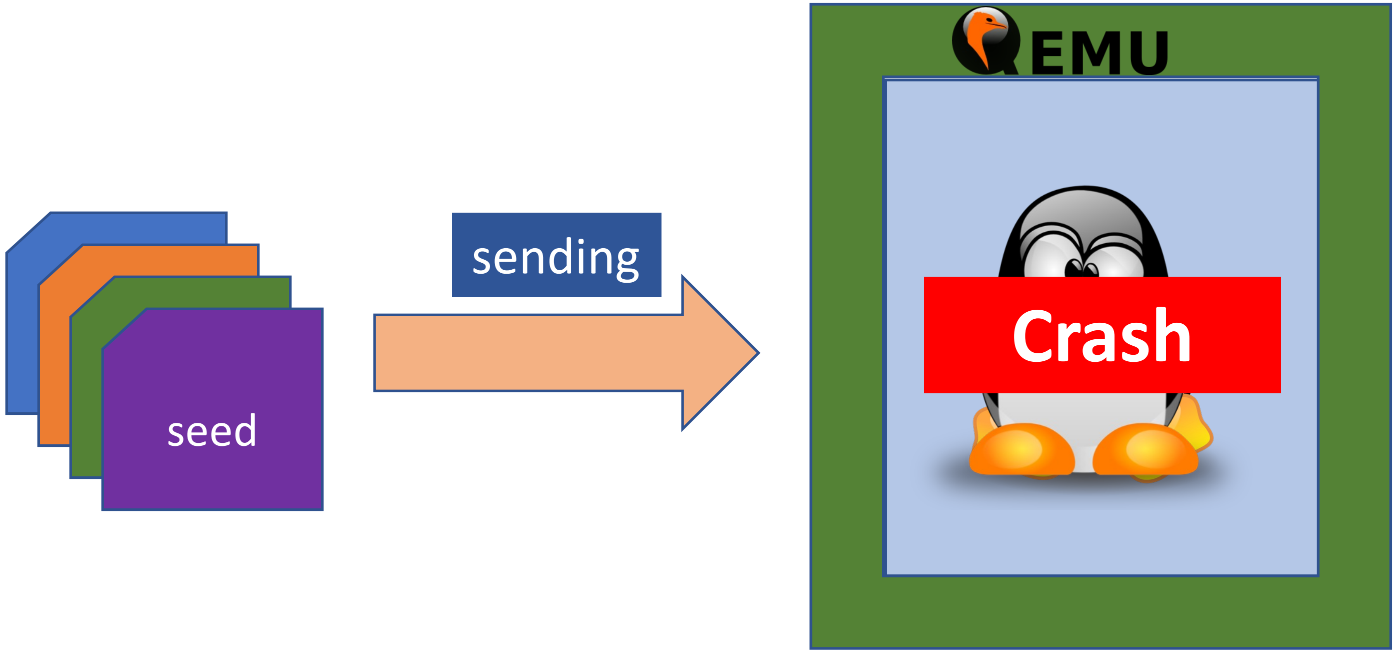
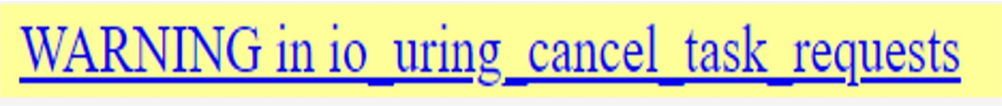
SyzScope: Revealing High-Risk Security Impacts of Fuzzer-Exposed Bugs in Linux kernel

