tNFA_HCI_EVT_DATA buffer overflow crash

Assigned Bug P3 [AOSP] assigned adexe s nau

STATUS UPDATE No update yet. Edit

DESCRIPTION vn...@gmail.com created issue #1

The `nfaHciCallback` function in https://android.googlesource.com/platform/packages/apps/Nfc/+/refs/heads/master/nci/jni/HciEventManager.cpp will access fields in `eventData` without ch

```
04-12 05:52:24.145 F/DEBUG   (19740): backtrace:
04-12 05:52:24.145 F/DEBUG   (19740):     #00 pc 00019f28  /system/lib/libc.so (__memcpy_base+176)
04-12 05:52:24.145 F/DEBUG   (19740):     #01 pc 0000ede1  /system/lib/libnfc_nci_jni.so (_ZNSt3__16vectorIhNS_9allocatorIhEEEC2IPhEET_NS_9enable_ifIXaasr21_
04-12 05:52:24.145 F/DEBUG   (19740):     #02 pc 0000e689  /system/lib/libnfc_nci_jni.so (HciEventManager::nfaHciCallback(unsigned char, tNFA_HCI_EVT_DATA*)+
04-12 05:52:24.145 F/DEBUG   (19740):     #03 pc 00037f00  /system/lib/libnfc-nci.so (nfa_hciu_send_to_all_apps(unsigned char, tNFA_HCI_EVT_DATA*)+40)
04-12 05:52:24.145 F/DEBUG   (19740):     #04 pc 00034d40  /system/lib/libnfc-nci.so (nfa_hci_sys_disable()+84)
04-12 05:52:24.145 F/DEBUG   (19740):     #05 pc 00044e4c  /system/lib/libnfc-nci.so (nfa_sys_disable_subsystems(bool)+96)
04-12 05:52:24.145 F/DEBUG   (19740):     #06 pc 0001cd78  /system/lib/libnfc-nci.so (nfa_dm_disable(tNFA_DM_MSG*)+228)
04-12 05:52:24.145 F/DEBUG   (19740):     #07 pc 000448f8  /system/lib/libnfc-nci.so (nfa_sys_event(NFC_HDR*)+92)
04-12 05:52:24.145 F/DEBUG   (19740):     #08 pc 00059858  /system/lib/libnfc-nci.so (nfc_task(unsigned int)+280)
04-12 05:52:24.145 F/DEBUG   (19740):     #09 pc 00086b38  /system/lib/libnfc-nci.so (gki_task_entry(void*)+60)
04-12 05:52:24.146 F/DEBUG   (19740):     #10 pc 00063775  /system/lib/libc.so (__pthread_start(void*)+22)
04-12 05:52:24.146 F/DEBUG   (19740):     #11 pc 0001dff9  /system/lib/libc.so (__start_thread+24)
```

```
void HciEventManager::nfaHciCallback(tNFA_HCI_EVT event,
                                     tNFA_HCI_EVT_DATA* eventData) {
  if (eventData == nullptr) {
    return;
  }

  DLOG_IF(INFO, nfc_debug_enabled) << StringPrintf(
      "event=%d code=%d pipe=%d len=%d", event, eventData->rcvd_evt.evt_code,
      eventData->rcvd_evt.pipe, eventData->rcvd_evt.evt_len);

  std::string evtSrc;
  if (eventData->rcvd_evt.pipe == sEsePipe) {
    evtSrc = "eSE1";
  } else if (eventData->rcvd_evt.pipe == sSimPipe) {
    evtSrc = "SIM1";
  } else {
    LOG(WARNING) << "Incorrect Pipe Id";
    return;
  }

  uint8_t* buff = eventData->rcvd_evt.p_evt_buf;
  uint32_t buffLength = eventData->rcvd_evt.evt_len;
  std::vector<uint8_t> event_buff(buff, buff + buffLength);
  // Check the event and check if it contains the AID
  if (event == NFA_HCI_EVENT_RCVD_EVT &&
      eventData->rcvd_evt.evt_code == NFA_HCI_EVT_TRANSACTION &&
      buffLength > 3 && event_buff[0] == 0x81) {
    uint32_t aidlen = event_buff[1];
```

deleted

0 B

Links (2)

Links (2)

" ...faHciCallback function in https://android.googlesource.com/platform/packages/apps/Nfc/+/refs/heads/master/nci/jni/HciEventManager.cpp will access fields in eventData without checking the
"I think this issue starts from the nfa_hci_sys_disable function in https://android.googlesource.com/platform/system/nfc/+/refs/heads/master/src/nfa/hci/nfa_hci_main.cc#664 ."

COMMENTS

vn...@gmail.com <vn...@gmail.com> #2

I think this issue starts from the `nfa_hci_sys_disable` function in https://android.googlesource.com/platform/system/nfc/+/refs/heads/master/src/nfa/hci/nfa_hci_main.cc#664

I think this issue starts from the nfa_hci_sys_disable function in https://android.googlesource.com/platform/system/nfc/+/refs/heads/master/src/nfa/hci/nfa_hci_main.cc#364:

```
static void nfa_hci_sys_disable(void) {
  tNFA_HCI_EVT_DATA evt_data;
  nfa_sys_stop_timer(&nfa_hci_cb.timer);
  if (nfa_hci_cb.conn_id) {
    if (nfa_sys_is_graceful_disable()) {
      /* Tell all applications stack is down */
      if (NFC_GetNCIVersion() == NCI_VERSION_1_0) {
        nfa_hciu_send_to_all_apps(NFA_HCI_EXIT_EVT, &evt_data);
        NFC_ConnClose(nfa_hci_cb.conn_id);
        return;
      }
    }
    nfa_hci_cb.conn_id = 0;
  }
  nfa_hci_cb.hci_state = NFA_HCI_STATE_DISABLED;
  /* deregister message handler on NFA SYS */
  nfa_sys_deregister(NFA_ID_HCI);
}
```

It sends out an event with an uninitialized `tNFA_HCI_EVT_DATA` which contains arbitrary data from the stack. When `nfaHciCallback` then receives this event and tries to create a vector fron

```
uint8_t* buff = eventData->rcvd_evt.p_evt_buf;
uint32_t buffLength = eventData->rcvd_evt.evt_len;
std::vector<uint8_t> event_buff(buff, buff + buffLength);
```

---

ra...@google.com <ra...@google.com>

Assigned to ra...@google.com.

---

ra...@google.com <ra...@google.com> #3

We have shared this with our product and engineering team and will update this issue with more information as it becomes available.