author	Alexey Nepomnyashih <sdl@nppct.ru>	2025-04-17 12:21:17 +0000
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2025-05-02 07:46:56 +0200
commit	5b83d30c63f9964acb1bc63eb8e670b9e0d2c240 (patch)	
tree	02d984dc8868628fa568c77f2e4555e838128028	
parent	baa332e22f4f58618e934888a7c1fd02f35e80bd (diff)	
download	linux-5b83d30c63f9964acb1bc63eb8e670b9e0d2c240.tar.gz	

xen-netfront: handle NULL returned by xdp_convert_buff_to_frame()

commit cc3628dcd851ddd8d418bf0c897024b4621ddc92 upstream.

The function xdp_convert_buff_to_frame() may return NULL if it fails
to correctly convert the XDP buffer into an XDP frame due to memory
constraints, internal errors, or invalid data. Failing to check for NULL
may lead to a NULL pointer dereference if the result is used later in
processing, potentially causing crashes, data corruption, or undefined
behavior.

On XDP redirect failure, the associated page must be released explicitly
if it was previously retained via get_page(). Failing to do so may result
in a memory leak, as the pages reference count is not decremented.

Cc: stable@vger.kernel.org # v5.9+
Fixes: 6c5aa6fc4def ("xen networking: add basic XDP support for xen-netfront")
Signed-off-by: Alexey Nepomnyashih <sdl@nppct.ru>
Link: https://patch.msgid.link/20250417122118.1009824-1-sdl@nppct.ru
Signed-off-by: Jakub Kicinski <kuba@kernel.org>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

Diffstat

-rw-r--r-- drivers/net/xen-netfront.c 17

1 files changed, 12 insertions, 5 deletions

```
diff --git a/drivers/net/xen-netfront.c b/drivers/net/xen-netfront.c
index 8425226c09f0d9..69ef50fb2e1b73 100644
--- a/drivers/net/xen-netfront.c
+++ b/drivers/net/xen-netfront.c
@@ -985,20 +985,27 @@ static u32 xennet_run_xdp(struct netfront_queue *queue, struct page *pdata,
        act = bpf_prog_run_xdp(prog, xdp);
        switch (act) {
        case XDP_TX:
-               get_page(pdata);
                xdpf = xdp_convert_buff_to_frame(xdp);
+               if (unlikely(!xdpf)) {
+                       trace_xdp_exception(queue->info->netdev, prog, act);
+                       break;
+               }
+               get_page(pdata);
                err = xennet_xdp_xmit(queue->info->netdev, 1, &xdpf, 0);
-               if (unlikely(!err))
```

```
+                if (unlikely(err <= 0)) {
+                        if (err < 0)
+                                trace_xdp_exception(queue->info->netdev, prog, act);
                         xdp_return_frame_rx_napi(xdpf);
-                else if (unlikely(err < 0))
-                        trace_xdp_exception(queue->info->netdev, prog, act);
+                }
                 break;
         case XDP_REDIRECT:
                 get_page(pdata);
                 err = xdp_do_redirect(queue->info->netdev, xdp, prog);
                 *need_xdp_flush = true;
-                if (unlikely(err))
+                if (unlikely(err)) {
                         trace_xdp_exception(queue->info->netdev, prog, act);
+                        xdp_return_buff(xdp);
+                }
                 break;
         case XDP_PASS:
         case XDP_DROP:
```