Xiaochen Zou, Guoren Li, Weiteng Chen, Hang Zhang, Zhiyun Qian

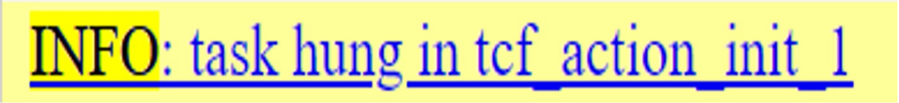
Linux kernel bugs



KASAN: use-after-free Read in sctp_auth_free

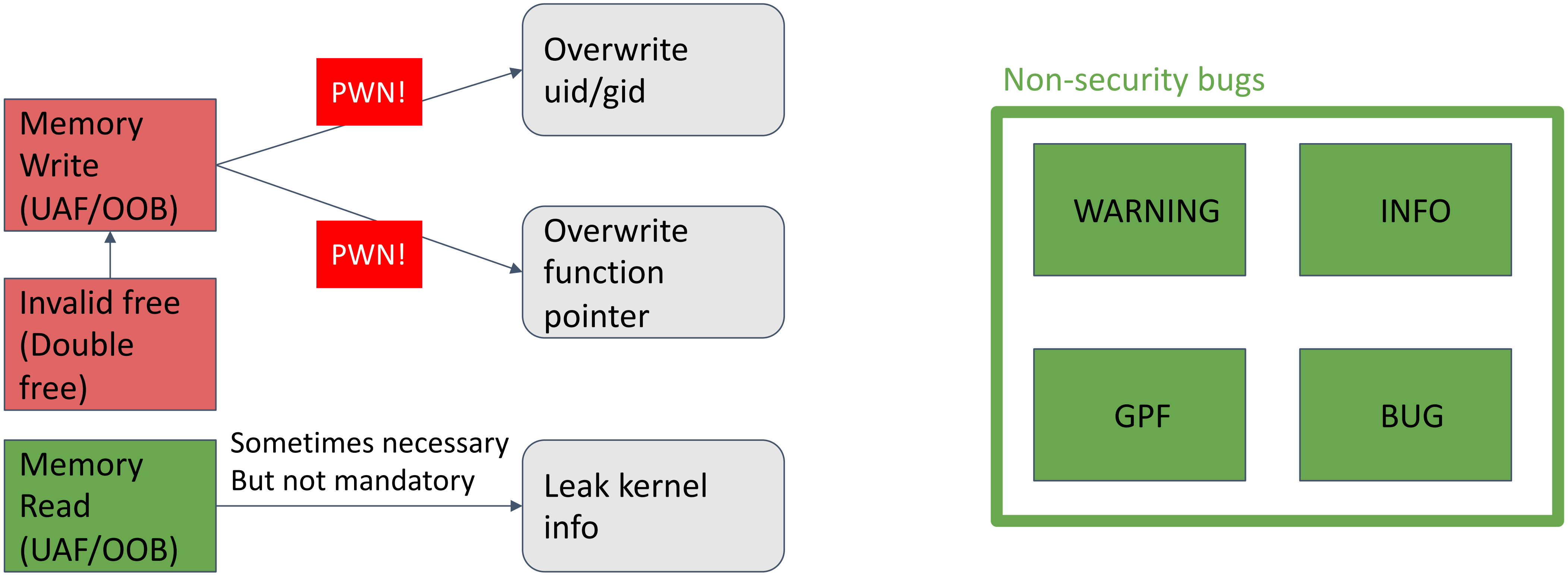

WARNING in io_uring_cancel_task_requests

general protection fault in strncasecmp

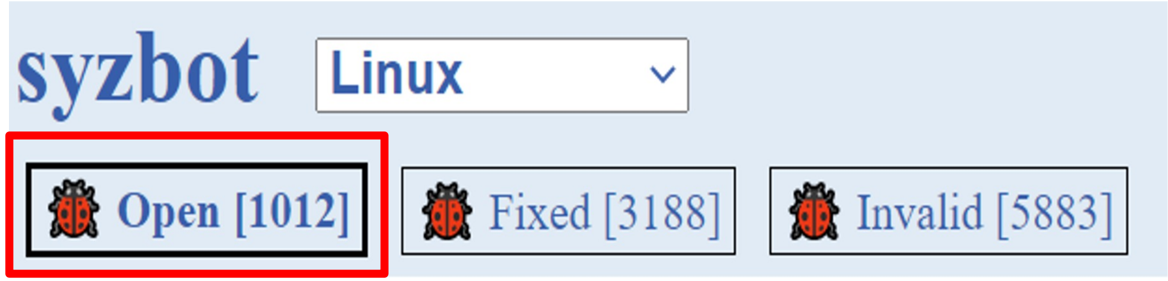

INFO: task hung in tcf_action_init_1

BUG: receive list entry not found for dev vxcan1, id 003, mask C000...

