author	Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>	2022-11-07 10:21:40 -0800
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2022-11-25 17:42:19 +0100
commit	5d53797ce7ce8fb1d95a5bebc5efa9418c4217a3 (patch)	
tree	4541497a44205c2d096f043adc569e55a929dac7	
parent	89c0c27ab39a854b7f16fbe382a41354434c8e7d (diff)	
download	linux-5d53797ce7ce8fb1d95a5bebc5efa9418c4217a3.tar.gz	

diff options

context: 3

space: include

mode: unified

Input: iforce - invert valid length check when fetching device IDs

```
commit b8ebf250997c5fb253582f42bfe98673801ebebd upstream.

syzbot is reporting uninitialized value at iforce_init_device() [1], for
commit 6ac0aec6b0a6 ("Input: iforce - allow callers supply data buffer
when fetching device IDs") is checking that valid length is shorter than
bytes to read. Since iforce_get_id_packet() stores valid length when
returning 0, the caller needs to check that valid length is longer than or
equals to bytes to read.

Reported-by: syzbot <syzbot+4dd880c1184280378821@syzkaller.appspotmail.com>
Signed-off-by: Tetsuo Handa <penguin-kernel@I-love.SAKURA.ne.jp>
Fixes: 6ac0aec6b0a6 ("Input: iforce - allow callers supply data buffer when fetching device IDs")
Link: https://lore.kernel.org/r/531fb432-7396-ad37-ecba-3e42e7f56d5c@I-love.SAKURA.ne.jp
Cc: stable@vger.kernel.org
Signed-off-by: Dmitry Torokhov <dmitry.torokhov@gmail.com>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

Diffstat

-rw-r--r--	drivers/input/joystick/iforce/iforce-main.c	8

1 files changed, 4 insertions, 4 deletions

```
diff --git a/drivers/input/joystick/iforce/iforce-main.c b/drivers/input/joystick/iforce/iforce-main.c
index b86de1312512bd..84b87526b7ba31 100644
--- a/drivers/input/joystick/iforce/iforce-main.c
+++ b/drivers/input/joystick/iforce/iforce-main.c
@@ -273,22 +273,22 @@ int iforce_init_device(struct device *parent, u16 bustype,
  * Get device info.
  */

-       if (!iforce_get_id_packet(iforce, 'M', buf, &len) || len < 3)
+       if (!iforce_get_id_packet(iforce, 'M', buf, &len) && len >= 3)
                input_dev->id.vendor = get_unaligned_le16(buf + 1);
        else
                dev_warn(&iforce->dev->dev, "Device does not respond to id packet M\n");

-       if (!iforce_get_id_packet(iforce, 'P', buf, &len) || len < 3)
+       if (!iforce_get_id_packet(iforce, 'P', buf, &len) && len >= 3)
                input_dev->id.product = get_unaligned_le16(buf + 1);
        else
                dev_warn(&iforce->dev->dev, "Device does not respond to id packet P\n");

-       if (!iforce_get_id_packet(iforce, 'B', buf, &len) || len < 3)
+       if (!iforce_get_id_packet(iforce, 'B', buf, &len) && len >= 3)
```

```
                iforce->device_memory.end = get_unaligned_le16(buf + 1);
        else
                dev_warn(&iforce->dev->dev, "Device does not respond to id packet B\n");

-       if (!iforce_get_id_packet(iforce, 'N', buf, &len) || len < 2)
+       if (!iforce_get_id_packet(iforce, 'N', buf, &len) && len >= 2)
                ff_effects = buf[1];
        else
                dev_warn(&iforce->dev->dev, "Device does not respond to id packet N\n");
```