



author Chen Zhongjin <chenzhongjin@huawei.com> 2023-03-09 16:02:30 +0800
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2023-03-22 13:27:12 +0100
commit 7569ee04b0e3b32df79f64db3a7138573edad9bc (patch)
tree 2777099e5c1e38e5b7ffc9607ee2fdc2da643e46
parent 4bc5a4dfb4ccb36b23d6f94340c5b4356b75e9fb (diff)
download [linux-7569ee04b0e3b32df79f64db3a7138573edad9bc.tar.gz](#)

diff options

context: 3
space: include
mode: unified

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 9c779556643629..5c0463dbe16ee4 100644

--- a/kernel/trace/ftrace.c

+++ b/kernel/trace/ftrace.c

@@ -1581,7 +1581,8 @@ 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,
```