Bug’s security impacts



Too many bugs to fix



UAF Read
63 days

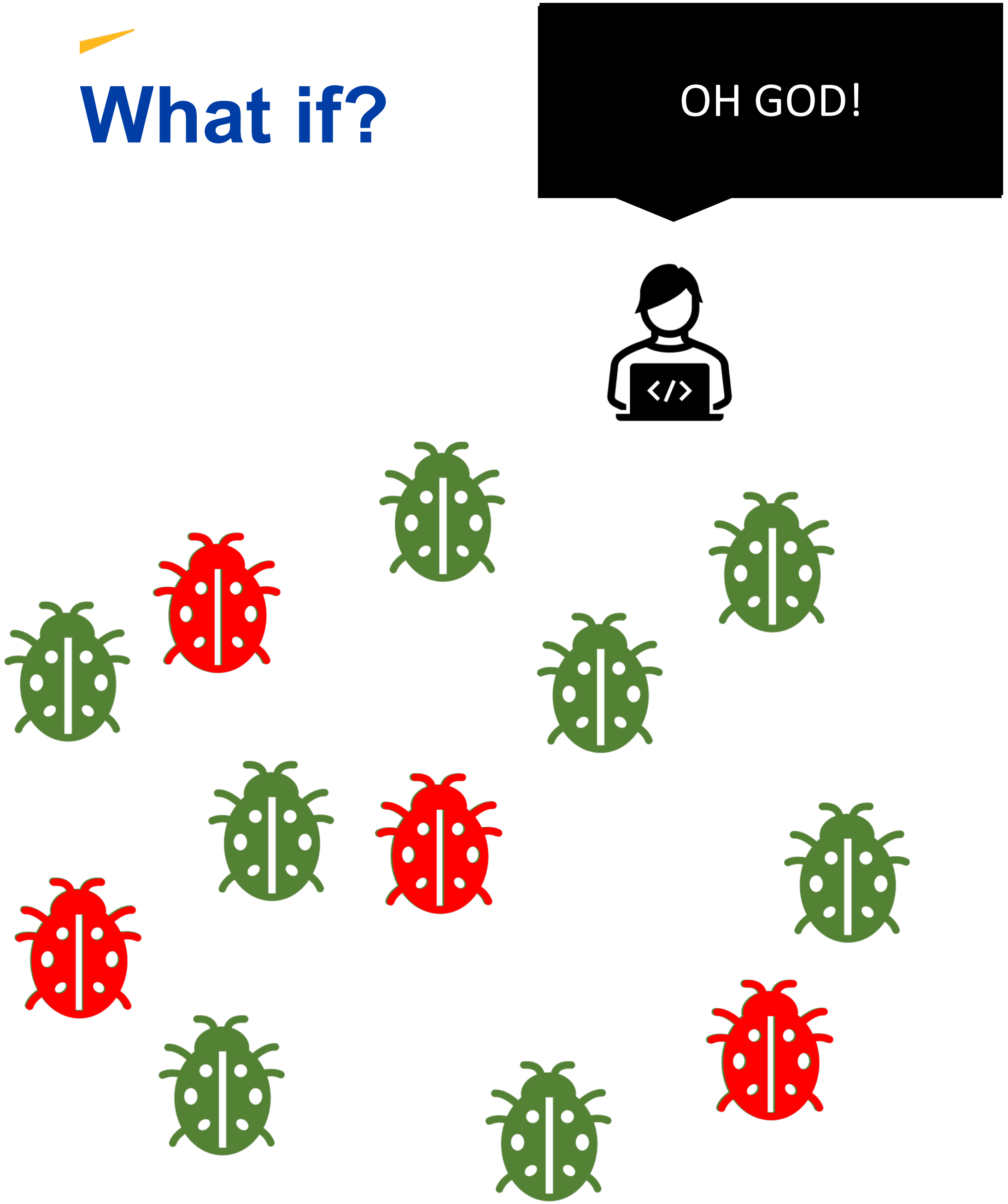
UAF Write
37 days

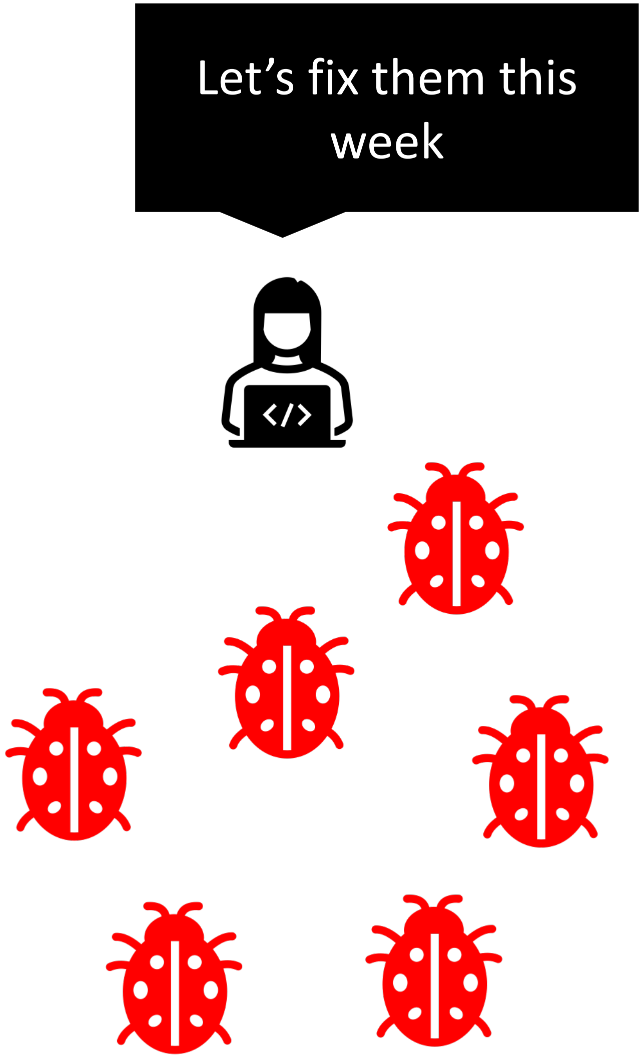
