From: butt3rflyh4ck <butterflyhuangxx@gmail.com>
To: perex@perex.cz, tiwai@suse.com, cuibixuan@linux.alibaba.com
Cc: alsa-devel@alsa-project.org, LKML <linux-kernel@vger.kernel.org>
Subject: A new null-ptr-deref Write bug in snd_pcm_format_set_silence
Date: Sun, 4 Sep 2022 17:48:37 +0800 [thread overview]
Message-ID: <CAFcO6XN7JDM4xSXGhtusQfS2mSBcx50VJKwQpCq=WeLt57aaZA@mail.gmail.com> (raw)

Hi, there is a new null-ptr-deref Write bug in
snd_pcm_format_set_slience in sound/core/pcm_misc.c in the latest
upstream kernel and can reproduce it.
We call SNDCTL_DSP_SYNC and SNDCTL_DSP_SPEED in multiple threads to
trigger the vulnerability.

See the Call Trace:
```
==================================================================
Call Trace:
 <TASK>
 __dump_stack lib/dump_stack.c:88 [inline]
 dump_stack_lvl+0xcd/0x134 lib/dump_stack.c:106
 kasan_report+0xb1/0x1e0 mm/kasan/report.c:495
 check_region_inline mm/kasan/generic.c:183 [inline]
 kasan_check_range+0x13d/0x180 mm/kasan/generic.c:189
 memset+0x20/0x40 mm/kasan/shadow.c:44
 snd_pcm_format_set_silence sound/core/pcm_misc.c:441 [inline]
 snd_pcm_format_set_silence+0x215/0x350 sound/core/pcm_misc.c:424
 snd_pcm_oss_sync+0x60e/0x800 sound/core/oss/pcm_oss.c:1690
 snd_pcm_oss_ioctl+0x2087/0x3420 sound/core/oss/pcm_oss.c:2634
 vfs_ioctl fs/ioctl.c:51 [inline]
 __do_sys_ioctl fs/ioctl.c:870 [inline]
 __se_sys_ioctl fs/ioctl.c:856 [inline]
 __x64_sys_ioctl+0x193/0x200 fs/ioctl.c:856
 do_syscall_x64 arch/x86/entry/common.c:50 [inline]
 do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80
 entry_SYSCALL_64_after_hwframe+0x63/0xcd
==================================================================
```
We can see the function snd_pcm_format_set_silence code below:
```
int snd_pcm_format_set_silence(snd_pcm_format_t format, void *data,
unsigned int samples)
{
        int width;
        unsigned char *dst;
        const unsigned char *pat;

        if (!valid_format(format))
                return -EINVAL;
        if (samples == 0)
                return 0;
        width = pcm_formats[(INT)format].phys; /* physical width */
        pat = pcm_formats[(INT)format].silence;
        if (!width || !pat)
                return -EINVAL;
        /* signed or 1 byte data */
        if (pcm_formats[(INT)format].signd == 1 || width <= 8) {
                unsigned int bytes = samples * width / 8;
                memset(data, *pat, bytes);    ///// [1] ---------> data is NULL
                return 0;
        }
       ......
}
```

[1], the data pointer is NULL, we can know snd_pcm_format_set_silence called in line 1690 in sound/core/oss/pcm_oss.c from call stack trace. let we see code below:

```
static int snd_pcm_oss_sync(struct snd_pcm_oss_file *pcm_oss_file)
{
        int err = 0;
        unsigned int saved_f_flags;
        struct snd_pcm_substream *substream;
        struct snd_pcm_runtime *runtime;
        snd_pcm_format_t format;
        unsigned long width;
        size_t size;

        substream = pcm_oss_file->streams[SNDRV_PCM_STREAM_PLAYBACK];
        if (substream != NULL) {
                runtime = substream->runtime;
                if (atomic_read(&substream->mmap_count))
                        goto __direct;
                err = snd_pcm_oss_make_ready(substream);
                if (err < 0)
                        return err;
                atomic_inc(&runtime->oss.rw_ref);
                if (mutex_lock_interruptible(&runtime->oss.params_lock)) {
                        atomic_dec(&runtime->oss.rw_ref);
                        return -ERESTARTSYS;
                }
                format = snd_pcm_oss_format_from(runtime->oss.format);
                width = snd_pcm_format_physical_width(format);
                if (runtime->oss.buffer_used > 0) {
#ifdef OSS_DEBUG
                        pcm_dbg(substream->pcm, "sync: buffer_used\n");
#endif
                        size = (8 * (runtime->oss.period_bytes -
runtime->oss.buffer_used) + 7) / width;
                        snd_pcm_format_set_silence(format,
                                                   runtime->oss.buffer
+ runtime->oss.buffer_used,   ///// [2]
                                                   size);
                        err = snd_pcm_oss_sync1(substream,
runtime->oss.period_bytes);
                        if (err < 0)
                                goto unlock;
                } else if (runtime->oss.period_ptr > 0) {

```

[2] runtime->oss.buffer + runtime->oss.buffer_used is the data pointer, but runtime->oss.buffer is NULL here but it doesn't make sense.
runtime->oss.buffter is allocated by kvzalloc, if runtime->oss_buffer is NULL, it would return an ENOMEM error.
Maybe I think there is a race condition, the runtime->oss.buffer is freed and set to NULL but we can use runtime->oss.buffter via ioctl.

