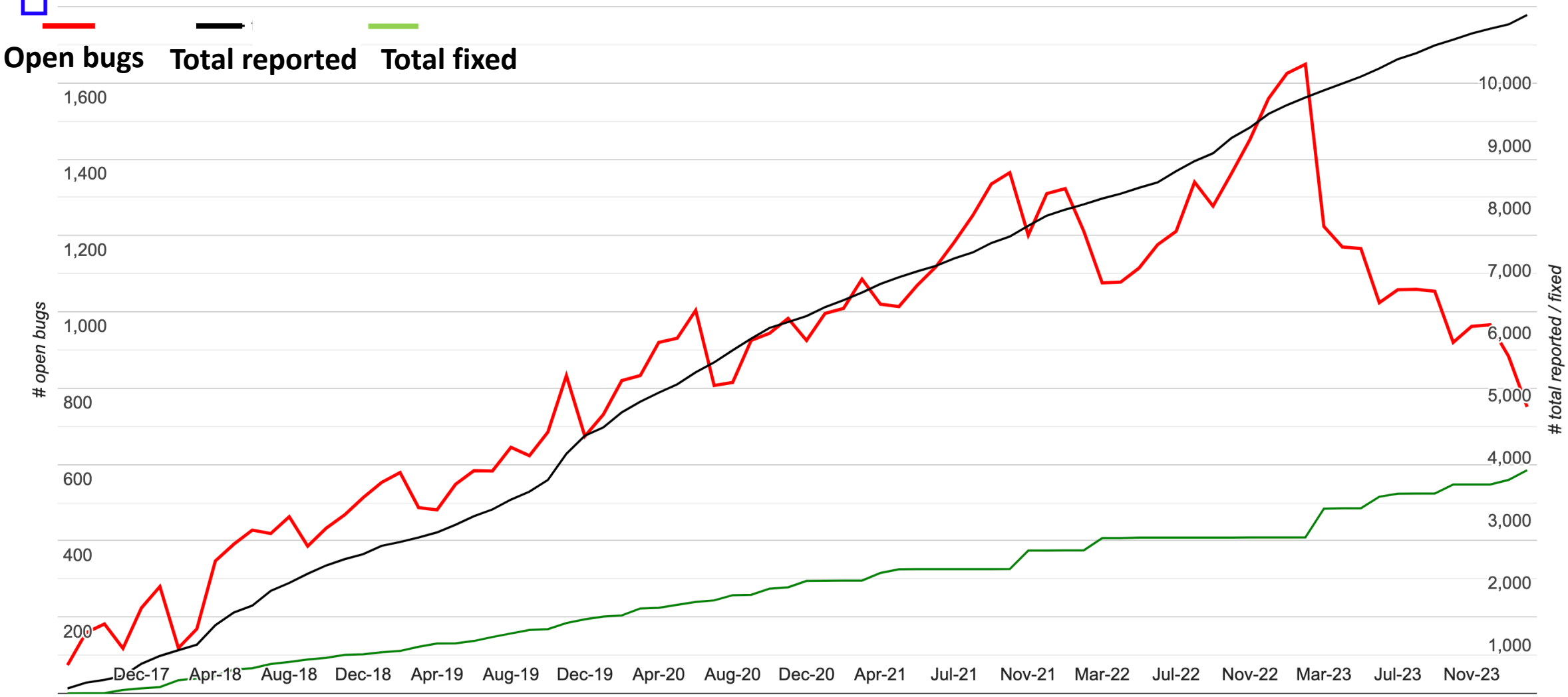
SyzBridge: Bridging the Gap in Exploitability Assessment of Linux Kernel Bugs in the Linux Ecosystem

Xiaochen Zou, Yu Hao, Zheng Zhang, Juefei Pu, Weiteng Chen, Zhiyun Qian

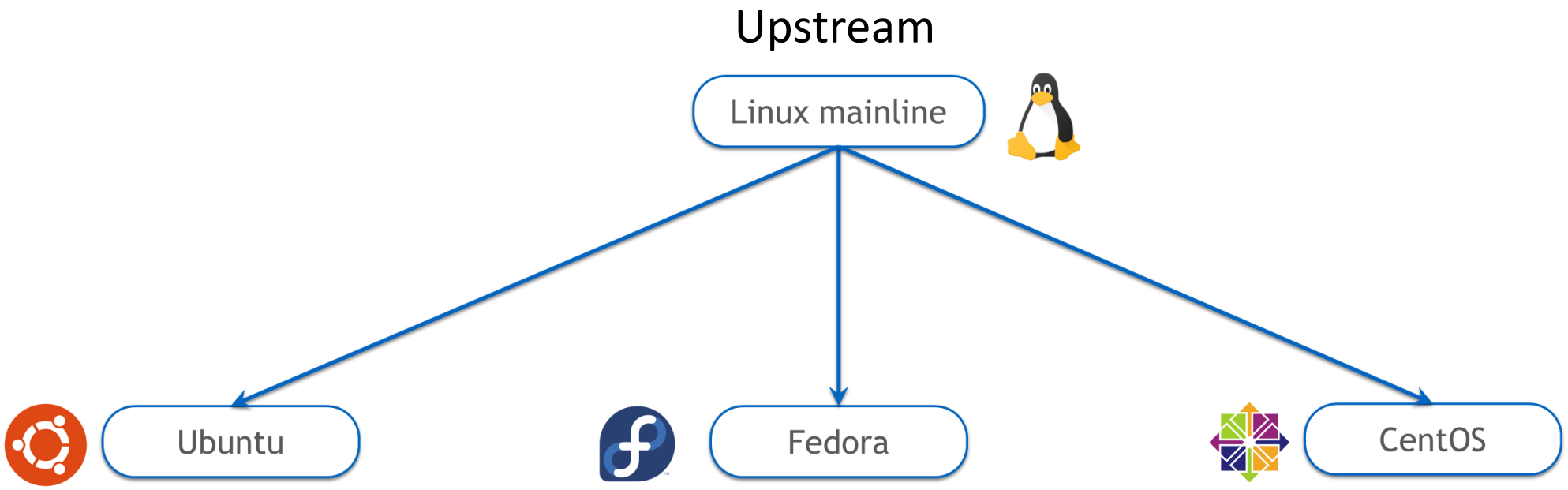
UC RIVERSIDE

Is Linux kernel safe?

