camelot-os / **sentry-kernel** Public

Code | Issues 17 | Pull requests 10 | Actions | Projects | Wiki | Secu

Commit `150b7ed`

PThierry authored last week · ✕ 3 / 16 · Verified

```
Merge pull request #108 from PThierry/fix-irq-gate

fix,cve: fixing incomplete ownership check for IRQ manipulation
```

main (#108) · v0.4.8 v0.4.7

2 parents 1dd2378 + f7450cb commit 150b7ed

3 files changed +16 -1 lines changed

Top

Filter files…

- kernel/src/syscalls
  - sysgate_int_acknowledge.c
  - sysgate_int_disable.c
  - sysgate_int_enable.c

3 files changed +16 -1 lines changed

Search within code

kernel/src/syscalls/sysgate_int_acknowledge.c

```
@@ -39,8 +39,13 @@ stack_frame_t *gate_int_acknowledge(stack_frame_t *frame,
uint16_t IRQn)
39  39          /* user interrupt with no owning task. Should not happen as the kernel
do not hold any IRQ */
40  40          panic(PANIC_KERNEL_INVALID_MANAGER_RESPONSE);
41  41      }
    42  +   if (unlikely(owner != current)) {
    43  +       /* device associated IRQ is not owned by the current task */
    44  +       mgr_task_set_sysreturn(current, STATUS_DENIED);
    45  +       goto end;
```

```
      46  +     }
42    47        /* push the inth event into the task input events queue */
43          -   if (unlikely(mgr_interrupt_acknowledge_irq(IRQn) == K_STATUS_OKAY)) {
      48  +     if (unlikely(mgr_interrupt_acknowledge_irq(IRQn) != K_STATUS_OKAY)) {
44    49            /* should not rise while IRQ ownership has been checked! see dts file */
45    50            panic(PANIC_KERNEL_INVALID_MANAGER_RESPONSE);
46    51        }
```

kernel/src/syscalls/sysgate_int_disable.c

```
@@ -41,6 +41,11 @@ stack_frame_t *gate_int_disable(stack_frame_t *frame,
uint16_t IRQn)
41    41            /* user interrupt with no owning task. Should not happen as the kernel
do not hold any IRQ */
42    42            panic(PANIC_KERNEL_INVALID_MANAGER_RESPONSE);
43    43        }
      44  +     if (unlikely(owner != current)) {
      45  +         /* device associated IRQ is not owned by the current task */
      46  +         mgr_task_set_sysreturn(current, STATUS_DENIED);
      47  +         goto end;
      48  +     }
44    49        /* push the inth event into the task input events queue */
45    50        if (unlikely(mgr_interrupt_disable_irq(IRQn) != K_STATUS_OKAY)) {
46    51            /* should not rise while IRQ ownership has been checked! see dts file */
```

kernel/src/syscalls/sysgate_int_enable.c

```
@@ -41,6 +41,11 @@ stack_frame_t *gate_int_enable(stack_frame_t *frame,
uint16_t IRQn)
41    41            /* user interrupt with no owning task. Should not happen as the kernel
do not hold any IRQ */
42    42            panic(PANIC_KERNEL_INVALID_MANAGER_RESPONSE);
43    43        }
      44  +     if (unlikely(owner != current)) {
      45  +         /* device associated IRQ is not owned by the current task */
      46  +         mgr_task_set_sysreturn(current, STATUS_DENIED);
      47  +         goto end;
      48  +     }
44    49        /* push the inth event into the task input events queue */
```

```
45  50      if (unlikely(mgr_interrupt_enable_irq(IRQn) != K_STATUS_OKAY)) {
46  51          /* should not rise while IRQ ownership has been checked! see dts file */
```

Comments 0