author	Ryusuke Konishi <konishi.ryusuke@gmail.com>	2023-03-07 17:55:48 +0900
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2023-04-05 11:23:41 +0200
commit	d18db946cc6a394291539e030df32324285648f7 (patch)	
tree	fbfec9f46cd0525e4b450266195cf8958eb8d7b5	
parent	560437bba14d0546de72e3e9d0a2f78d88e8cd91 (diff)	
download	linux-d18db946cc6a394291539e030df32324285648f7.tar.gz	

diff options

context: 3

space: include

mode: unified

nilfs2: fix kernel-infoleak in nilfs_ioctl_wrap_copy()

```
commit 003587000276f81d0114b5ce773d80c119d8cb30 upstream.

The ioctl helper function nilfs_ioctl_wrap_copy(), which exchanges a
metadata array to/from user space, may copy uninitialized buffer regions
to user space memory for read-only ioctl commands NILFS_IOCTL_GET_SUINFO
and NILFS_IOCTL_GET_CPINFO.

This can occur when the element size of the user space metadata given by
the v_size member of the argument nilfs_argv structure is larger than the
size of the metadata element (nilfs_suinfo structure or nilfs_cpinfo
structure) on the file system side.

KMSAN-enabled kernels detect this issue as follows:

 BUG: KMSAN: kernel-infoleak in instrument_copy_to_user
 include/linux/instrumented.h:121 [inline]
 BUG: KMSAN: kernel-infoleak in _copy_to_user+0xc0/0x100 lib/usercopy.c:33
  instrument_copy_to_user include/linux/instrumented.h:121 [inline]
  _copy_to_user+0xc0/0x100 lib/usercopy.c:33
  copy_to_user include/linux/uaccess.h:169 [inline]
  nilfs_ioctl_wrap_copy+0x6fa/0xc10 fs/nilfs2/ioctl.c:99
  nilfs_ioctl_get_info fs/nilfs2/ioctl.c:1173 [inline]
  nilfs_ioctl+0x2402/0x4450 fs/nilfs2/ioctl.c:1290
  nilfs_compat_ioctl+0x1b8/0x200 fs/nilfs2/ioctl.c:1343
  __do_compat_sys_ioctl fs/ioctl.c:968 [inline]
  __se_compat_sys_ioctl+0x7dd/0x1000 fs/ioctl.c:910
  __ia32_compat_sys_ioctl+0x93/0xd0 fs/ioctl.c:910
  do_syscall_32_irqs_on arch/x86/entry/common.c:112 [inline]
  __do_fast_syscall_32+0xa2/0x100 arch/x86/entry/common.c:178
  do_fast_syscall_32+0x37/0x80 arch/x86/entry/common.c:203
  do_SYSENTER_32+0x1f/0x30 arch/x86/entry/common.c:246
  entry_SYSENTER_compat_after_hwframe+0x70/0x82

 Uninit was created at:
  __alloc_pages+0x9f6/0xe90 mm/page_alloc.c:5572
  alloc_pages+0xab0/0xd80 mm/mempolicy.c:2287
  __get_free_pages+0x34/0xc0 mm/page_alloc.c:5599
  nilfs_ioctl_wrap_copy+0x223/0xc10 fs/nilfs2/ioctl.c:74
  nilfs_ioctl_get_info fs/nilfs2/ioctl.c:1173 [inline]
  nilfs_ioctl+0x2402/0x4450 fs/nilfs2/ioctl.c:1290
  nilfs_compat_ioctl+0x1b8/0x200 fs/nilfs2/ioctl.c:1343
```

```
  __do_compat_sys_ioctl fs/ioctl.c:968 [inline]
  __se_compat_sys_ioctl+0x7dd/0x1000 fs/ioctl.c:910
  __ia32_compat_sys_ioctl+0x93/0xd0 fs/ioctl.c:910
  do_syscall_32_irqs_on arch/x86/entry/common.c:112 [inline]
  __do_fast_syscall_32+0xa2/0x100 arch/x86/entry/common.c:178
  do_fast_syscall_32+0x37/0x80 arch/x86/entry/common.c:203
  do_SYSENTER_32+0x1f/0x30 arch/x86/entry/common.c:246
  entry_SYSENTER_compat_after_hwframe+0x70/0x82

 Bytes 16-127 of 3968 are uninitialized
 ...
```

This eliminates the leak issue by initializing the page allocated as
buffer using get_zeroed_page().

```
Link: https://lkml.kernel.org/r/20230307085548.6290-1-konishi.ryusuke@gmail.com
Signed-off-by: Ryusuke Konishi <konishi.ryusuke@gmail.com>
Reported-by: syzbot+132fdd2f1e1805fdc591@syzkaller.appspotmail.com
  Link: https://lkml.kernel.org/r/000000000000a5bd2d05f63f04ae@google.com
Tested-by: Ryusuke Konishi <konishi.ryusuke@gmail.com>
Cc: <stable@vger.kernel.org>
Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

Diffstat

-rw-r--r-- fs/nilfs2/ioctl.c 2

1 files changed, 1 insertions, 1 deletions

```
diff --git a/fs/nilfs2/ioctl.c b/fs/nilfs2/ioctl.c
index 3a1dea5d144847..01235fac5971f1 100644
--- a/fs/nilfs2/ioctl.c
+++ b/fs/nilfs2/ioctl.c
@@ -70,7 +70,7 @@ static int nilfs_ioctl_wrap_copy(struct the_nilfs *nilfs,
        if (argv->v_index > ~(__u64)0 - argv->v_nmembs)
                return -EINVAL;

-       buf = (void *)__get_free_pages(GFP_NOFS, 0);
+       buf = (void *)get_zeroed_page(GFP_NOFS);
        if (unlikely(!buf))
                return -ENOMEM;
        maxmembs = PAGE_SIZE / argv->v_size;
```