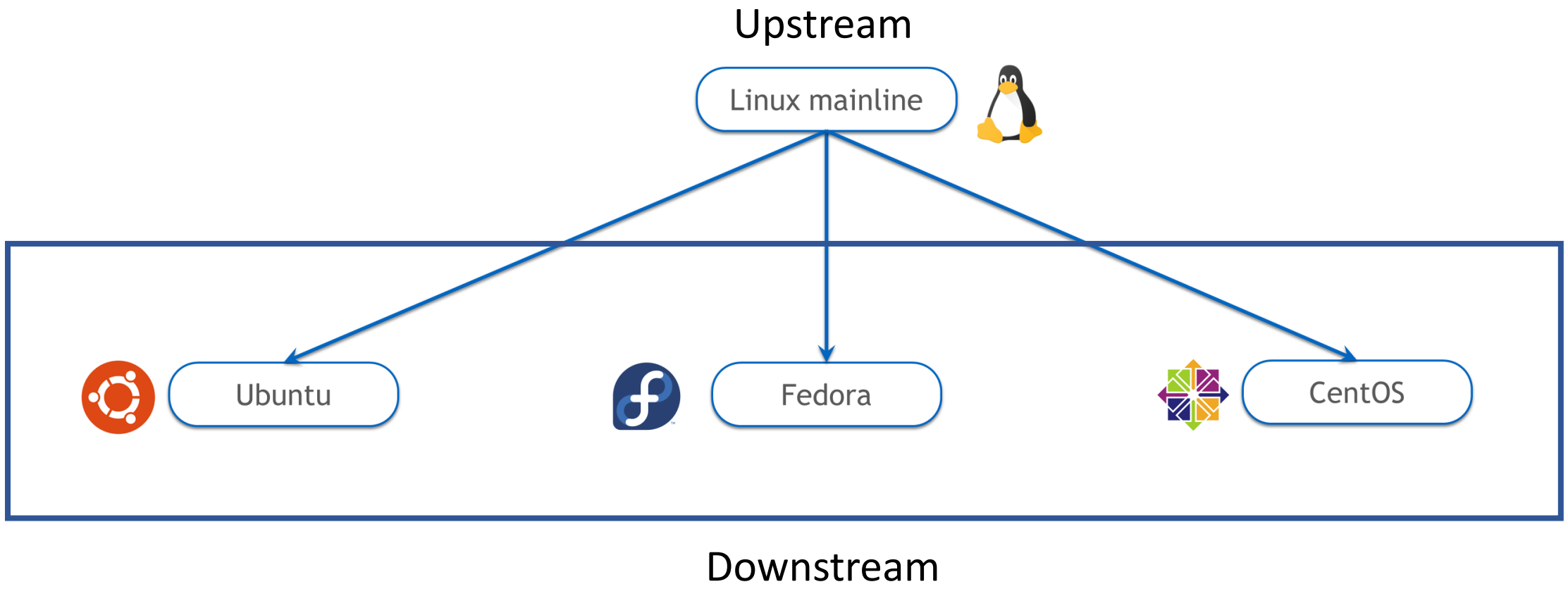
Bugs found by syzbot



Downstream Distros



Downstream Distros



Upstream bugs impact

High-risk upstream bugs: ~183

Valid bugs*	High-risk bugs*
215	17
293	15
202	99
83	4
292	10
85	38
1170	183

Table from {SyzScope: Revealing High-Risk Security Impacts of Fuzzer-Exposed Bugs in Linux kernel} USENIX 2023

