author	Chen Zhongjin <chenzhongjin@huawei.com>	2023-03-09 16:02:30 +0800
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2023-03-22 13:28:09 +0100
commit	ac58b88ccbbb8e9fb83e137cee04a856b1ea6635 (patch)	
tree	e1745fe46f840c99984c7c3ca56e7f7e6f588a3f	
parent	65e4c9a6d0c9a8c81ce75576869d46fff5d7964f (diff)	
download	linux-ac58b88ccbbb8e9fb83e137cee04a856b1ea6635.tar.gz	

ftrace: Fix invalid address access in lookup_rec() when index is 0

```
commit ee92fa443358f4fc0017c1d0d325c27b37802504 upstream.

KASAN reported follow problem:

 BUG: KASAN: use-after-free in lookup_rec
 Read of size 8 at addr ffff000199270ff0 by task modprobe
 CPU: 2 Comm: modprobe
 Call trace:
  kasan_report
  __asan_load8
  lookup_rec
  ftrace_location
  arch_check_ftrace_location
  check_kprobe_address_safe
  register_kprobe

When checking pg->records[pg->index - 1].ip in lookup_rec(), it can get a
pg which is newly added to ftrace_pages_start in ftrace_process_locs().
Before the first pg->index++, index is 0 and accessing pg->records[-1].ip
will cause this problem.

Don't check the ip when pg->index is 0.

Link: https://lore.kernel.org/linux-trace-kernel/20230309080230.36064-1-chenzhongjin@huawei.com

Cc: stable@vger.kernel.org
Fixes: 9644302e3315 ("ftrace: Speed up search by skipping pages by address")
Suggested-by: Steven Rostedt (Google) <rostedt@goodmis.org>
Signed-off-by: Chen Zhongjin <chenzhongjin@huawei.com>
Signed-off-by: Steven Rostedt (Google) <rostedt@goodmis.org>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

Diffstat

-rw-r--r-- kernel/trace/ftrace.c 3

1 files changed, 2 insertions, 1 deletions

```
diff --git a/kernel/trace/ftrace.c b/kernel/trace/ftrace.c
index 61f8c78daf178e..8e3c76dcc0ffef 100644
--- a/kernel/trace/ftrace.c
+++ b/kernel/trace/ftrace.c
@@ -1557,7 +1557,8 @@ unsigned long ftrace_location_range(unsigned long start, unsigned long end)
```

```
        key.flags = end;        /* overload flags, as it is unsigned long */

        for (pg = ftrace_pages_start; pg; pg = pg->next) {
-               if (end < pg->records[0].ip ||
+               if (pg->index == 0 ||
+                   end < pg->records[0].ip ||
                    start >= (pg->records[pg->index - 1].ip + MCOUNT_INSN_SIZE))
                        continue;
                rec = bsearch(&key, pg->records, pg->index,
```