Fuzzing Device Interfaces of Protected Virtual Machines

Felicitas Hetzelt (TUB), Martin Radev (AISEC*), Robert Buhren (TUB), Mathias Morbitzer (AISEC), Jean-Pierre Seifert (TUB)

Background

- AMD SEV(-ES, -SNP), INTEL TDX
- Protect complete commodity operating system
- Hypervisor excluded from TCB
  - Guest memory protection via encryption, integrity protection, access restriction
  - Guest state protection
  - E/Nested - Page table integrity protection (SEV-SNP)
- Virtual device interface:
  - Device whitelists (TDX)
  - swiotlb

Trust Boundary between Virtual HW and OS

- (virtual) Devices used to be trusted
  - Data received via DMA, MMIO/PIO lacks sanitization
  - Secret or control data (e.g. kernel pointers) shared with or controlled by device
  - Unhandled initialization failures

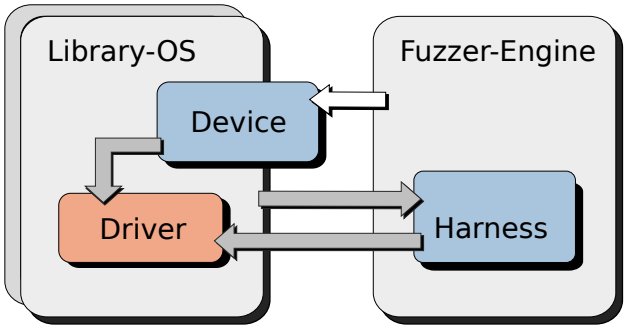
Fuzzing the HW-OS interface

- Delays in driver code
- IO interface / IO interception
- IRQ scheduling

- State accumulation

- Shallow and non-exploitable bugs / BUG() error handling
- Coverage stability

Current State



- Targeted driver fuzzing tool build on lkl and libfuzzer
  - Target drivers loaded as shared library
  - VIRTIO, PCI and Platform device stubs
  - Configuration files
  - Userspace harness
- IO interception:
  - Streaming DMA, MMIO and PIO → kernel interfaces (read*, in*, sync_for_cpu, dma_unmap)
  - Coherent DMA → ASAN
- Remove delays (*delay, *sleep, schedule_timeout[_*], time_before/after)
- Track “waiting” workloads to schedule IRQs (wait_for_completion_*, *_wait_event_*)
- Load/Unload driver in each fuzzing iteration to reset state
- ~570 execution/s on average (1 thread)
- Code: https://github.com/file-citas/via
- Paper: Soon

Design Considerations

- Move to VM setup
  - Existing device implementations
  - Concurrency / Race-Conditions
  - Lowlevel Interfaces (?)
- Keep in userspace
  - Libfuzzer features (Dataflow tracing)
  - Symbolic Execution
  - Overhead
- Other Interfaces:
  - HV-UEFI/BIOS
  - QEMU Firmware Configuration
  - Special emulated instructions
- QEMU device interface layer to re-use device code
- Static analysis (smatch, [1])

[1] Static Detection of Unsafe DMA Accesses in Device Drivers: https://www.usenix.org/conference/usenixsecurity21/presentation/bai

Backup Slides

BUGS

- Analyzed VIRTIO, PCI and Platform drivers from qemu devs and google confidential vm (SEV)
  - ~50 bugs across 22 analyzed drivers (2 drivers had no issues): https://github.com/fuzzsa/fuzzsa-bugs
  - Incomplete / failed initialization
  - Missing sanitization
  - Shared control data (pointers, indices, bounds)
  - Some integer under/overflows in the net core code
  - One userspace bug in e100
- Exploitability:
  - Many bugs are not exploitable / lead only to VM-crash (invalid access, reachable assertions, …)
  - HV has advanced capabilities to combine bugs, infer guest execution state / memory layout and location of allocated buffers
  - Build POCs for:
    - Shared pointer
    - Use after free
    - OOB write

Bug Class	Count
Out-of-Bounds access	14
Invalid memory access	10
Slab management	8
Device-shared pointer	5
Miscellaneous	3
Assertion failure (BUG)	4
Unbounded allocation	5
Deadlock	1