Real-world Exploits from syzbot: ~5

CVE-2022-0185
CVE-2021-22600
CVE-2021-22555
CVE-2021-4154
CVE-2021-3715

Upstream bugs impact

High-risk upstream bugs: ~183

Valid bugs*	High-risk bugs*
215	17
293	15
202	99
83	4
292	10
85	38
1170	183

15.6% upstream bugs are high-risk

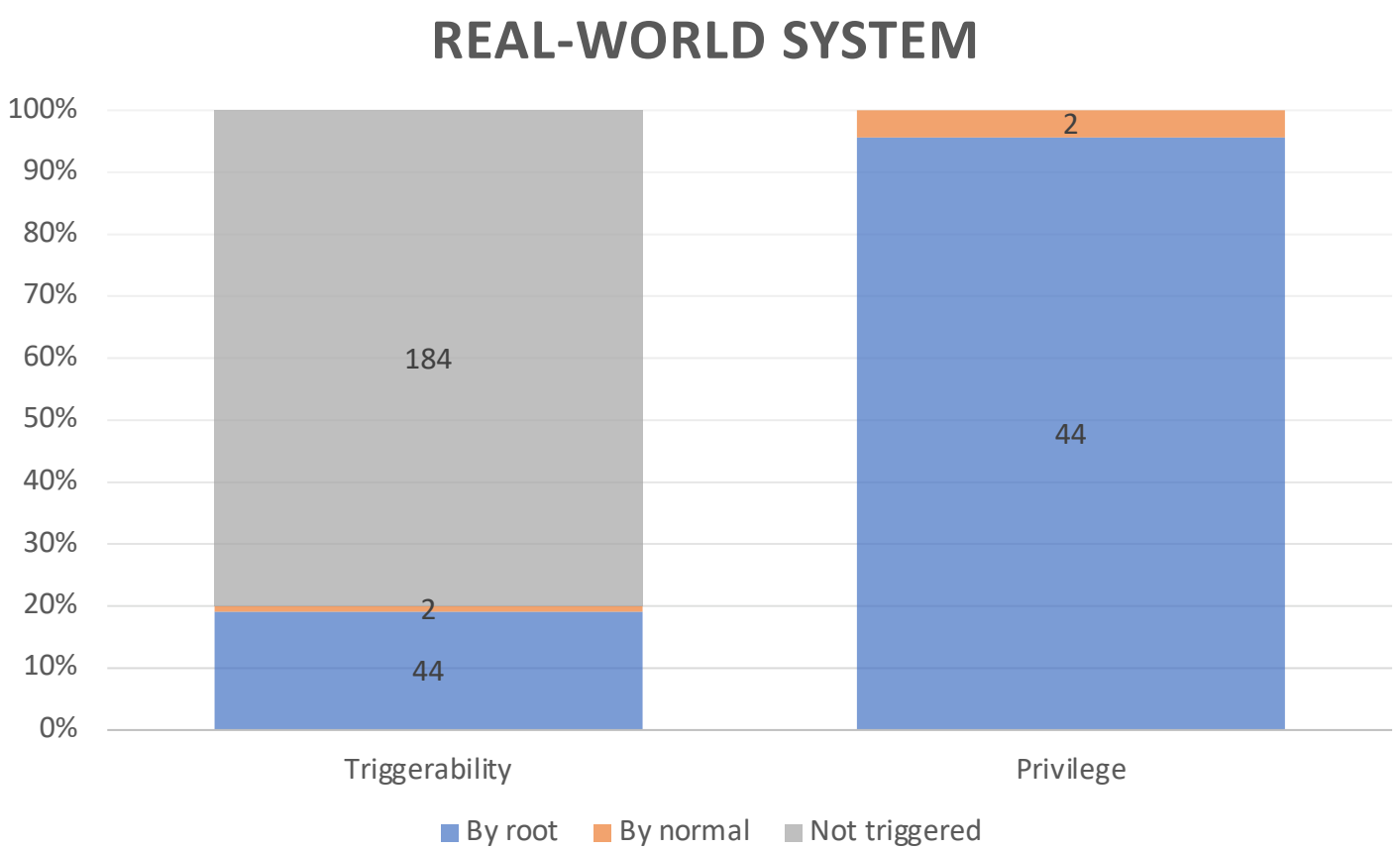
Real-world Exploits from syzbot: ~5

CVE-2022-0185
CVE-2021-22600
CVE-2021-22555
CVE-2021-4154
CVE-2021-3715

Table from {SyzScope: Revealing High-Risk Security Impacts of Fuzzer-Exposed Bugs in Linux kernel} USENIX 2023

