



author Reka Norman <rekanorman@chromium.org> 2023-02-27 13:49:38 +1100
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2023-03-30 12:51:28 +0200
commit [d3ce3afd9f791dd1b7daedfcf8c396b60af5dec0](#) (patch)
tree [17c832c538b6260b7ad874272e67a5783156d7c9](#)
parent [7dea60d60f80af011fbfea493049bfe19afe9911](#) (diff)
download [linux-d3ce3afd9f791dd1b7daedfcf8c396b60af5dec0.tar.gz](#)

diff options

context:

3 ▼

space:

include ▼

mode:

unified ▼

HID: intel-ish-hid: ipc: Fix potential use-after-free in work function

[Upstream commit 8ae2f2b0a28416ed2f6d8478ac8b9f7862f36785]

When a reset notify IPC message is received, the ISR schedules a work function and passes the ISHTP device to it via a global pointer `ishtp_dev`. If `ish_probe()` fails, the devm-managed device resources including `ishtp_dev` are freed, but the work is not cancelled, causing a use-after-free when the work function tries to access `ishtp_dev`. Use `devm_work_autocancel()` instead, so that the work is automatically cancelled if probe fails.

Signed-off-by: Reka Norman <rekanorman@chromium.org>

Acked-by: Srinivas Pandruvada <srinivas.pandruvada@linux.intel.com>

Signed-off-by: Jiri Kosina <jkosina@suse.cz>

Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r--	drivers/hid/intel-ish-hid/ipc/ipc.c	9
------------	---	---

1 files changed, 8 insertions, 1 deletions

diff --git a/drivers/hid/intel-ish-hid/ipc/ipc.c b/drivers/hid/intel-ish-hid/ipc/ipc.c
index 15e14239af829b..a49c6affd7c4c4 100644

--- a/drivers/hid/intel-ish-hid/ipc/ipc.c

+++ b/drivers/hid/intel-ish-hid/ipc/ipc.c

@@ -5,6 +5,7 @@

*** Copyright (c) 2014-2016, Intel Corporation.**

***/**

+#include <linux/devm-helpers.h>

#include <linux/sched.h>

#include <linux/spinlock.h>

#include <linux/delay.h>

@@ -621,7 +622,6 @@ static void recv_ipc(struct ishtp_device *dev, uint32_t doorbell_val)

case MNG_RESET_NOTIFY:

if (!ishtp_dev) {

ishtp_dev = dev;

- INIT_WORK(&fw_reset_work, fw_reset_work_fn);

}

schedule_work(&fw_reset_work);

break;

@@ -940,6 +940,7 @@ struct ishtp_device *ish_dev_init(struct pci_dev *pdev)

```

{
    struct ishtp_device *dev;
    int    i;
+   int    ret;

    dev = devm_kzalloc(&pdev->dev,
                      sizeof(struct ishtp_device) + sizeof(struct ish_hw),
@@ -975,6 +976,12 @@ struct ishtp_device *ish_dev_init(struct pci_dev *pdev)
        list_add_tail(&tx_buf->link, &dev->wr_free_list);
    }

+   ret = devm_work_autocancel(&pdev->dev, &fw_reset_work, fw_reset_work_fn);
+   if (ret) {
+       dev_err(dev->devc, "Failed to initialise FW reset work\n");
+       return NULL;
+   }
+
    dev->ops = &ish_hw_ops;
    dev->devc = &pdev->dev;
    dev->mtu = IPC_PAYLOAD_SIZE - sizeof(struct ishtp_msg_hdr);

```