OOB Read
89 days

OOB Write
29 days

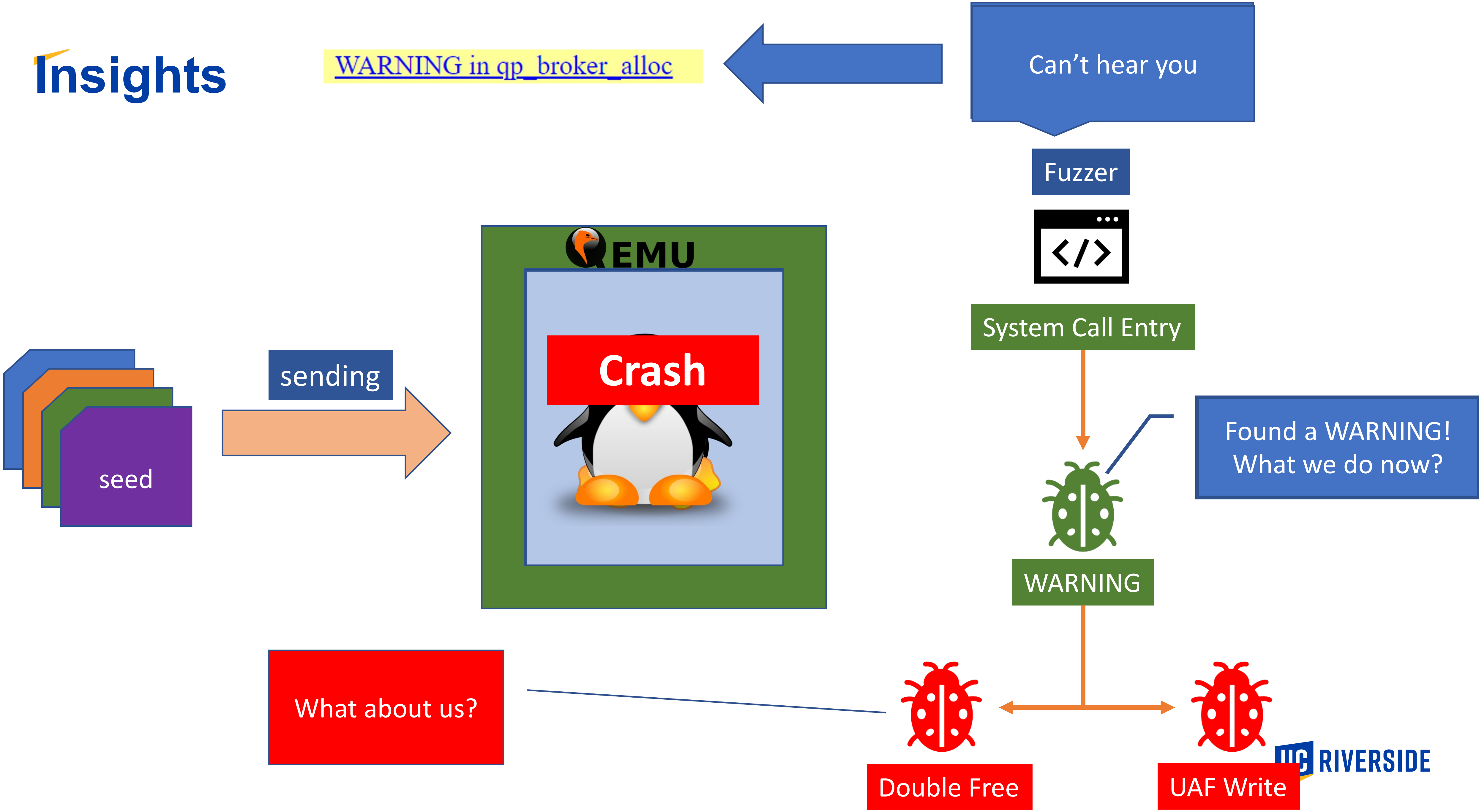
Prioritize limited resources to fix high-risk bugs