UC RIVERSIDE

Upstream V.S. Downstream



230 upstream bugs

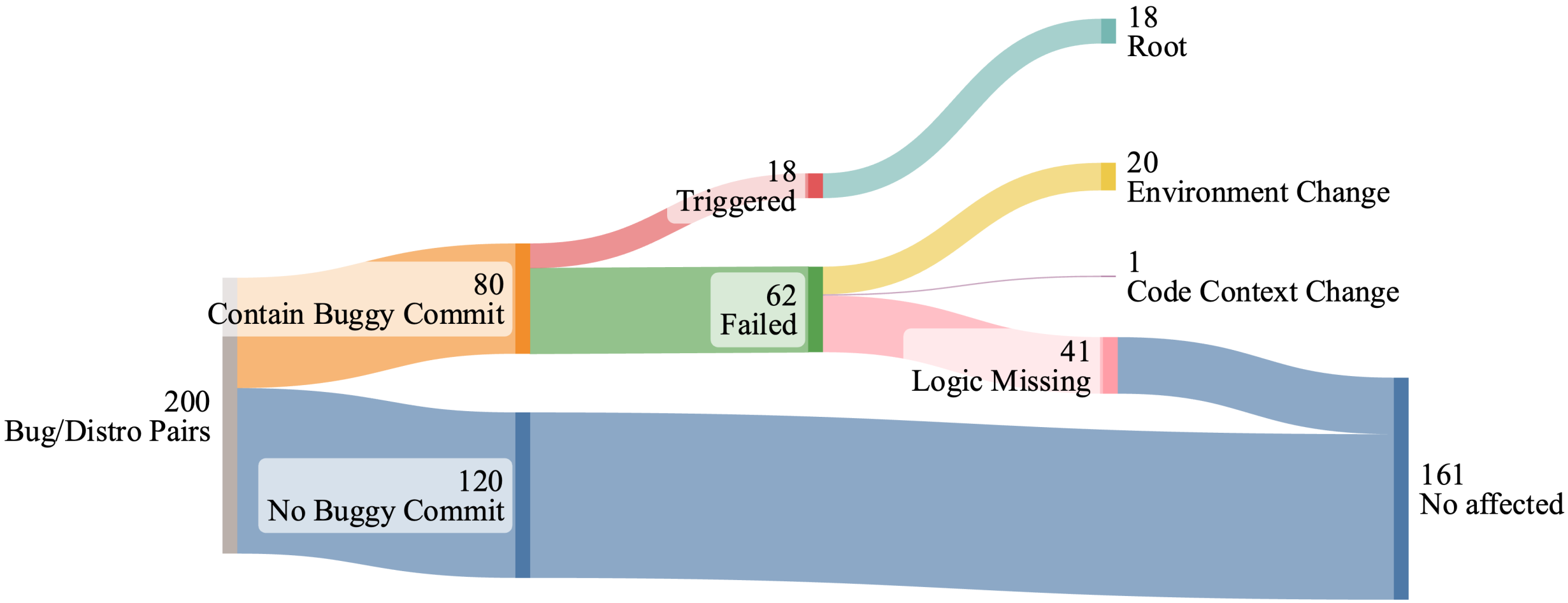
43 downstream distros

Triggered by root: 19.1%

Triggered by normal: 0.9%

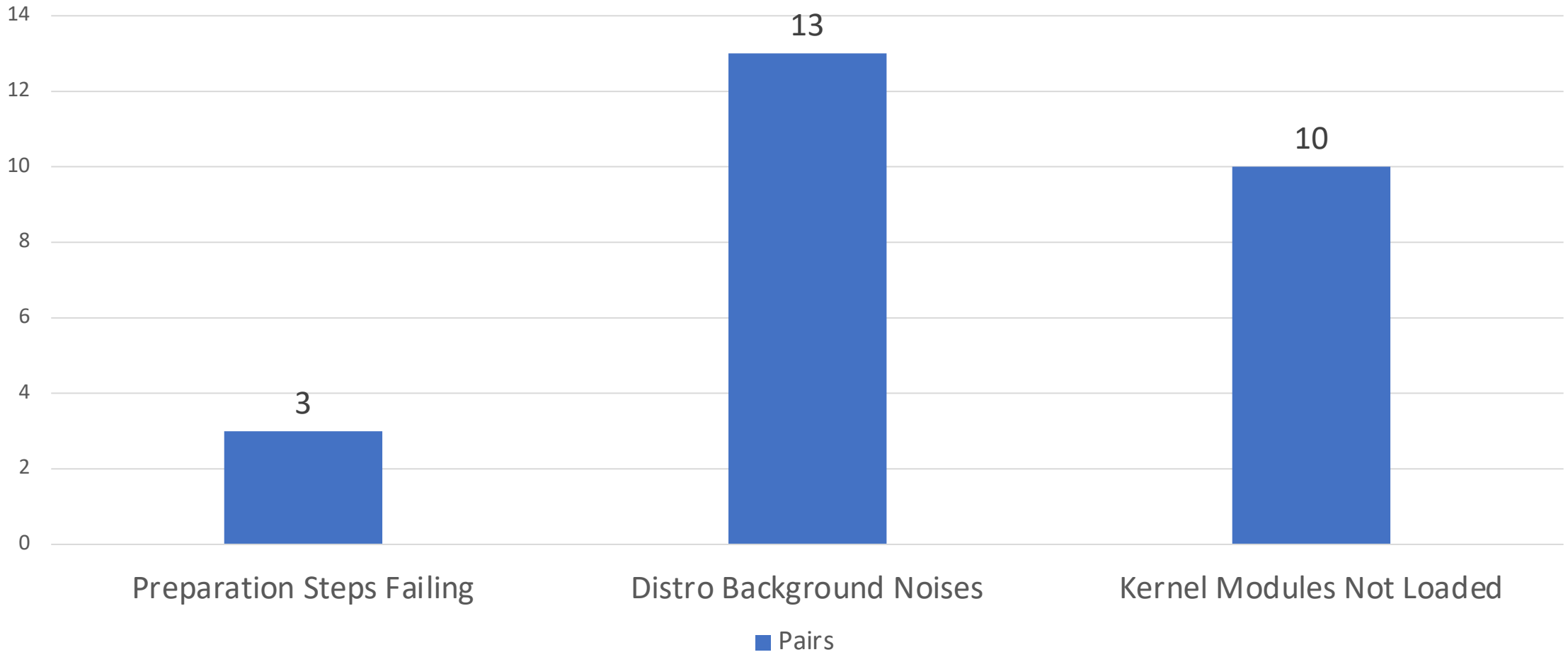
UC RIVERSIDE

Exploratory Experiment Results



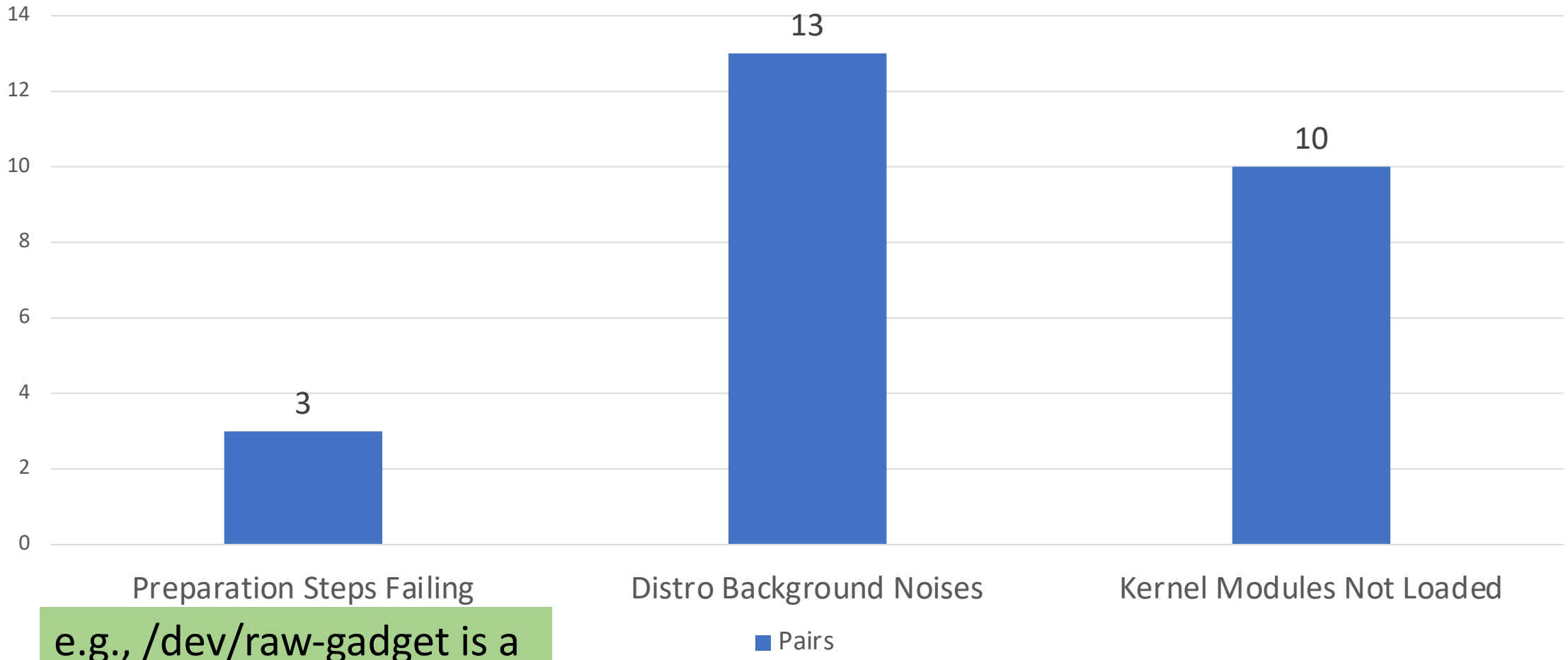
UC RIVERSIDE

Root Cause – Environment Change



UC RIVERSIDE

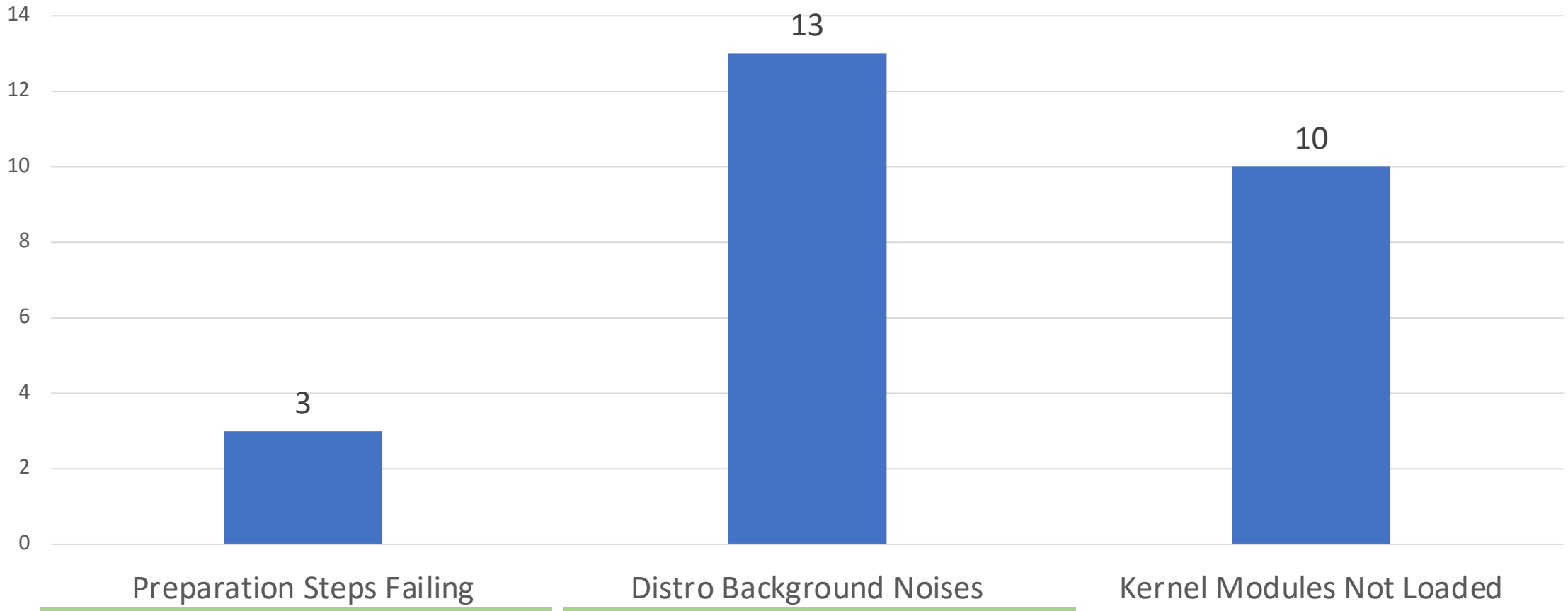
