
Subject: [PATCH] Invalid return value of execve() resulting in oopses
Posted by [xemul](#) on Thu, 03 May 2007 10:11:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Alexey Kuznetsov <alexey@openvz.org>

When elf loader fails to map executable (due to memory shortage or because binary is malformed), it can return 0. Normally, this is invisible because process is killed with SIGKILL and it never returns to user space.

But if exec() is called from kernel thread (hotplug, whatever) consequences are more interesting and vary depending on architecture.

i386. Nothing especially interesting, execve() just returns with "success" :-)

x86_64. Fake zero frame is used on way to caller, RSP/RIP are loaded with zeros, ergo... double fault.

ia64. Similar to i386, but r32...r95 are corrupted. Sometimes it oopses due to return to zero PC, sometimes it sees NaT in rXX and oopses due to NaT consumption.

Signed-off-by: Alexey Kuznetsov <alexey@openvz.org>

Signed-off-by: Kirill Korotaev <dev@openvz.org>

Signed-off-by: Pavel Emelianov <xemul@openvz.org>

```
diff --git a/fs/binfmt_elf.c b/fs/binfmt_elf.c
index 67d9b31..fa8ea33 100644
--- a/fs/binfmt_elf.c
+++ b/fs/binfmt_elf.c
@@ -871,6 +871,8 @@ static int load_elf_binary(struct linux_
    elf_prot, elf_flags);
    if (BAD_ADDR(error)) {
        send_sig(SIGKILL, current, 0);
+   retval = IS_ERR((void *)error) ?
+   PTR_ERR((void *)error) : -EINVAL;
        goto out_free_dentry;
    }

@@ -900,6 +902,7 @@ static int load_elf_binary(struct linux_
    TASK_SIZE - elf_ppnt->p_memsz < k) {
    /* set_brk can never work. Avoid overflows. */
    send_sig(SIGKILL, current, 0);
+   retval = -EINVAL;
```

```
goto out_free_dentry;  
}
```
