author	Damien Le Moal <damien.lemoal@opensource.wdc.com>	2023-03-06 10:13:13 +0900
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2023-03-22 13:37:50 +0100
commit	8ed9813871038b25a934b21ab76b5b7dbf44fc3a (patch)	
tree	7e97b7cd86ea53db0004b05557070772a7e6ba97	
parent	e40a30a96252a47b5840e8844db5d5f347e30f32 (diff)	
download	linux-8ed9813871038b25a934b21ab76b5b7dbf44fc3a.tar.gz	

nvmet: avoid potential UAF in nvmet_req_complete()

```
[ Upstream commit 6173a77b7e9d3e202bdb9897b23f2a8afe7bf286 ]

An nvme target ->queue_response() operation implementation may free the
request passed as argument. Such implementation potentially could result
in a use after free of the request pointer when percpu_ref_put() is
called in nvmet_req_complete().

Avoid such problem by using a local variable to save the sq pointer
before calling __nvmet_req_complete(), thus avoiding dereferencing the
req pointer after that function call.

Fixes: a07b4970f464 ("nvmet: add a generic NVMe target")
Signed-off-by: Damien Le Moal <damien.lemoal@opensource.wdc.com>
Reviewed-by: Chaitanya Kulkarni <kch@nvidia.com>
Signed-off-by: Christoph Hellwig <hch@lst.de>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r-- drivers/nvme/target/core.c 4

1 files changed, 3 insertions, 1 deletions

```
diff --git a/drivers/nvme/target/core.c b/drivers/nvme/target/core.c
index f66ed13d7c11de..3935165048e741 100644
--- a/drivers/nvme/target/core.c
+++ b/drivers/nvme/target/core.c
@@ -756,8 +756,10 @@ static void __nvmet_req_complete(struct nvmet_req *req, u16 status)

 void nvmet_req_complete(struct nvmet_req *req, u16 status)
 {
+       struct nvmet_sq *sq = req->sq;
+
        __nvmet_req_complete(req, status);
-       percpu_ref_put(&req->sq->ref);
+       percpu_ref_put(&sq->ref);
 }
 EXPORT_SYMBOL_GPL(nvmet_req_complete);
```