Root Cause – Environment Change



e.g., /dev/raw-gadget is a debug feature, does not exist in production kernel

UC RIVERSIDE

Root Cause – Environment Change

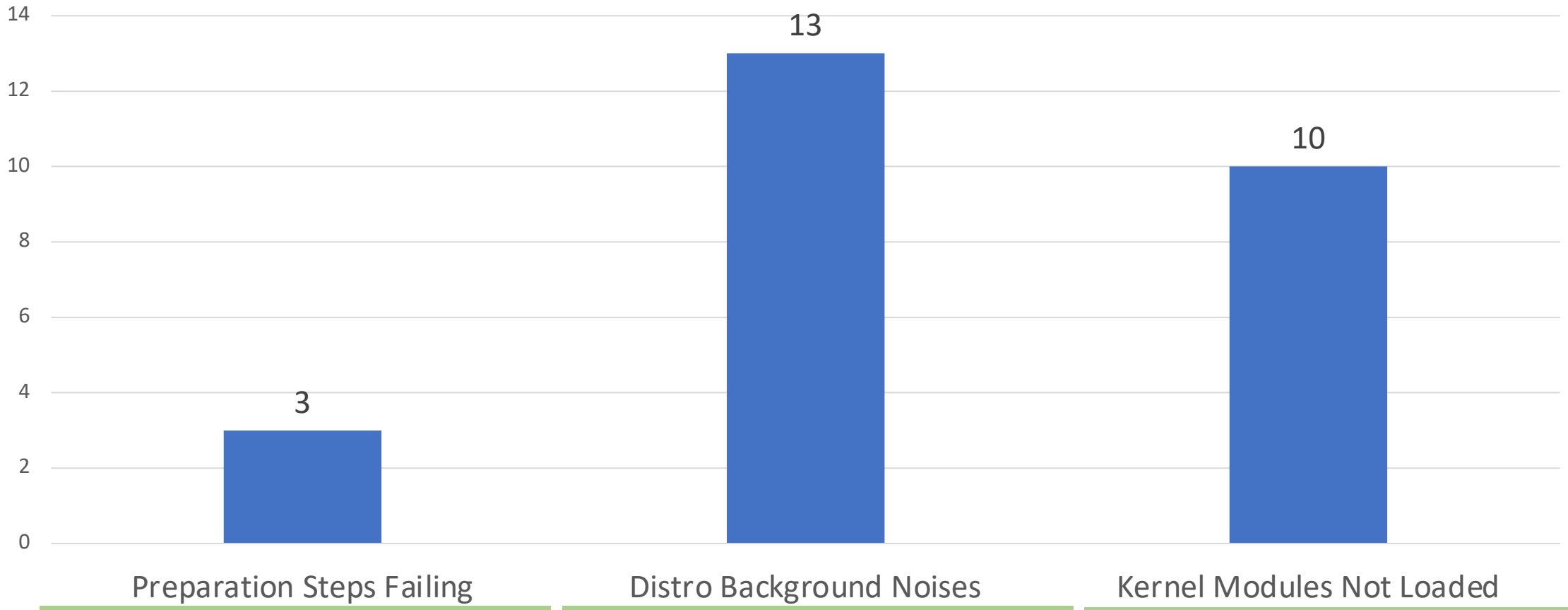


e.g., /dev/raw-gadget is a debug feature, does not exist in production kernel

e.g., Race-condition bugs require more resources (time and cores) to trigger

UC RIVERSIDE

