MWR InfoSecurity Security Advisory

# Linux Kernel caiaq USB Drivers Buffer Overflow Vulnerability

7th March 2011

MWR InfoSecurity

MWR INFOSECURITY

# caiaq USB Drivers –Buffer Overflow Vulnerabitlity

| | |
|---|---|
| **Package Name:** | Linux Kernel - caiaq USB drivers |
| **Date Reported** | 14th February 2011 |
| **Affected Versions:** | Linux kernel before 2.6.38-rc4-next-20110215 |

| | |
|---|---|
| CVE Reference | CVE-2011-0712 |
| Author | Rafael Dominguez Vega |
| Severity | High Risk |
| Vulnerability Class | Buffer overflow |
| Vendor | Linux Kernel |
| Vendor Response | The vendor implemented a fix: http://git.kernel.org/?p=linux/kernel/git/tiwai/sound-2.6.git;a=commitdiff;h=eaae55dac6b64c0616046436b294e69fc5311581 |

## Overview

MWR InfoSecurity identified a buffer overflow vulnerability in the caiaq USB drivers. These dirvers are in the kernel tree and installed by default in most Linux distributions.

This device's drivers are vulnerable to a buffer overflow condition which could be exploited by an attacker with physical access to the system. This vulnerability could be exploited in order to execute arbitrary code on the target system.

## Impact

The vulnerabilities would enable an attacker to execute arbitrary code on the target system at the kernel level. This could allow full control to be gained over the system.

## Cause

A buffer overflow vulnerability was identified in the code handling the USB product name in the following drivers:

- . /sound/usb/caiaq/audio.c
- ./sound/usb/caiaq/midi.c

## Interim Workaround

Removing the affected drivers will prevent users from exploiting the vulnerability.

## Solution

The vendor has implemented a fix.
http://git.kernel.org/?p=linux/kernel/git/tiwai/sound-2.6.git;a=commitdiff;h=eaae55dac6b64c0616046436b294e69fc5311581

## Dependencies

In order to successfully exploit the vulnerability described in this advisory, an attacker would need to have physical access to the affected system in order to be able to plug in a malicious USB device.

## Detailed Vulnerability Description

The issue is a buffer overflow vulnerability affecting the following drivers, in the code responsible for handling the USB product name.

- ./sound/usb/caiaq/audio.c
- ./sound/usb/caiaq/midi.c

The affected code is included here for the two affected drivers. The vulnerability is in the strcpy shown below, as the product name that the USB device sends ("dev->product_name") can be larger than the buffer of "dev->pcm->name" and "rmidi->name", where the data is being copied to (80 bytes).

### Affected driver audio.c

```
int snd usb caiaq audio init(struct snd usb caiaqdev *dev)
{
...

        ret = snd pcm new(dev->chip.card, dev->product name, 0,
                          dev->n audio out, dev->n audio in, &dev->pcm);
...

dev->pcm->private data = dev;
strcpy(dev->pcm->name, dev->product_name);
...
```

Source code from /linux-2.6.38/sound/usb/caiaq/audio.c

```
struct snd pcm {
        struct snd_card *card;
        struct list_head list;
        int device; /* device number */
        unsigned int info flags;
        unsigned short dev class;
        unsigned short dev subclass;
        char id[64];
        char name[80];
        struct snd_pcm_str streams[2];
        struct mutex open mutex;
        wait queue head t open wait;
        void *private data;
        void (*private_free) (struct snd_pcm *pcm);
        struct device *dev; /* actual hw device this belongs to */
#if defined(CONFIG_SND_PCM_OSS) || defined(CONFIG_SND_PCM_OSS_MODULE)
        struct snd pcm oss oss;
#endif
};
```

Source code from /linux-2.6.38/include/sound/pcm.h

## Affected driver midi.c

```
nt snd_usb_caiaq_midi_init(struct snd_usb_caiaqdev *device)
{
        int ret;
        struct snd rawmidi *rmidi;

        ret = snd_rawmidi_new(device->chip.card, device->product_name, 0,
                                        device->spec.num_midi_out,
                                        device->spec.num_midi_in,
                                        &rmidi);

        if (ret < 0)
                return ret;

        strcpy(rmidi->name, device->product_name);
...
```

