author	Shang XiaoJing <shangxiaojing@huawei.com>	2022-11-17 09:23:46 +0800
committer	Steven Rostedt (Google) <rostedt@goodmis.org>	2022-11-17 18:24:58 -0500
commit	1b5f1c34d3f5a664a57a5a7557a50e4e3cc2505c (patch)	
tree	699d098b9f20e555415a3f1e866c3fb1f1baa6ff	
parent	a4527fef9afe5c903c718d0cd24609fe9c754250 (diff)	
download	linux-1b5f1c34d3f5a664a57a5a7557a50e4e3cc2505c.tar.gz	

tracing: Fix wild-memory-access in register_synth_event()

```
In register_synth_event(), if set_synth_event_print_fmt() failed, then
both trace_remove_event_call() and unregister_trace_event() will be
called, which means the trace_event_call will call
__unregister_trace_event() twice. As the result, the second unregister
will causes the wild-memory-access.

register_synth_event
    set_synth_event_print_fmt failed
    trace_remove_event_call
        event_remove
            if call->event.funcs then
            __unregister_trace_event (first call)
    unregister_trace_event
        __unregister_trace_event (second call)

Fix the bug by avoiding to call the second __unregister_trace_event() by
checking if the first one is called.

general protection fault, probably for non-canonical address
        0xfbd59c0000000024: 0000 [#1] SMP KASAN PTI
KASAN: maybe wild-memory-access in range
[0xdead000000000120-0xdead000000000127]
CPU: 0 PID: 3807 Comm: modprobe Not tainted
6.1.0-rc1-00186-g76f33a7eedb4 #299
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS
rel-1.15.0-0-g2dd4b9b3f840-prebuilt.qemu.org 04/01/2014
RIP: 0010:unregister_trace_event+0x6e/0x280
Code: 00 fc ff df 4c 89 ea 48 c1 ea 03 80 3c 02 00 0f 85 0e 02 00 00 48
b8 00 00 00 00 00 fc ff df 4c 8b 63 08 4c 89 e2 48 c1 ea 03 <80> 3c 02
00 0f 85 e2 01 00 00 49 89 2c 24 48 85 ed 74 28 e8 7a 9b
RSP: 0018:ffff88810413f370 EFLAGS: 00010a06
RAX: dfffc0000000000 RBX: ffff888105d050b0 RCX: 0000000000000000
RDX: 1bd5a00000000024 RSI: ffff888119e276e0 RDI: ffffffff835a8b20
RBP: dead000000000100 R08: 0000000000000000 R09: fffffbfff0913481
R10: ffffffff8489a407 R11: fffffbfff0913480 R12: dead000000000122
R13: ffff888105d050b8 R14: 0000000000000000 R15: ffff888105d05028
FS:  00007f7823e8d540(0000) GS:ffff888119e00000(0000)
knlGS:0000000000000000
CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 00007f7823e7ebec CR3: 000000010a058002 CR4: 0000000000330ef0
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
```

```
DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
Call Trace:
 <TASK>
 __create_synth_event+0x1e37/0x1eb0
 create_or_delete_synth_event+0x110/0x250
 synth_event_run_command+0x2f/0x110
 test_gen_synth_cmd+0x170/0x2eb [synth_event_gen_test]
 synth_event_gen_test_init+0x76/0x9bc [synth_event_gen_test]
 do_one_initcall+0xdb/0x480
 do_init_module+0x1cf/0x680
 load_module+0x6a50/0x70a0
 __do_sys_finit_module+0x12f/0x1c0
 do_syscall_64+0x3f/0x90
 entry_SYSCALL_64_after_hwframe+0x63/0xcd

Link: https://lkml.kernel.org/r/20221117012346.22647-3-shangxiaojing@huawei.com

Fixes: 4b147936fa50 ("tracing: Add support for 'synthetic' events")
Signed-off-by: Shang XiaoJing <shangxiaojing@huawei.com>
Cc: stable@vger.kernel.org
Cc: <mhiramat@kernel.org>
Cc: <zanussi@kernel.org>
Cc: <fengguang.wu@intel.com>
Signed-off-by: Steven Rostedt (Google) <rostedt@goodmis.org>
```

Diffstat

-rw-r--r-- kernel/trace/trace_events_synth.c 5

1 files changed, 2 insertions, 3 deletions

```
diff --git a/kernel/trace/trace_events_synth.c b/kernel/trace/trace_events_synth.c
index e310052dc83ce4..29fbfb27c2b2cc 100644
--- a/kernel/trace/trace_events_synth.c
+++ b/kernel/trace/trace_events_synth.c
@@ -828,10 +828,9 @@ static int register_synth_event(struct synth_event *event)
        }

        ret = set_synth_event_print_fmt(call);
-       if (ret < 0) {
+       /* unregister_trace_event() will be called inside */
+       if (ret < 0)
                trace_remove_event_call(call);
-               goto err;
-       }
  out:
        return ret;
  err:
```