MWR InfoSecurity Security Advisory

Linux USB Device Driver - Buffer Overflow

29th October 2009

MWR INFOSECURITY

MWR InfoSecurity

Contents

Linux USB Device Driver - Buffer Overflow

Package Name:	Auerswald Linux USB Device Driver
Date Discovered:	October 2008
Affected Versions:	Confirmed in Linux Kernel 2.6.26 (Driver included by default)

CVE Reference	CVE-2009-4067
Author	R. Dominguez Vega
Severity	High Risk
Local/Remote	Local with physical access required
Vulnerability Class	Buffer Overflow
Vendor	Linux Kernel
Vendor Informed Timeline	This issue was discussed with the Linux Kernel Security Team via email (security@kernel.org): - • Vendor first contacted - 14/09/2009 • Vulnerability details provided - 19/09/2009 • Vendor responded - 19/09/2009
Vendor Response	The vendor did not supply a fix but instead recommended that users should upgrade to the latest stable Linux kernel version.

Overview:

The Auerswald Linux USB device driver is used to allow compability of an Auerswald PBX/System telephone with Linux Operating Systems via the USB port.

This device driver is vulnerable to a buffer overflow 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 vulnerability 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 has been identified in the code handling the USB “string descriptors”. This vulnerability is associated with the memory for the element “dev_desc” of the USB device context structure and could result in the overwriting of other elements of the structure.

Solution:

Remove the Auerswald USB device driver from your kernel or upgrade to the latest stable Linux kernel version.

1 Detailed Vulnerability Description

1.1 Technical Background

"In information technology, Universal Serial Bus (USB) is a serial bus standard to connect devices to a host computer. USB was designed to allow many peripherals to be connected using a single standardized interface socket and to improve plug and play capabilities by allowing hot swapping; that is, by allowing devices to be connected and disconnected without rebooting the computer or turning off the device." [1]

1.2 Vulnerability Details

A vulnerability was identified in the code responsible for handling the USB "string descriptors", such that an attacker could bypass the validation length check for the element "dev_desc" of the USB device context structure, used for storing the USB textual description; therefore allowing the overwrite of other elements of the structure.

The "auerswald_probe" function is called when a new device is attached to the bus and is responsible for the handling of the USB string descriptors in which specific parameters (such as the device name, serial number and manufacturer name) from the device are passed to the host computer.

The vulnerability results from the following code: [2]

```
/* Try to get a suitable textual description of the device */
/* Device name:*/
ret = usb_string( cp->usbdev, AUSI_DEVICE, cp->dev_desc, AUSI_DLEN-1);
 if (ret >= 0) {
  u += ret;
  /* Append Serial Number */
  memcpy(&cp->dev_desc[u], ",Ser# ", 6);
  u += 6;
  ret = usb_string( cp->usbdev, AUSI_SERIALNR, &cp->dev_desc[u], AUSI_DLEN-u-1);
  if (ret >= 0) {
      u += ret;
      /* Append subscriber number */
      memcpy(&cp->dev_desc[u], ", ", 2);
      u += 2;
      ret = usb_string( cp->usbdev, AUSI_MSN, &cp->dev_desc[u], AUSI_DLEN-u-1);
      if (ret >= 0) {
             u += ret;
      }
  }
 }
```

It is possible to overflow the buffer assigned to the element "dev_desc[AUSI_DLEN]"; belonging to the USB device context structure. The element "dev_desc" is of type char and "AUSI_DLEN" delimits the maximum length of the device descriptor and is set to 100.

An attacker could craft a large answer to the request for "AUSI_DEVICE" (name of device) in order to produce a condition where "u" was much greater than "AUSI_DLEN - 1" such that "AUSI_DLEN - u - 1" was a large, negative number. As the argument to usb_string () is unsigned this would result in a very large value. Consequently, this would allow an attacker to write a certain amount of arbitrary data to the static sized buffer "dev_desc".

Overflowing this buffer would allow other elements of the structure to be overwritten, such as “maxControlLength”, “inturbp” and “intbufp”. This could allow an attacker to gain control of the EIP register and so, potentially, to execute arbitrary code under the context of the system kernel.

1.3 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. Alternatively, an attacker would require sufficient access to a system to emulate a malicious USB device.

2 Proof of Concept

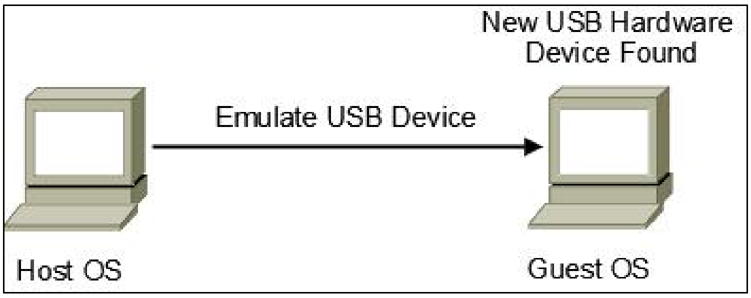
Proof of concept details are provided in this document in order to reproduce the kernel bug identified and facilitate its resolution.

2.1 Environment

Qemu will be used for the environment set up [3]. Qemu is a generic open source machine emulator and virtualizer, which allows the emulation of USB devices from the Host system to the Guest System.

In this set up, the Host system will be the attacker and will be emulating the Auerswald USB device that will be inserted into the target (the Guest system) which is running a Linux based OS with the Auerswald USB driver compiled within it.

The diagram below illustrates the environment used to reproduce the identified kernel bug:-



2.2 Malicious Qemu USB Device

The malicious USB device used to trigger the vulnerability in the Auerswald USB driver should be programmed before compiling Qemu in the environment. For this proof of concept illustration, one of the USB device emulators provided by Qemu will be used to contain the malicious data that will trigger this bug. In this example, the “usb-wacom.c” [4] in Qemu will be modified.

First of all, the “idVendor” and “idProduct” will be modified with the following code, in order for the Auerswald driver to be called when the USB device is emulated.

```
    0x00,       /*  u8  bDeviceProtocol; [ low/full speeds only ] */
    0x08,       /*  u8  bMaxPacketSize0; 8 Bytes */

    0xbf, 0x09, /*  u16 idVendor; */
    0xc0, 0x00, /*  u16 idProduct; */
    0x10, 0x42, /*  u16 bcdDevice */
```

Then the data that will cause the buffer overflow and the kernel crash to occur will be added.

```
            case 1:
                /* serial number */
                ret = set_usb_string(data, “”);
                break;
            case 2:
                ret = set_usb_string(data,
“AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA”);
```

2.3 Device Emulation

Once the malicious USB device has been programmed, Qemu successfully compiled and the Guest system has been started up; the Auerswald device can be emulated.

The Qemu Guest system control console can be accessed by pressing "Ctrl+Alt+2", and the device is emulated by entering the following in the console: -

```
usb_add wacom-tablet
```

The USB device will be emulated and the bug triggered causing a segmentation fault and kernel crash.

3 Recommendations

Remove the Auerswald USB device driver from your kernel or upgrade to the latest stable and secure Linux kernel version.

4 References

[1] Universal Serial Bus
http://en.wikipedia.org/wiki/USB

[2] Qemu
http://www.qemu.org

[3] Auerswald PBX/System Telephone USB driver by Wolfgang (auerswald.c)

[4] Wacom PenPartner USB tablet emulation by Andrzej Zaborowski (usb-wacom.c)

5 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