VIA: Analyzing Device Interfaces of Protected Virtual Machines

Felicitas Hetzelt (TUB), Martin Radev (AISEC*), Robert Buhren (TUB), Mathias Morbitzer (AISEC), Jean-Pierre Seifert (TUB)
Protected Virtualization (Motivation and Background)

New Technologies: AMD SEV(-ES, -SNP), INTEL TDX

- Protect complete commodity operating system
- Hypervisor excluded from TCB

**Trust Boundary between virtual Devices and protected VM**
Vulnerabilities in the Hardware-OS interface

(virtual) Devices used to be trusted
Vulnerabilities in the Hardware-OS interface

- SoC Peripherals

_BIZ & IT —

Broadcom chip bug opened 1 billion phones to a Wi-Fi-hopping worm attack

Wi-Fi chips used in iPhones and Android may revive worm attacks of old.

DAN GOODIN - 7/28/2017, 9:35 PM
Vulnerabilities in the Hardware-OS interface

- SoC Peripherals
- PCI

Thunderclap flaws impact how Windows, Mac, Linux handle Thunderbolt peripherals

Thunderclap vulnerabilities allow the creation of highly dangerous malicious peripherals that can steal data from OS memory.
Vulnerabilities in the Hardware-OS interface

- SoC Peripherals
- PCI
- USB

These are the bugs that were manually reported before USB fuzzing was integrated into syzbot.

**USB drivers**

- `usbc/core`: memory corruption due to an out-of-bounds access in `usb_destroy_configuration` [fix] [CVE-2017-17558]
- `usbc/net/zd1211rw`: possible deadlock in `zd_chip_disable_rtx` [CVE-2017-17558]
- `usbc/sound`: use-after-free in `__uac_clock_find_source` [fix]
- `usbc/sound`: slab-out-of-bounds in `parse_audio_unit` [fix]
- `usbc/media/em28xx`: use-after-free in `dvb_unregister_frontend` [fix]
- `usbc/media/technisat`: slab-out-of-bounds in `technisat_usb2_rc_query`
- `usbc/media/tm6000`: use-after-free in `tm6000_read_write_usb`
- `usbc/net/qmi_wwan`: divide error in `qmi_wwan_probe/usbnnet_probe` [fix1, fix2] [CVE-2017-16649, CVE-2017-16650]
- `usbc/media/uvc`: slab-out-of-bounds in `uvc_probe`
- `usbc/media/em28xx`: use-after-free in `em28xx_dvb_fini`
- `usbc/media/em28xx`: use-after-free in `v4l2_fh_init`
- `usbc/media/pvrusb2`: WARNING in `pvr2_i2c_core_done/sysfs_remove_group`
VIA’s Goal

Provide a generic tool to analyze the hardware interface of device drivers commonly used in virtual machines to find software vulnerabilities.
VIA’s Approach - Overview

Targeted driver fuzzing tool build on libkl and libfuzzer

- Target drivers loaded as shared library
- Generic VIRTIO, PCI and Platform device stubs
- Configuration files
- Userspace harness
Challenges in Fuzzing the Virtual Device Interface

- Low testcase throughput
  - Delays in driver code
  - Inefficient IO-Interception (VMEXIT, Page-Faults)

- No Interface for Coherent DMA Interception
  - Fresh values need to be provided on each access to coherent DMA area

- In-efficient Interrupt Scheduling
  - Driver stalls until interrupt is scheduled
  - Performance loss when triggering too many interrupts

- State Accumulation
VIA’s Approach - Details

- **Improve Test Case Throughput**
  - Remove delays in driver code (*delay, *sleep, schedule_timeout[_*], time_before/after)

- **IO-Interception**
  - Streaming DMA, MMIO and PIO: adapt existing kernel interfaces (read*, in*, sync_for_cpu, dma_unmap)
  - Coherent DMA: adapt ASAN instrumentation

- **Interrupt Scheduling:**
  - Track “waiting” workloads (wait_for_completion_* , *_wait_event_*) to schedule interrupts at useful execution points

- **Reset State by Reloading Driver in each Iteration**
VIA Performance

- 570 executions/s on average
- 163 improvement on average due to delay reduction
- 2706 improvement in executions/s (1915 without delay optimization) and 2.26 improvement in coverage compared to VM-based approach (Agamotto)