What if?





Insights



WARNING in qp_broker_alloc

Can’t hear you

Found a WARNING! What we do now?

What about us?

Insights

WARNING in qp_broker_alloc

Can’t hear you

Fuzzing only present the first impact of a bug

But the first impact may not represent the most critical impact

Insight

High-risk impacts:

- UAF/OOB Write
- Invalid Free
- Control flow hijacking
- Arbitrary/Constrained value write
- Arbitrary/Constrained address write

Follow up impacts

Low-risk impacts:

- UAF/OOB Read
- WARNING/INFO
- General protection fault
- BUG
- All other non-security bugs

Blue means this impact **can** be detected by fuzzing

Red means this impact **can not** be detected by fuzzing

Motivating example

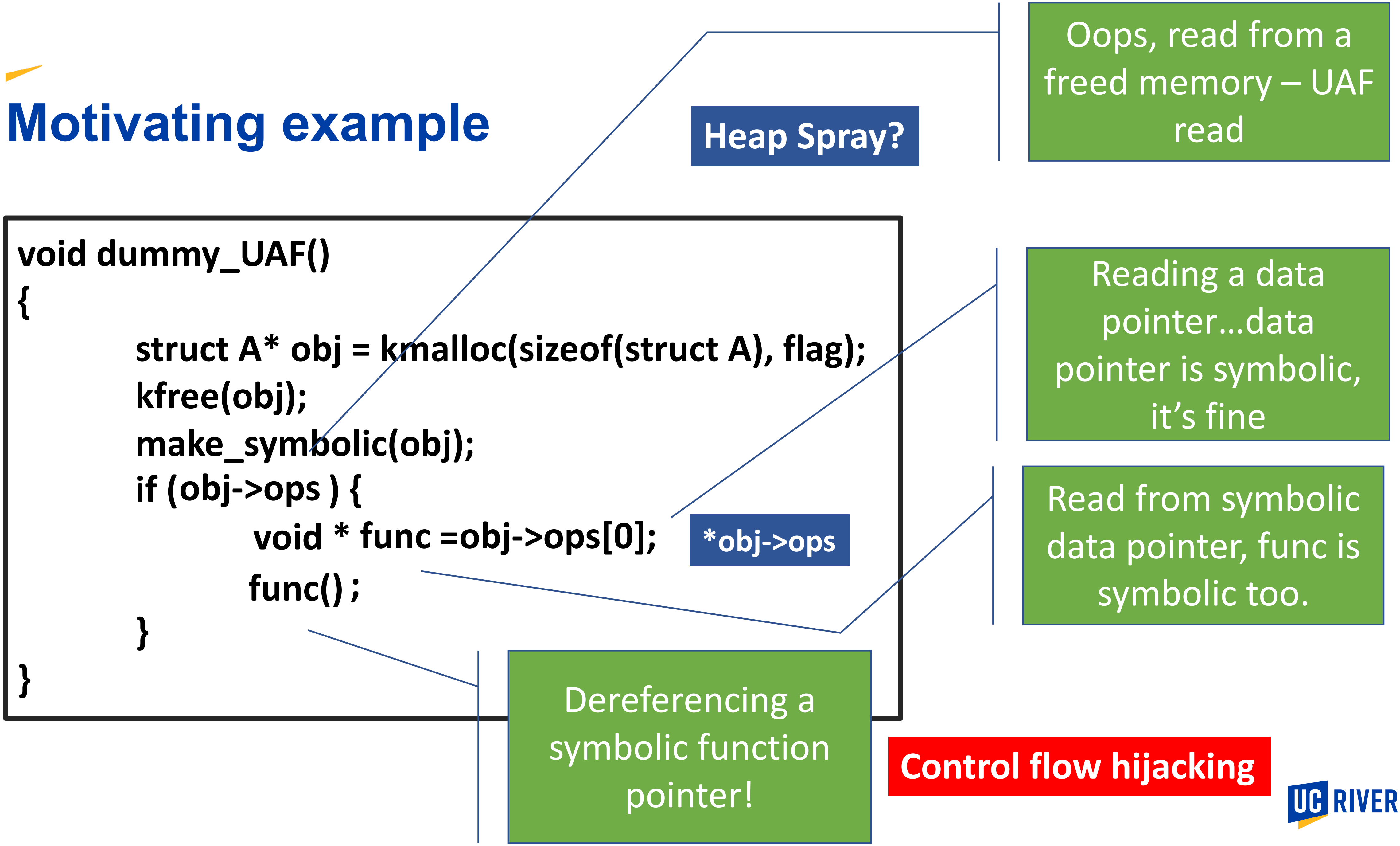
```
void dummy_UAF()
{
        struct A* obj = kmalloc(sizeof(struct A), flag);
        kfree(obj);

        if (obj->ops ) {
                void *func = obj->ops[0];
                func();
