author	Lizhi Xu <lizhi.xu@windriver.com>	2025-02-24 09:31:39 +0800
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2025-04-20 10:15:11 +0200
commit	11ae4fec1f4b4ee06770a572c37d89cbaecbf66e (patch)	
tree	8120542b3306286377941559b0a5b24c2a7933c9	
parent	89a4db7a67e707533d20649fc4eaa2fcbf29472a (diff)	
download	linux-11ae4fec1f4b4ee06770a572c37d89cbaecbf66e.tar.gz	

PM: hibernate: Avoid deadlock in hibernate_compressor_param_set()

```
[ Upstream commit 52323ed1444ea5c2a5f1754ea0a2d9c8c216ccdf ]

syzbot reported a deadlock in lock_system_sleep() (see below).

The write operation to "/sys/module/hibernate/parameters/compressor"
conflicts with the registration of ieee80211 device, resulting in a deadlock
when attempting to acquire system_transition_mutex under param_lock.

To avoid this deadlock, change hibernate_compressor_param_set() to use
mutex_trylock() for attempting to acquire system_transition_mutex and
return -EBUSY when it fails.

Task flags need not be saved or adjusted before calling
mutex_trylock(&system_transition_mutex) because the caller is not going
to end up waiting for this mutex and if it runs concurrently with system
suspend in progress, it will be frozen properly when it returns to user
space.

syzbot report:

syz-executor895/5833 is trying to acquire lock:
ffffffff8e0828c8 (system_transition_mutex){+.+.}-{4:4}, at: lock_system_sleep+0x87/0xa0 kernel/power/main.c:56

but task is already holding lock:
ffffffff8e07dc68 (param_lock){+.+.}-{4:4}, at: kernel_param_lock kernel/params.c:607 [inline]
ffffffff8e07dc68 (param_lock){+.+.}-{4:4}, at: param_attr_store+0xe6/0x300 kernel/params.c:586

which lock already depends on the new lock.

the existing dependency chain (in reverse order) is:

-> #3 (param_lock){+.+.}-{4:4}:
       __mutex_lock_common kernel/locking/mutex.c:585 [inline]
       __mutex_lock+0x19b/0xb10 kernel/locking/mutex.c:730
       ieee80211_rate_control_ops_get net/mac80211/rate.c:220 [inline]
       rate_control_alloc net/mac80211/rate.c:266 [inline]
       ieee80211_init_rate_ctrl_alg+0x18d/0x6b0 net/mac80211/rate.c:1015
       ieee80211_register_hw+0x20cd/0x4060 net/mac80211/main.c:1531
       mac80211_hwsim_new_radio+0x304e/0x54e0 drivers/net/wireless/virtual/mac80211_hwsim.c:5558
       init_mac80211_hwsim+0x432/0x8c0 drivers/net/wireless/virtual/mac80211_hwsim.c:6910
       do_one_initcall+0x128/0x700 init/main.c:1257
       do_initcall_level init/main.c:1319 [inline]
       do_initcalls init/main.c:1335 [inline]
       do_basic_setup init/main.c:1354 [inline]
       kernel_init_freeable+0x5c7/0x900 init/main.c:1568
       kernel_init+0x1c/0x2b0 init/main.c:1457
       ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:148
       ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244
```

```
-> #2 (rtnl_mutex){+.+.}-{4:4}:
       __mutex_lock_common kernel/locking/mutex.c:585 [inline]
       __mutex_lock+0x19b/0xb10 kernel/locking/mutex.c:730
       wg_pm_notification drivers/net/wireguard/device.c:80 [inline]
       wg_pm_notification+0x49/0x180 drivers/net/wireguard/device.c:64
       notifier_call_chain+0xb7/0x410 kernel/notifier.c:85
       notifier_call_chain_robust kernel/notifier.c:120 [inline]
       blocking_notifier_call_chain_robust kernel/notifier.c:345 [inline]
       blocking_notifier_call_chain_robust+0xc9/0x170 kernel/notifier.c:333
       pm_notifier_call_chain_robust+0x27/0x60 kernel/power/main.c:102
       snapshot_open+0x189/0x2b0 kernel/power/user.c:77
       misc_open+0x35a/0x420 drivers/char/misc.c:179
       chrdev_open+0x237/0x6a0 fs/char_dev.c:414
       do_dentry_open+0x735/0x1c40 fs/open.c:956
       vfs_open+0x82/0x3f0 fs/open.c:1086
       do_open fs/namei.c:3830 [inline]
       path_openat+0x1e88/0x2d80 fs/namei.c:3989
       do_filp_open+0x20c/0x470 fs/namei.c:4016
       do_sys_openat2+0x17a/0x1e0 fs/open.c:1428
       do_sys_open fs/open.c:1443 [inline]
       __do_sys_openat fs/open.c:1459 [inline]
       __se_sys_openat fs/open.c:1454 [inline]
       __x64_sys_openat+0x175/0x210 fs/open.c:1454
       do_syscall_x64 arch/x86/entry/common.c:52 [inline]
       do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83
       entry_SYSCALL_64_after_hwframe+0x77/0x7f

-> #1 ((pm_chain_head).rwsem){++++}-{4:4}:
       down_read+0x9a/0x330 kernel/locking/rwsem.c:1524
       blocking_notifier_call_chain_robust kernel/notifier.c:344 [inline]
       blocking_notifier_call_chain_robust+0xa9/0x170 kernel/notifier.c:333
       pm_notifier_call_chain_robust+0x27/0x60 kernel/power/main.c:102
       snapshot_open+0x189/0x2b0 kernel/power/user.c:77
       misc_open+0x35a/0x420 drivers/char/misc.c:179
       chrdev_open+0x237/0x6a0 fs/char_dev.c:414
       do_dentry_open+0x735/0x1c40 fs/open.c:956
       vfs_open+0x82/0x3f0 fs/open.c:1086
       do_open fs/namei.c:3830 [inline]
       path_openat+0x1e88/0x2d80 fs/namei.c:3989
       do_filp_open+0x20c/0x470 fs/namei.c:4016
       do_sys_openat2+0x17a/0x1e0 fs/open.c:1428
       do_sys_open fs/open.c:1443 [inline]
       __do_sys_openat fs/open.c:1459 [inline]
       __se_sys_openat fs/open.c:1454 [inline]
       __x64_sys_openat+0x175/0x210 fs/open.c:1454
       do_syscall_x64 arch/x86/entry/common.c:52 [inline]
       do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83
       entry_SYSCALL_64_after_hwframe+0x77/0x7f

-> #0 (system_transition_mutex){+.+.}-{4:4}:
       check_prev_add kernel/locking/lockdep.c:3163 [inline]
       check_prevs_add kernel/locking/lockdep.c:3282 [inline]
       validate_chain kernel/locking/lockdep.c:3906 [inline]
       __lock_acquire+0x249e/0x3c40 kernel/locking/lockdep.c:5228
       lock_acquire.part.0+0x11b/0x380 kernel/locking/lockdep.c:5851
       __mutex_lock_common kernel/locking/mutex.c:585 [inline]
       __mutex_lock+0x19b/0xb10 kernel/locking/mutex.c:730
       lock_system_sleep+0x87/0xa0 kernel/power/main.c:56
       hibernate_compressor_param_set+0x1c/0x210 kernel/power/hibernate.c:1452
       param_attr_store+0x18f/0x300 kernel/params.c:588
       module_attr_store+0x55/0x80 kernel/params.c:924
       sysfs_kf_write+0x117/0x170 fs/sysfs/file.c:139
       kernfs_fop_write_iter+0x33d/0x500 fs/kernfs/file.c:334
       new_sync_write fs/read_write.c:586 [inline]
       vfs_write+0x5ae/0x1150 fs/read_write.c:679
       ksys_write+0x12b/0x250 fs/read_write.c:731
       do_syscall_x64 arch/x86/entry/common.c:52 [inline]
       do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83
       entry_SYSCALL_64_after_hwframe+0x77/0x7f
```

```
other info that might help us debug this:

Chain exists of:
  system_transition_mutex --> rtnl_mutex --> param_lock

 Possible unsafe locking scenario:

       CPU0                    CPU1
       ----                    ----
  lock(param_lock);
                               lock(rtnl_mutex);
                               lock(param_lock);
  lock(system_transition_mutex);

 *** DEADLOCK ***
```