Source code from /linux-2.6.38/sound/usb/caiaq/midi.c

```
struct snd_rawmidi {
        struct snd_card *card;
        struct list_head list;
        unsigned int device;            /* device number */
        unsigned int info_flags;        /* SNDRV_RAWMIDI_INFO_XXXX */
        char id[64];
        char name[80];
...
```

Source code from /linux-2.6.38/include/sound/rawmidi.h

During the investigation of this vulnerability a Proof-of-Concept USB device was created. The malicious USB device was specially developed to trigger this issue.

```
[  125.278401] Stack:
[  125.278403]  c4807d58 c89e755e 00000000 00000000 c3da4c80 000000c8 050a0004
00000000
[  125.278415] <0> c3da4204 c4807dd4 c4807e04 c4807d8c c89e8f3d 00000003 c3da4094
c4807dac
[  125.278422] <0> ff0a0000 ffffffff c3da4204 c4807dd4 c4010000 c3da4204 c4807dd4
c4807e04
[  125.278432] Call Trace:
[  125.278448]  [<c89e755e>] ? snd_usb_caiaq_audio_init+0x1ce/0x3a0 [snd_usb_caiaq]
[  125.278458]  [<c89e8f3d>] ? setup card+0x1c5/0x268 [snd usb caiaq]
[  125.278464]  [<c89e92a9>] ? init_card+0x2c9/0x2e5 [snd_usb_caiaq]
[  125.278495]  [<c0165e10>] ? autoremove_wake_function+0x0/0x50
[  125.278510]  [<c012ce18>] ? default_spin_lock_flags+0x8/0x10
[  125.278515]  [<c89e937d>] ? snd probe+0xb8/0xea [snd usb caiaq]
[  125.278532]  [<c0475a3c>] ? usb probe interface+0xec/0x1b0
[  125.278548]  [<c0271ed7>] ? sysfs create link+0x17/0x20
[  125.278561]  [<c0400550>] ? really_probe+0x50/0x150
[  125.278568]  [<c0407bc7>] ? pm_runtime_barrier+0x57/0xb0
[  125.278572]  [<c040068c>] ? driver_probe_device+0x3c/0x60
[  125.278575]  [<c0400731>] ?   driver attach+0x81/0x90
[  125.278579]  [<c03ffb53>] ? bus for each dev+0x53/0x80
[  125.278583]  [<c040041e>] ? driver_attach+0x1e/0x20
[  125.278586]  [<c04006b0>] ? __driver_attach+0x0/0x90
[  125.278589]  [<c03ffde5>] ? bus_add_driver+0xd5/0x280
[  125.278593]  [<c0400a2a>] ? driver register+0x6a/0x130
[  125.278598]  [<c0475751>] ? usb register driver+0x81/0x130
[  125.278613]  [<c01b05e7>] ? tracepoint module notify+0x27/0x30
[  125.278629]  [<c05cc6b3>] ? notifier_call_chain+0x43/0x60
[  125.278635]  [<c8a2a017>] ? snd_module_init+0x17/0x19 [snd_usb_caiaq]
[  125.278639]  [<c0101132>] ? do_one_initcall+0x32/0x1a0
[  125.278647]  [<c8a2a000>] ? snd_module_init+0x0/0x19 [snd_usb_caiaq]
```

```
[  125.278659]  [<c0180c2b>] ? sys init module+0x9b/0x1e0
[  125.278670]  [<c0218fa2>] ? sys write+0x42/0x70
[  125.278675]  [<c05c90a4>] ? syscall call+0x7/0xb
[  125.278677] Code: f8 ff ff eb ca 90 90 90 90 90 90 90 90 90 90 90 90 90 55 89
e5 0f 1f 44 00 00 6b d2 1c 8b 84 10 b8 00 00 00 85 c0 74 0c 66 90 <89> 48 5c 8b 40
6c 85 c0 75 f6 5d c3 8d b6 00 00 00 00 8d bf 00
[  125.278719] EIP: [<c8a10dc8>] snd pcm set ops+0x18/0x30 [snd pcm] SS:ESP
0068:c4807d2c
[  125.278728] CR2: 0000000080c3811f
[  125.278737] ---[ end trace 44f4e7357c0cdf45 ]---
```


## Acknowledgement

This research has been conducted in partnership with VulnDev Ltd.

MWR InfoSecurity
St. Clement House
1-3 Alencon Link
Basingstoke, RG21 7SB
Tel: +44 (0)1256 300920
Fax: +44 (0)1256 844083
mwrinfosecurity.com