Root Cause – Environment Change



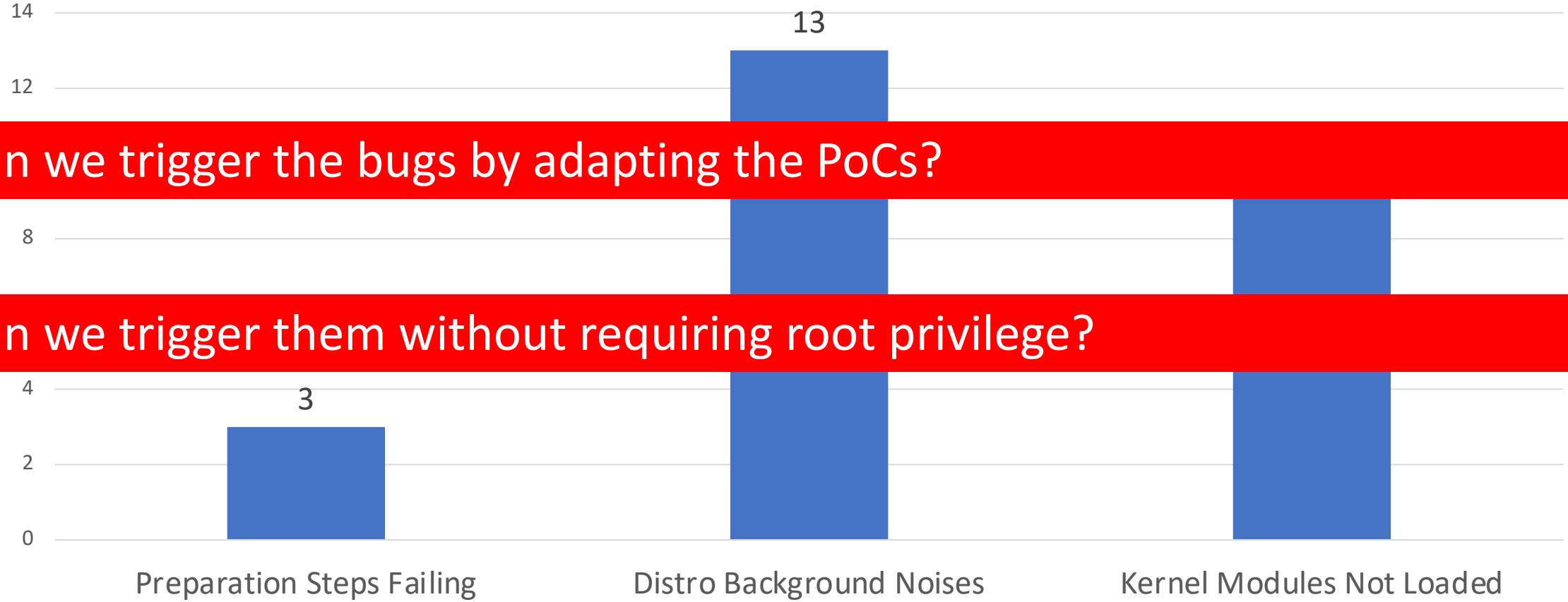
e.g., /dev/raw-gadget is a debug feature, does not exist in production kernel

e.g., Race-condition bugs require more resources (time and cores) to trigger

Kernel modules not loaded

UC RIVERSIDE

Root Cause – Environment Change



Can we trigger the bugs by adapting the PoCs?

Can we trigger them without requiring root privilege?