        }
}
```

*obj->ops

Oops, use-after-free Read, but let’s continue executing

Reading a data pointer…wait, it’s not a valid memory address

Motivating example



```
void dummy_UAF()
{
        struct A* obj = kmalloc(sizeof(struct A), flag);
        kfree(obj);
        make_symbolic(obj);
        if (obj->ops ) {
                void * func =obj->ops[0];
                func() ;
        }
}
```

Heap Spray?

Oops, read from a freed memory – UAF read

Reading a data pointer…data pointer is symbolic, it’s fine

*obj->ops

Read from symbolic data pointer, func is symbolic too.

Dereferencing a symbolic function pointer!

Control flow hijacking

Workflow - Fixed bugs & Open bugs

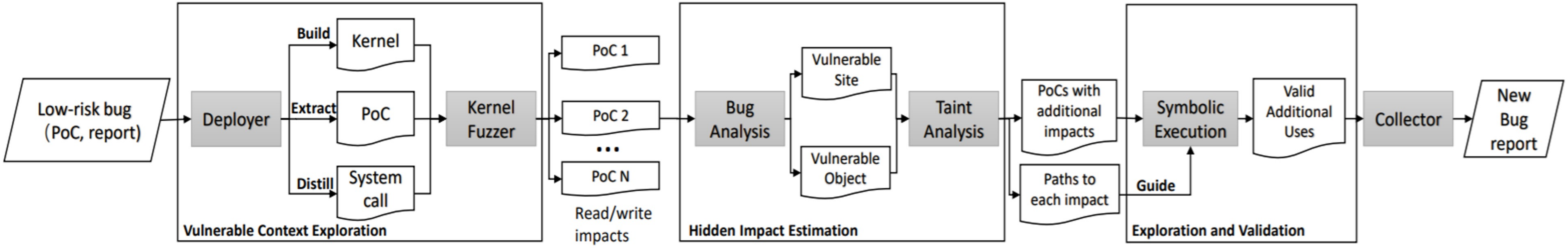


Figure 2: Workflow

Evaluation

1170 valid bugs from syzbot

183

Low-risk bugs to high-risk

Syzbot only found around 170 high-risk bugs in the past 4 years, we double the number of high-risk bugs