###reproduce it

```
[  167.258988][T25615]
==================================================================
[  167.265917][T25615] BUG: KASAN: null-ptr-deref in
snd_pcm_format_set_silence+0x215/0x350
[  167.266704][T25615] Write of size 16383 at addr 0000000000000001 by
task snd_pcm_format_/25615
[  167.267506][T25615]
[  167.267732][T25615] CPU: 0 PID: 25615 Comm: snd_pcm_format_ Not
tainted 6.0.0-rc3-00299-gd895ec7938c4 #11
[  167.268617][T25615] Hardware name: QEMU Standard PC (i440FX + PIIX,
```

```
1996), BIOS 1.14.0-2 04/01/2014
[  167.269410][T25615] Call Trace:
[  167.269697][T25615]  <TASK>
[  167.269977][T25615]  dump_stack_lvl+0xcd/0x134
[  167.270428][T25615]  ? snd_pcm_format_set_silence+0x215/0x350
[  167.270985][T25615]  kasan_report+0xb1/0x1e0
[  167.271405][T25615]  ? snd_pcm_format_set_silence+0x215/0x350
[  167.271964][T25615]  kasan_check_range+0x13d/0x180
[  167.272440][T25615]  memset+0x20/0x40
[  167.272809][T25615]  snd_pcm_format_set_silence+0x215/0x350
[  167.273366][T25615]  snd_pcm_oss_sync+0x60e/0x800
[  167.273831][T25615]  snd_pcm_oss_ioctl+0x2087/0x3420
[  167.274320][T25615]  ? snd_pcm_oss_release+0x300/0x300
[  167.274817][T25615]  ? __fget_files+0x26a/0x440
[  167.275262][T25615]  ? bpf_lsm_file_ioctl+0x5/0x10
[  167.275731][T25615]  ? snd_pcm_oss_release+0x300/0x300
[  167.276222][T25615]  __x64_sys_ioctl+0x193/0x200
[  167.276677][T25615]  do_syscall_64+0x35/0xb0
[  167.277108][T25615]  entry_SYSCALL_64_after_hwframe+0x63/0xcd
[  167.277679][T25615] RIP: 0033:0x44af9d
[  167.278058][T25615] Code: 66 2e 0f 1f 84 00 00 00 00 00 0f 1f 00 f3
0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b
4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b8 ff ff 8
[  167.279792][T25615] RSP: 002b:00007f0cb636a178 EFLAGS: 00000246
ORIG_RAX: 0000000000000010
[  167.280546][T25615] RAX: ffffffffffffffda RBX: 0000000000000000
RCX: 000000000044af9d
[  167.281285][T25615] RDX: 0000000000000000 RSI: 0000000000005001
RDI: 0000000000000003
[  167.282001][T25615] RBP: 00007f0cb636a1a0 R08: 0000000000000000
R09: 0000000000000000
[  167.282715][T25615] R10: 0000000000000000 R11: 0000000000000246
R12: 00007ffcfea8e08e
[  167.283432][T25615] R13: 00007ffcfea8e08f R14: 0000000000000000
R15: 00007f0cb636a640
[  167.284164][T25615]  </TASK>
[  167.284453][T25615]
================================================================
```

If needed I would provide a reproduce.

Regards,
   butt3rflyh4ck.

--
Active Defense Lab of Venustech

---

next reply other threads:[~2022-09-04 9:49 UTC|newest]

Thread overview: 7+ messages / expand[flat|nested] mbox.gz Atom feed top
2022-09-04 9:48 butt3rflyh4ck [this message]
2022-09-04 10:27 ` A new null-ptr-deref Write bug in snd_pcm_format_set_silence Takashi Iwai
2022-09-04 10:27   ` Takashi Iwai
2022-09-04 17:06   ` butt3rflyh4ck
2022-09-04 17:06   ` butt3rflyh4ck
2022-09-05 6:06   ` Takashi Iwai
2022-09-05 6:06   ` Takashi Iwai

Reply instructions:

You may reply publicly to this message via plain-text email
using any one of the following methods:

* Save the following mbox file, import it into your mail client,
  and reply-to-all from there: mbox

  Avoid top-posting and favor interleaved quoting:
  https://en.wikipedia.org/wiki/Posting_style#Interleaved_style

* Reply using the **--to**, **--cc**, and **--in-reply-to**
  switches of git-send-email(1):

  ```
  git send-email \
    --in-reply-to='CAFcO6XN7JDM4xSXGhtusQfS2mSBcx50VJKwQpCq=WeLt57aaZA@mail.gmail.com' \
    --to=butterflyhuangxx@gmail.com \
    --cc=alsa-devel@alsa-project.org \
    --cc=cuibixuan@linux.alibaba.com \
    --cc=linux-kernel@vger.kernel.org \
    --cc=perex@perex.cz \
    --cc=tiwai@suse.com \
    /path/to/YOUR_REPLY
  ```

  https://kernel.org/pub/software/scm/git/docs/git-send-email.html

* If your mail client supports setting the **In-Reply-To** header
  via mailto: links, try the mailto: link

Be sure your reply has a **Subject:** header at the top and a blank line before the message body.
