author	Dan Carpenter <error27@gmail.com>	2022-11-15 16:16:43 +0300
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2022-11-26 09:27:37 +0100
commit	c2a00b149836d60c222930bbea6b2139caf34d4f (patch)	
tree	0e8c45598a2d5e56f2f61244f538c663ac524bd9	
parent	f8926e2d2225eb7b7e11cd3fa266aaad9075b767 (diff)	
download	linux-c2a00b149836d60c222930bbea6b2139caf34d4f.tar.gz	

diff options

context: 3

space: include

mode: unified

drbd: use after free in drbd_create_device()

```
[ Upstream commit a7a1598189228b5007369a9622ccdf587be0730f ]

The drbd_destroy_connection() frees the "connection" so use the _safe()
iterator to prevent a use after free.

Fixes: b6f85ef9538b ("drbd: Iterate over all connections")
Signed-off-by: Dan Carpenter <error27@gmail.com>
Reviewed-by: Christoph Böhmwalder <christoph.boehmwalder@linbit.com>
Link: https://lore.kernel.org/r/Y3Jd5iZRbNQ9w6gm@kili
Signed-off-by: Jens Axboe <axboe@kernel.dk>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r-- drivers/block/drbd/drbd_main.c 4

1 files changed, 2 insertions, 2 deletions

```
diff --git a/drivers/block/drbd/drbd_main.c b/drivers/block/drbd/drbd_main.c
index f3e4db16fd07bb..8532b839a3435c 100644
--- a/drivers/block/drbd/drbd_main.c
+++ b/drivers/block/drbd/drbd_main.c
@@ -2672,7 +2672,7 @@ static int init_submitter(struct drbd_device *device)
 enum drbd_ret_code drbd_create_device(struct drbd_config_context *adm_ctx, unsigned int minor)
 {
        struct drbd_resource *resource = adm_ctx->resource;
-       struct drbd_connection *connection;
+       struct drbd_connection *connection, *n;
        struct drbd_device *device;
        struct drbd_peer_device *peer_device, *tmp_peer_device;
        struct gendisk *disk;
@@ -2789,7 +2789,7 @@ enum drbd_ret_code drbd_create_device(struct drbd_config_context *adm_ctx, unsig
        return NO_ERROR;

 out_idr_remove_from_resource:
-       for_each_connection(connection, resource) {
+       for_each_connection_safe(connection, n, resource) {
                peer_device = idr_remove(&connection->peer_devices, vnr);
                if (peer_device)
                        kref_put(&connection->kref, drbd_destroy_connection);
```