15

High-risk WARN & INFO

SyzScope found 15 WARNING and INFO bugs had strong security impacts

17

High-risk GPF & BUG

SyzScope found 17 general protection fault bugs and BUG bugs had strong security impacts

179

Control flow hijacking

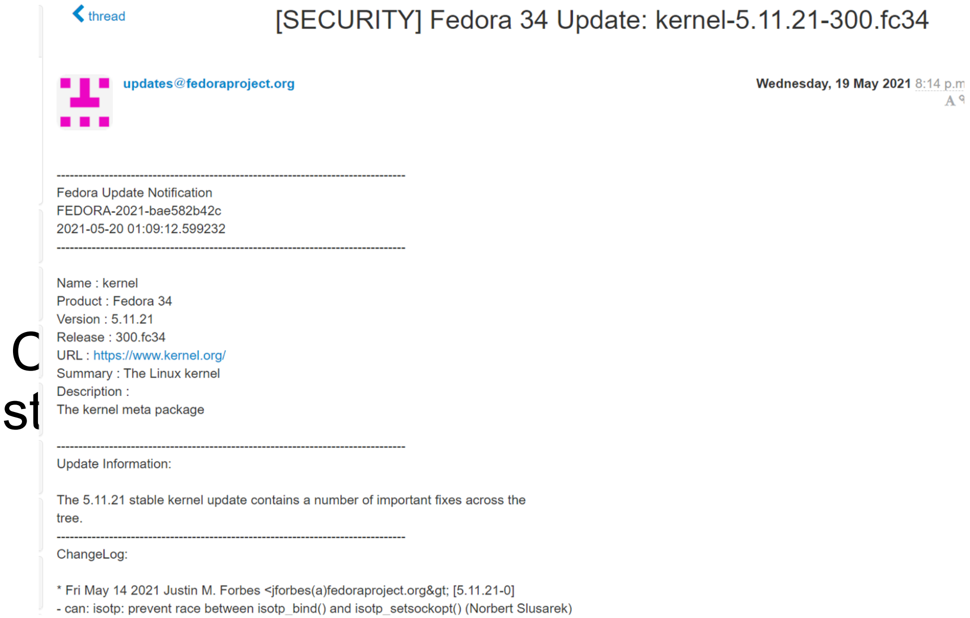
SyzScope discovered 179 control flow hijacking among 183 seemingly low-risk bugs

Overall results

		High-risk bug found	OOB/UAF write	Arbitrary address write	Constrained address write	Arbitrary value write	Constrained value write	Control flow hijacking	Invalid Free
Fixed	**GPF and BUG**	17	71	124	62	29	20	8	9
	WARNING and INFO	15	85	166	66	20	30	9	3
	UAF and OOB Read	99	319	1490	446	271	153	104	83
Open	**GPF and BUG**	4	4	0	0	2	0	0	2
	WARNING and INFO	10	97	213	91	47	22	18	13
	UAF and OOB Read	38	151	381	113	43	22	40	18
	Total	183	727	2374	778	410	247	179	128

Disclosure

We submitted 32 high-risk bugs since Linux kernel v4.19 to C maintainers and 8 of them have been assigned CVE, the rest still pending for responses.



