)
                        rsize = clnt->msize - P9_IOHDRSZ;
@@ -1630,7 +1631,7 @@ p9_client_write(struct p9_fid *fid, u64 offset, struct iov_iter *from, int *err)
                if (count < rsize)
                        rsize = count;

-               p9_debug(P9_DEBUG_9P, ">>> TWRITE fid %d offset %llu count %d (/%d)\n",
+               p9_debug(P9_DEBUG_9P, ">>> TWRITE fid %d offset %llu count %u (/%zu)\n",
                         fid->fid, offset, rsize, count);

                /* Don't bother zerocopy for small IO (< 1024) */
@@ -1656,11 +1657,11 @@ p9_client_write(struct p9_fid *fid, u64 offset, struct iov_iter *from, int *err)
                        break;
                }
                if (rsize < written) {
-                       pr_err("bogus RWRITE count (%d > %d)\n", written, rsize);
+                       pr_err("bogus RWRITE count (%u > %u)\n", written, rsize);
                        written = rsize;
                }

-               p9_debug(P9_DEBUG_9P, "<<< RWRITE count %d\n", written);
+               p9_debug(P9_DEBUG_9P, "<<< RWRITE count %u\n", written);

                p9_req_put(clnt, req);
                iov_iter_revert(from, count - written - iov_iter_count(from));
@@ -2056,7 +2057,8 @@ EXPORT_SYMBOL_GPL(p9_client_xattrcreate);

 int p9_client_readdir(struct p9_fid *fid, char *data, u32 count, u64 offset)
 {
-       int err, rsize, non_zc = 0;
+       int err, non_zc = 0;
+       u32 rsize;
        struct p9_client *clnt;
        struct p9_req_t *req;
        char *dataptr;
@@ -2065,7 +2067,7 @@ int p9_client_readdir(struct p9_fid *fid, char *data, u32 count, u64 offset)

        iov_iter_kvec(&to, ITER_DEST, &kv, 1, count);
```

```
-       p9_debug(P9_DEBUG_9P, ">>> TREADDIR fid %d offset %llu count %d\n",
+       p9_debug(P9_DEBUG_9P, ">>> TREADDIR fid %d offset %llu count %u\n",
                 fid->fid, offset, count);

        err = 0;
@@ -2101,11 +2103,11 @@ int p9_client_readdir(struct p9_fid *fid, char *data, u32 count, u64 offset)
                goto free_and_error;
        }
        if (rsize < count) {
-               pr_err("bogus RREADDIR count (%d > %d)\n", count, rsize);
+               pr_err("bogus RREADDIR count (%u > %u)\n", count, rsize);
                count = rsize;
        }

-       p9_debug(P9_DEBUG_9P, "<<< RREADDIR count %d\n", count);
+       p9_debug(P9_DEBUG_9P, "<<< RREADDIR count %u\n", count);

        if (non_zc)
                memmove(data, dataptr, count);
```