e.g., /dev/raw-gadget is a debug feature, does not exist in production kernel

e.g., Race-condition bugs require more resources (time and cores) to trigger

Kernel modules not loaded

UC RIVERSIDE

PoC Adaptation

PoC Adaptation

Preparation Steps Failing

- Rule out unnecessary preparation steps

PoC Adaptation

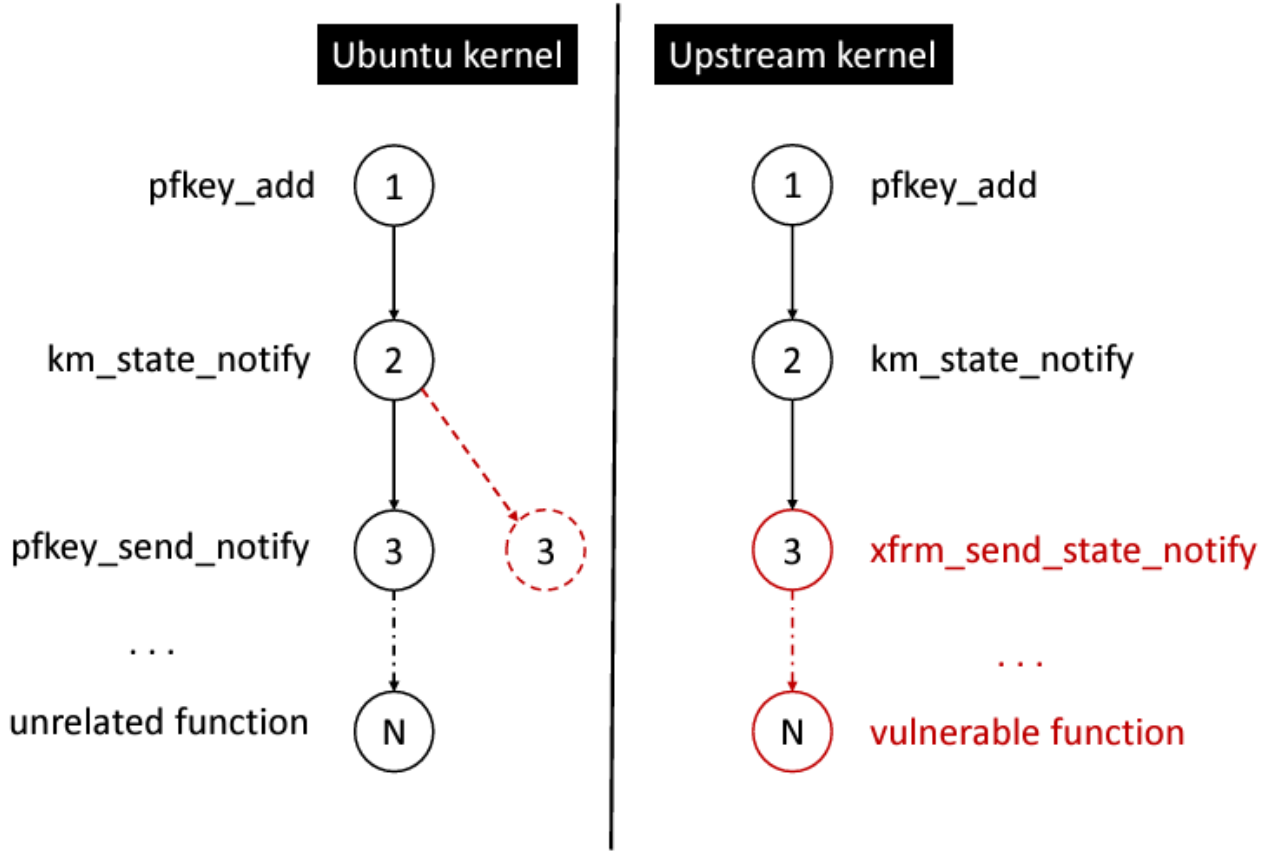
Preparation Steps Failing

- Rule out unnecessary preparation steps

Distro Background Noises

- Force to free occupied resources and enable “for loop” in PoC for data collision

