author	Wang Yufen <wangyufen@huawei.com>	2022-11-08 13:11:31 +0800
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2022-11-26 09:24:36 +0100
commit	d4944497827a3d14bc5a26dbcfb7433eb5a956c0 (patch)	
tree	01d2e2558c61c5464564b5eff1c05003eace75e3	
parent	25521fd2e217b03e24609908cdba8bcab186595b (diff)	
download	linux-d4944497827a3d14bc5a26dbcfb7433eb5a956c0.tar.gz	

bpf: Fix memory leaks in __check_func_call

```
[ Upstream commit eb86559a691cea5fa63e57a03ec3dc9c31e97955 ]

kmemleak reports this issue:

unreferenced object 0xffff88817139d000 (size 2048):
  comm "test_progs", pid 33246, jiffies 4307381979 (age 45851.820s)
  hex dump (first 32 bytes):
    01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  backtrace:
    [<0000000045f075f0>] kmalloc_trace+0x27/0xa0
    [<0000000098b7c90a>] __check_func_call+0x316/0x1230
    [<00000000b4c3c403>] check_helper_call+0x172e/0x4700
    [<00000000aa3875b7>] do_check+0x21d8/0x45e0
    [<000000001147357b>] do_check_common+0x767/0xaf0
    [<00000000b5a595b4>] bpf_check+0x43e3/0x5bc0
    [<0000000011e391b1>] bpf_prog_load+0xf26/0x1940
    [<0000000007f765c0>] __sys_bpf+0xd2c/0x3650
    [<000000000839815d6>] __x64_sys_bpf+0x75/0xc0
    [<00000000946ee250>] do_syscall_64+0x3b/0x90
    [<0000000000506b7f>] entry_SYSCALL_64_after_hwframe+0x63/0xcd

The root case here is: In function prepare_func_exit(), the callee is
not released in the abnormal scenario after "state->curframe--;". To
fix, move "state->curframe--;" to the very bottom of the function,
right when we free callee and reset frame[] pointer to NULL, as Andrii
suggested.

In addition, function __check_func_call() has a similar problem. In
the abnormal scenario before "state->curframe++;", the callee also
should be released by free_func_state().

Fixes: 69c087ba6225 ("bpf: Add bpf_for_each_map_elem() helper")
Fixes: fd978bf7fd31 ("bpf: Add reference tracking to verifier")
Signed-off-by: Wang Yufen <wangyufen@huawei.com>
Link: https://lore.kernel.org/r/1667884291-15666-1-git-send-email-wangyufen@huawei.com
Signed-off-by: Martin KaFai Lau <martin.lau@kernel.org>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r-- kernel/bpf/verifier.c 14

1 files changed, 9 insertions, 5 deletions

```
diff --git a/kernel/bpf/verifier.c b/kernel/bpf/verifier.c
index 8a73a165ac7695..cceb29b0585f09 100644
```

```
--- a/kernel/bpf/verifier.c
+++ b/kernel/bpf/verifier.c
@@ -5808,11 +5808,11 @@ static int __check_func_call(struct bpf_verifier_env *env, struct bpf_insn *insn
        /* Transfer references to the callee */
        err = copy_reference_state(callee, caller);
        if (err)
-               return err;
+               goto err_out;

        err = set_callee_state_cb(env, caller, callee, *insn_idx);
        if (err)
-               return err;
+               goto err_out;

        clear_caller_saved_regs(env, caller->regs);

@@ -5829,6 +5829,11 @@ static int __check_func_call(struct bpf_verifier_env *env, struct bpf_insn *insn
                print_verifier_state(env, callee);
        }
        return 0;
+
+err_out:
+       free_func_state(callee);
+       state->frame[state->curframe + 1] = NULL;
+       return err;
 }

 int map_set_for_each_callback_args(struct bpf_verifier_env *env,
@@ -5966,8 +5971,7 @@ static int prepare_func_exit(struct bpf_verifier_env *env, int *insn_idx)
                return -EINVAL;
        }

-       state->curframe--;
-       caller = state->frame[state->curframe];
+       caller = state->frame[state->curframe - 1];
        if (callee->in_callback_fn) {
                /* enforce R0 return value range [0, 1]. */
                struct tnum range = tnum_range(0, 1);
@@ -6006,7 +6010,7 @@ static int prepare_func_exit(struct bpf_verifier_env *env, int *insn_idx)
        }
        /* clear everything in the callee */
        free_func_state(callee);
-       state->frame[state->curframe + 1] = NULL;
+       state->frame[state->curframe--] = NULL;
        return 0;
 }
```