Reported-by: syzbot+ace60642828c074eb913@syzkaller.appspotmail.com
Closes: https://syzkaller.appspot.com/bug?extid=ace60642828c074eb913
Signed-off-by: Lizhi Xu <lizhi.xu@windriver.com>
Link: https://patch.msgid.link/20250224013139.3994500-1-lizhi.xu@windriver.com
[rjw: New subject matching the code changes, changelog edits]
Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

-rw-r--r-- kernel/power/hibernate.c 6

1 files changed, 3 insertions, 3 deletions

```
diff --git a/kernel/power/hibernate.c b/kernel/power/hibernate.c
index b483fcea811b1a..d8bad1eeedd3e5 100644
--- a/kernel/power/hibernate.c
+++ b/kernel/power/hibernate.c
@@ -1443,10 +1443,10 @@ static const char * const comp_alg_enabled[] = {
 static int hibernate_compressor_param_set(const char *compressor,
                const struct kernel_param *kp)
 {
-       unsigned int sleep_flags;
        int index, ret;

-       sleep_flags = lock_system_sleep();
+       if (!mutex_trylock(&system_transition_mutex))
+               return -EBUSY;

        index = sysfs_match_string(comp_alg_enabled, compressor);
        if (index >= 0) {
@@ -1458,7 +1458,7 @@ static int hibernate_compressor_param_set(const char *compressor,
                ret = index;
        }

-       unlock_system_sleep(sleep_flags);
+       mutex_unlock(&system_transition_mutex);

        if (ret)
                pr_debug("Cannot set specified compressor %s\n",
```