<table>
<thead>
<tr>
<th></th>
<th># Executions / s</th>
<th># Paths</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VIA-D (Increase)</td>
<td>VIA-ND</td>
</tr>
<tr>
<td>8139cp</td>
<td>1.32</td>
<td>1038</td>
</tr>
<tr>
<td>acpi</td>
<td>8.00</td>
<td>71</td>
</tr>
<tr>
<td>e100</td>
<td>63.19</td>
<td>573</td>
</tr>
<tr>
<td>e1000</td>
<td>3.00</td>
<td>1427</td>
</tr>
<tr>
<td>e1000e</td>
<td>0.70</td>
<td>1386</td>
</tr>
<tr>
<td>gve</td>
<td>2.00</td>
<td>147</td>
</tr>
<tr>
<td>ne2k-pci</td>
<td>1408.00</td>
<td>31</td>
</tr>
<tr>
<td>nvme</td>
<td>0.02</td>
<td>260</td>
</tr>
<tr>
<td>qemu-fw-cfg</td>
<td>1254.00</td>
<td>35</td>
</tr>
<tr>
<td>rocker</td>
<td>171.01</td>
<td>181</td>
</tr>
<tr>
<td>sungem</td>
<td>6.01</td>
<td>924</td>
</tr>
<tr>
<td>sunhme</td>
<td>195.00</td>
<td>1025</td>
</tr>
<tr>
<td>tpm-tis</td>
<td>2.00</td>
<td>150</td>
</tr>
<tr>
<td>vio-balloon</td>
<td>1291.00</td>
<td>281</td>
</tr>
<tr>
<td>vio-blk</td>
<td>625.00</td>
<td>333</td>
</tr>
<tr>
<td>vio-console</td>
<td>349.00</td>
<td>352</td>
</tr>
<tr>
<td>vio-crypto</td>
<td>270.00</td>
<td>258</td>
</tr>
<tr>
<td>vio-input</td>
<td>393.00</td>
<td>299</td>
</tr>
<tr>
<td>vio-net</td>
<td>553.00</td>
<td>1250</td>
</tr>
<tr>
<td>vio-rng</td>
<td>1.00</td>
<td>238</td>
</tr>
<tr>
<td>vmxnet3</td>
<td>37.07</td>
<td>51</td>
</tr>
</tbody>
</table>
Bugs

- Analyzed **VIRTIO, PCI** and **Platform** drivers from Qemu devices and **Google confidential VM (SEV)**
  - ~50 bugs across 22 analyzed drivers (2 drivers had no issues)
  - Missing sanitization
  - Incomplete / failed initialization
  - Shared control data

- **Exploitability:**
  - 23/50 bugs likely not exploitable
  - HV has advanced exploitation capabilities

<table>
<thead>
<tr>
<th>Bug Class</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out-of-Bounds access</td>
<td>14</td>
</tr>
<tr>
<td>Invalid memory access</td>
<td>10</td>
</tr>
<tr>
<td>Slab management</td>
<td>8</td>
</tr>
<tr>
<td>Device-shared pointer</td>
<td>5</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>3</td>
</tr>
<tr>
<td>Assertion failure (BUG)</td>
<td>4</td>
</tr>
<tr>
<td>Unbounded allocation</td>
<td>5</td>
</tr>
<tr>
<td>Deadlock</td>
<td>1</td>
</tr>
</tbody>
</table>
Device-shared pointer in vmxnet3

- Pointer to sk_buff placed in DMA memory area
- Device overwrites sk_buff pointer to point to device controlled memory
- Device points destructor function pointer to code gadget
- Gadget pivots stack to ROP chain in device controlled memory

```c
struct vmxnet3_rx_buf_info {
    struct sk_buff *skb; // shared ptr with device
    ...
};
struct sk_buff {
    void (*destructor)(struct sk_buff *skb);
    ...
};
void skb_release_head_state(struct sk_buff *skb) {
    if (skb->destructor) skb->destructor(skb);
}
```
Use-After-Free in `virtio_net`

- Device induces `virtnet_probe` fail; `virtio_device` is freed; error value is not set
- Device induces overlapping allocation of e1000 eeprom
Limitations and Summary

- Applying VIA to 22 device drivers uncovered a large amount of vulnerabilities undermining the efficacy of protected virtualization technologies.

- Intel TDX implements device white lists to limit the virtual device attack surface, however:
  - Many bugs affect drivers that are included in the white list
  - Cloud providers might have individual hardware requirements. E.g. none of the devices in the while list are used in the Google Confidential VM

- Limitations / Future Work:
  - No Concurrency
  - State Accumulation
  - Improved Fuzzing Methods
Thank you

https://github.com/file-citas/via