Don’t trust your USB

How to find bugs in USB device drivers
Motivation

- compromise systems via USB
- find bugs, fix or exploit them :-(
BadUSB (2014)  
Facedancer (2012)  
USB Fuzzing for the Masses (2011)
Impact

**CVE-2013-1285**

The USB kernel-mode drivers in Microsoft Windows XP SP2 and SP3, Windows Server 2003 SP2, Windows Vista SP2, Windows Server 2008 SP2, R2, and R2 SP1, Windows 7 Gold and SP1, Windows 8, and Windows Server 2012 do not properly handle objects in memory, which allows physically proximate attackers to execute arbitrary code by connecting a crafted USB device, aka "Windows USB Descriptor Vulnerability," a different vulnerability than CVE-2013-1286 and CVE-2013-1287.

Publish Date : 2013-03-12 Last Update Date : 2013-11-02

**CVE-2013-1680**

Heap-based buffer overflow in the wdm_in_callback function in drivers/usb/class/cdc-wdm.c in the Linux kernel before 3.8.4 allows physically proximate attackers to cause a denial of service (system crash) or possibly execute arbitrary code via a crafted cdc-wdm USB device.

Publish Date : 2013-03-22 Last Update Date : 2014-04-19
Plug and Root?

Darrin Barrall and David Dewey 2005 at Black Hat
## USB in a nutshell

<table>
<thead>
<tr>
<th>Driver initialization</th>
<th>Driver specific function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>usb-core-driver</strong></td>
<td><strong>specific driver</strong></td>
</tr>
<tr>
<td></td>
<td><strong>subsystems</strong></td>
</tr>
<tr>
<td></td>
<td><strong>HID</strong></td>
</tr>
<tr>
<td></td>
<td><strong>SCSI</strong></td>
</tr>
<tr>
<td></td>
<td><strong>r8192u_usb</strong></td>
</tr>
<tr>
<td></td>
<td><strong>udlfb</strong></td>
</tr>
<tr>
<td></td>
<td><strong>network</strong></td>
</tr>
<tr>
<td></td>
<td><strong>framebuffer</strong></td>
</tr>
</tbody>
</table>

- **Endpoint data transfer**
- **Enumeration**
Enumeration

Vendor: 0x0bb4 (HTC)
Product: 0x0a09 (PocketPC Sync)

Sergej Schumilo - schumilo@fh-muenster.de
vUSBf Framework

• usage of QEMU and KVM for virtualization

• usage of USB Redirection interface for USB data injection
  • send usb traffic through TCP, UDP or Unix sockets
  • wrap usb data into USB Redirection protocol headers
  • it supports USB 1.0, 1.1, 2.0, 3.0
USB Redirection

- provides remote USB-device
- part of the SPICE suite
- monitoring process start
- load snapshot
- recognition for image corruption
Sergej Schumilo - schumilo@fh-muenster.de

12

USB-Emulator

USB-Traffic

traffic fuzzing

Fuzzer-Modul

Read or generate Device-Descriptor

descriptor fuzzing

QEMU

usbredir-interface

VM

QEMU monitoring console

monitoring per serial port

Monitoring-Modul

- monitoring process start
- load snapshot
- recognition for image corruption

QEMU-Controller

write to logfile

QEMU monitoring console

traffic fuzzing

Document page 12
vUSBf
reload mode

1. Start QEMU
2. Load snapshot
3. Send payload
4. Write impact to logfile
vUSBf
Non-reload mode

start QEMU → load snapshot → send payload

issue detected? (yes/no)

write impact to logfile
Reproducibility

• test cases are related to unique IDs

• file export for a sequence of test cases

• offers high reproducibility in combination with snapshots
vUSBf Performance

- vUSBf (1 process)
  - 0,5
  - 0,5

- vUSBf Multiprocessing
  - 50
  - 120

- vUSBf Clustering
  - 150
  - 320

Tests per seconds

reload mode
non reload mode

Sergej Schumilo - schumilo@fh-muenster.de
vUSBf Performance

runtime for 1 million tests:
- vUSBf (1 process):
  - reload mode: 23d 4h
  - non reload mode: 23d 4h
- vUSBf Multiprocessing:
  - runtime: 5h 30m
- vUSBf Clustering:
  - runtime: 2h 18m
  - runtime: 1h 48m
  - runtime: 51m
Demo
Challenges

• Linux monitoring is simple!
  • other OS?

• investigation / fixing of bugs

• additional specific USB-emulators
Conclusion:

vast amount of bugs :-) 

verified using facedancer

USB-fuzzing in practical time frames
Questions?

www.github.com/schumilo

Comrade-in-arms are welcome :-)