



author Eric Dumazet <edumazet@google.com> 2023-03-07 16:45:30 +0000
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2023-03-17 08:48:57 +0100
commit [f3969427fb06a2c3cd6efd7faab63505cfa76e76](#) (patch)
tree [21962cb27c5aa4e6a20f2f8c742f32dd4d6deed8](#)
parent [a9f99eacf79f3b62653c21902c8ec830c6fdb624](#) (diff)
download [linux-f3969427fb06a2c3cd6efd7faab63505cfa76e76.tar.gz](#)

diff options

context:
space:
mode:

af_unix: fix struct pid leaks in OOB support

[Upstream commit 2aab4b96900272885bc157f8b236abf1cdc02e08]

syzbot reported struct pid leak [1].

Issue is that queue_oob() calls maybe_add_creds() which potentially holds a reference on a pid.

But skb->destructor is not set (either directly or by calling unix_scm_to_skb())

This means that subsequent kfree_skb() or consume_skb() would leak this reference.

In this fix, I chose to fully support scm even for the OOB message.

```
[1]
BUG: memory leak
unreferenced object 0xfffff8881053e7f80 (size 128):
comm "syz-executor242", pid 5066, jiffies 4294946079 (age 13.220s)
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:
[<fffffffff812ae26a>] alloc_pid+0x6a/0x560 kernel/pid.c:180
[<fffffffff812718df>] copy_process+0x169f/0x26c0 kernel/fork.c:2285
[<fffffffff81272b37>] kernel_clone+0xf7/0x610 kernel/fork.c:2684
[<fffffffff812730cc>] __do_sys_clone+0x7c/0xb0 kernel/fork.c:2825
[<fffffffff849ad699>] do_syscall_x64 arch/x86/entry/common.c:50 [inline]
[<fffffffff849ad699>] do_syscall_64+0x39/0xb0 arch/x86/entry/common.c:80
[<fffffffff84a0008b>] entry_SYSCALL_64_after_hwframe+0x63/0xcd
```

Fixes: 314001f0bf92 ("af_unix: Add OOB support")
Reported-by: syzbot+7699d9e5635c10253a27@syzkaller.appspotmail.com
Signed-off-by: Eric Dumazet <edumazet@google.com>
Cc: Rao Shoaib <rao.shoaib@oracle.com>
Reviewed-by: Kuniyuki Iwashima <kuniyu@amazon.com>
Link: <https://lore.kernel.org/r/20230307164530.771896-1-edumazet@google.com>
Signed-off-by: Jakub Kicinski <kuba@kernel.org>
Signed-off-by: Sasha Levin <sasha@kernel.org>

Diffstat

-rw-r--r-- net/unix/af_unix.c 10

1 files changed, 8 insertions, 2 deletions

```

diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 32ddf8fe32c696..a96026dbdf94e3 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -1970,7 +1970,8 @@ out:
     #define UNIX_SKB_FRAGS_SZ (PAGE_SIZE << get_order(32768))

     #if IS_ENABLED(CONFIG_AF_UNIX_OOB)
-    static int queue_oob(struct socket *sock, struct msghdr *msg, struct sock *other)
+    static int queue_oob(struct socket *sock, struct msghdr *msg, struct sock *other,
+        struct scm_cookie *scm, bool fds_sent)
     {
         struct unix_sock *ousk = unix_sk(other);
         struct sk_buff *skb;
@@ -1981,6 +1982,11 @@ static int queue_oob(struct socket *sock, struct msghdr *msg, struct sock *other
         if (!skb)
             return err;

+        err = unix_scm_to_skb(scm, skb, !fds_sent);
+        if (err < 0) {
+            kfree_skb(skb);
+            return err;
+        }
         skb_put(skb, 1);
         err = skb_copy_datagram_from_iter(skb, 0, &msg->msg_iter, 1);

@@ -2108,7 +2114,7 @@ static int unix_stream_sendmsg(struct socket *sock, struct msghdr *msg,

     #if IS_ENABLED(CONFIG_AF_UNIX_OOB)
         if (msg->msg_flags & MSG_OOB) {
-            err = queue_oob(sock, msg, other);
+            err = queue_oob(sock, msg, other, &scm, fds_sent);
             if (err)
                 goto out_err;
             sent++;

```