Driver	Usage	Type	Bus	#Bugs	
				Low	High
virtio_ring	QVD	Common	VIO	2	4
virtio_net	QVD	NET	VIO	1	1
virtio_blk	QVD	BLK	VIO	5	1
virtio_crypto	QVD	CRYPTO	VIO	2	0
virtio_rng	QVD	RNG	VIO	0	1
virtio_console	QVD	MISC	VIO	1	0
virtio_balloon	QVD	MISC	VIO	2	1
virtio_input	QVD	MISC	VIO	0	0
rocker	QVD	NET	PCI	1	3
sungem	QVD	NET	PCI	1	0*
sunhme	QVD	NET	PCI	2	0*
8139cp	QVD	NET	PCI	1	0
vmxnet3	QVD	NET	PCI	0	5
ne2k-pci	QVD	NET	PCI	0	0
e100	QVD	NET	PCI	0	1
e1000	QVD	NET	PCI	0	3
e1000e	QVD	NET	PCI	1	0
qemu_fw_cfg	QVD	MISC	PLT	1	0
acpi	QVD	MISC	PLT	0	1
gve	CVM	NET	PCI	2	3
nvme	CVM	BLK	PCI	1	1
tpm_tis	CVM	TPM	PLT	0	2

* Device-controlled values passed to `swiotlb_tbl_unmap_single` as described in Section 6.2 were not counted as separate bugs.

Device configuration

```
module="e1000.ko"
harness="harness_net.so"
devtype=0
interface="eth0"
vid=0x8086
did=0x1075
moddeps=[]
barsizes=[0x1fffff]
barflags=[0x40200]
```

```
module="virtio_blk.ko"
harness="harness_blk_char.so"
devtype=1
nqueues=4
vid=0x0
did=0x2
fuzz_dma=0
vio_nofuzz=[0x010, 0x000, 0x034, 0x070,
0x060, 0x44]
features_set_mask_low=0x20000
features_set_mask_high=0x1
features_unset_mask_low=0x3a24
features_unset_mask_high=0x4
moddeps=["virtio.ko", "virtio_ring.ko",
"virtio_mmio.ko"]
```

```
module="tpm_tis.ko"
harness="harness_blk_char.so"
plt_name="tpm_tis"
devtype=2
moddeps=["rng-core.ko", "tpm.ko",
"tpm_tis_core.ko"]
barsizes=[0x1fffff, 0x1fffff, 0x1fffff,
0x1fffff]
barflags=[0x40200, 0x40200, 0x40200,
0x40200]
```

Userspace Harness

- Init / Fuzz functions
- load/unload module
- Trigger IRQs
- 3 harness implementations:
  - Basic
  - Network
  - Block / Char

```
#include "fuzz_interface.h"
#include "util.h"
int mod_init(void) {
	start_irqthread_default();
	return 0;
}
int mod_fuzz(const uint8_t *data, size_t size) {
	int err;
	start_fuzz(data, size);
	err = init_module();
	if(err!=0) {
		goto out_noinit;
	}
	trigger_irq();
	uninit_module();
out_noinit:
	end_fuzz();
	return 0;
}
```

	# Executions / s			# Blocks	
	VIA-D	VIA-ND (Increase)		VIA-D	VIA-ND
8139cp	1.32	122.41	×**92.58**	1038	1040
acpi	8.00	8.00	×1.0	71	71
e100	63.19	231.98	×**3.67**	573	569
e1000	3.00	259.06	×**86.35**	1427	1535
e1000e	0.70	111.25	×**158.92**	1386	1579
gve	2.00	636.22	×**318.11**	147	594
ne2k-pci	1408.00	1658.00	×**1.18**	31	31
nvme	0.02	0.88	×**44.0**	260	291
qemu-fw-cfg	1254.00	1341.0	×**1.06**	35	37
rocker	171.01	203.25	×**1.19**	181	184
sungem	6.01	59.04	×**9.82**	924	1032
sunhme	195.00	428.00	×**2.19**	1025	1030
tpm-tis	2.00	857.00	×**428.50**	150	326
vio-balloon	1291.00	1328.00	×**1.03**	281	281
vio-blk	625.00	624.00	×1.00	333	333
vio-console	349.00	444.00	×**1.27**	352	352
vio-crypto	270.00	277.00	×**1.03**	258	258
vio-input	393.00	635.00	×**1.62**	299	299
vio-net	553.00	400.00	×0.72	1250	1257
vio-rng	1.00	2282.00	×**2282.00**	238	239
vmxnet3	37.07	59.94	×**1.62**	51	51

	TTB (s)		TTB (s)		TTB (s)
8139cp	4.60	acpi	1.87	e100	1424.80
e1000	51.54	e1000e	1650.98	gve	542.82
nvme	0.37	qemu-fw	2.03	rocker	6.78
sungem	1.50	tpm-tis	*7200+	vio-balloon	37.86
vio-blk	20.24	vio-console	18.28	vio-crypto	18.92
vio-net	4.93	vio-ring	132.34	vio-rng	2.08
vmxnet3	1.89				

* Estimate based on few runs, since the bug could not be triggered reliably.