[SECURITY] Fedora 34 Update: kernel-5.11.21-300.fc34

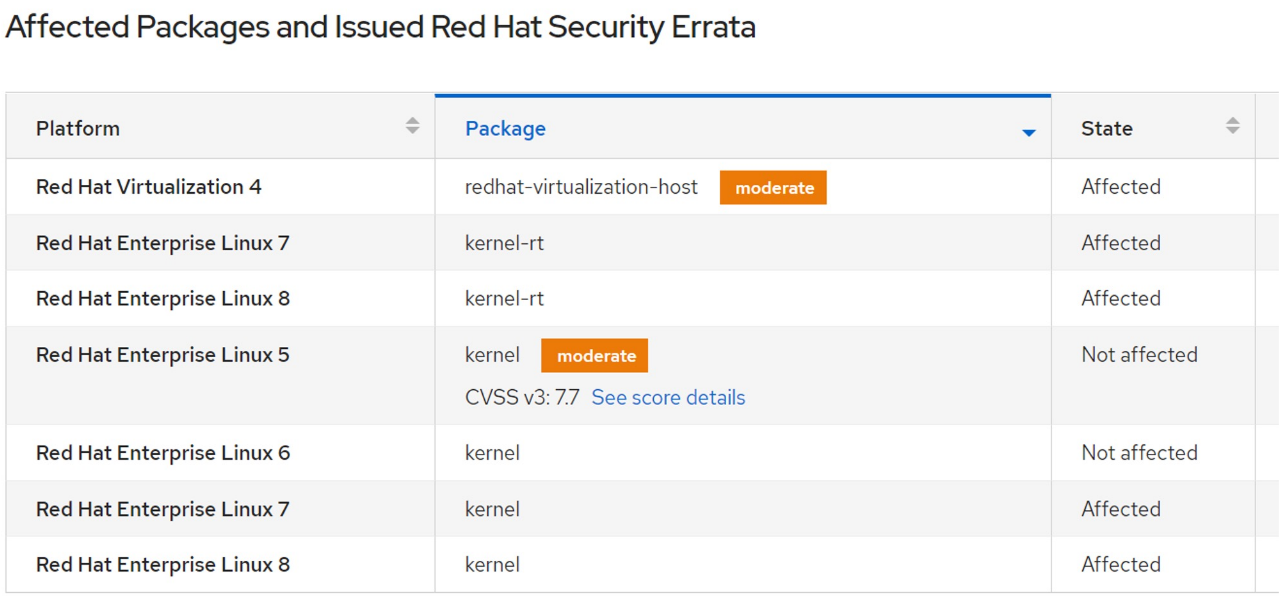
updates@fedoraproject.org

Wednesday, 19 May 2021 8:14 p.m

```
--------------------------------------------------------------------------------
Fedora Update Notification
FEDORA-2021-bae582b42c
2021-05-20 01:09:12.599232
--------------------------------------------------------------------------------

Name : kernel
Product : Fedora 34
Version : 5.11.21
Release : 300.fc34
URL : https://www.kernel.org/
Summary : The Linux kernel
Description :
The kernel meta package

--------------------------------------------------------------------------------
Update Information:

The 5.11.21 stable kernel update contains a number of important fixes across the
tree.
--------------------------------------------------------------------------------
ChangeLog:

* Fri May 14 2021 Justin M. Forbes <jforbes(a)fedoraproject.org&gt; [5.11.21-0]
- can: isotp: prevent race between isotp_bind() and isotp_setsockopt() (Norbert Slusarek)
```

- CVE-2021-33034
- CVE-2021-33033
- CVE-2019-25044
- CVE-2020-36386
- CVE-2020-36385
- CVE-2018-25015
- CVE-2020-36387
- CVE-2019-25045

Source Package
linux (PTS)
linux-4.19 (PTS)



Affected Packages and Issued Red Hat Security Errata

Platform	Package	State
Red Hat Virtualization 4	redhat-virtualization-host moderate	Affected
Red Hat Enterprise Linux 7	kernel-rt	Affected
Red Hat Enterprise Linux 8	kernel-rt	Affected
Red Hat Enterprise Linux 5	kernel moderate CVSS v3: 7.7 See score details	Not affected
Red Hat Enterprise Linux 6	kernel	Not affected
Red Hat Enterprise Linux 7	kernel	Affected
Red Hat Enterprise Linux 8	kernel	Affected

Q&A

Thank you for listening

Access my portfolio

Twitter: @ETenal7

ARTIFACT EVALUATED
usenix ASSOCIATION
FUNCTIONAL

ARTIFACT EVALUATED
usenix ASSOCIATION
AVAILABLE

Looking for summer intern for 2023

UC RIVERSIDE