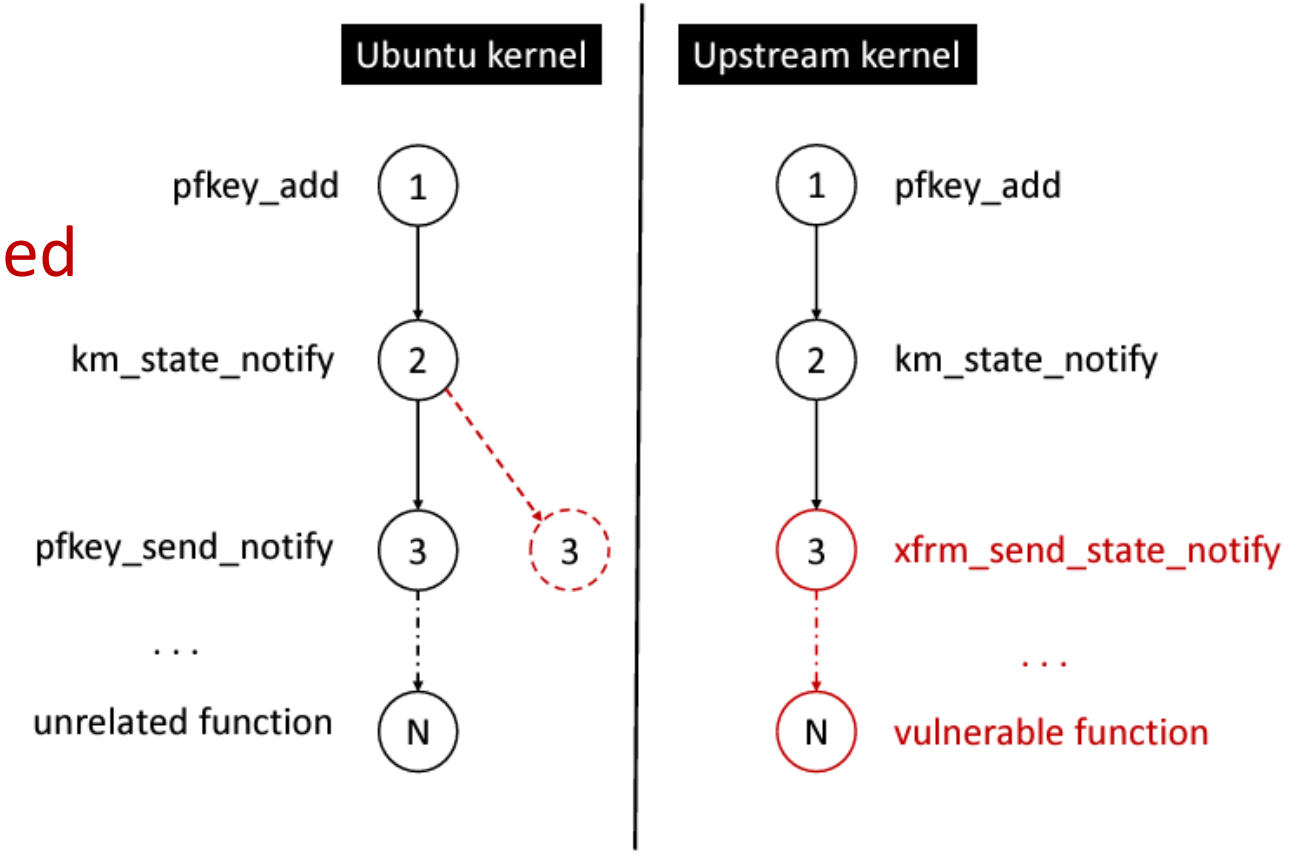
PoC Adaptation



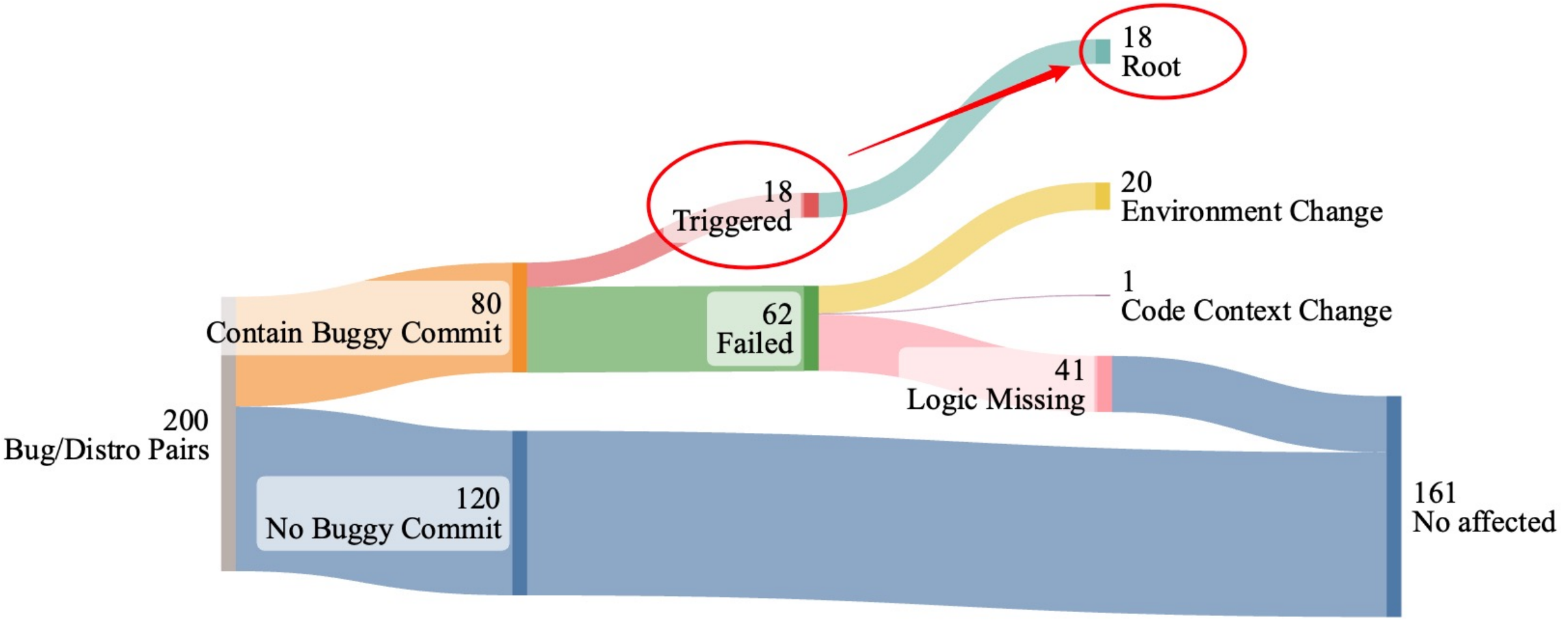
PoC Adaptation

Necessary Kernel Modules Not Loaded

- Using kernel trace to identify the missing kernel modules
- Load the missing modules by *modprobe*



Privilege Downgrading



UC RIVERSIDE

Privilege Downgrading

Privilege Downgrading

Kernel Internal Privilege Check

- *kernel capability check*

Privilege Downgrading

Kernel Internal Privilege Check

- *kernel capability check*

1. Monitoring ns_capable function

2. Force PoC using namespace

Privilege Downgrading

Kernel Internal Privilege Check

- *kernel capability check*

Kernel Privileged Operation

- *Releasing occupied resources*
- *Mounting filesystem*
- *Using modprobe to load kernel modules*

1. Monitoring ns_capable function

2. Force PoC using namespace

Privilege Downgrading

Kernel Internal Privilege Check

- *kernel capability check*

1. Monitoring ns_capable function

2. Force PoC using namespace

Kernel Privileged Operation

- ~~*Releasing occupied resources*~~
- *Mounting filesystem*
- *Using modprobe to load kernel modules*

Privilege Downgrading

Kernel Internal Privilege Check

- *kernel capability check*

1. Monitoring ns_capable function

2. Force PoC using namespace

Kernel Privileged Operation

- ~~*Releasing occupied resources*~~
- ~~*Mounting filesystem*~~
- *Using modprobe to load kernel modules*

Privilege Downgrading

Kernel Internal Privilege Check

- *kernel capability check*

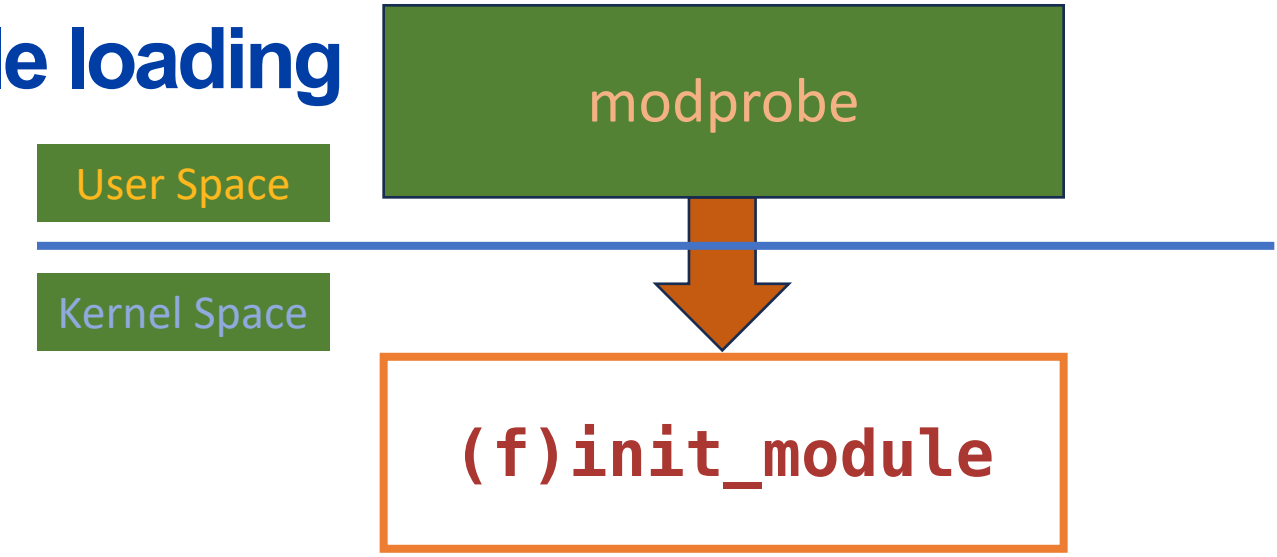
1. Monitoring ns_capable function
2. Force PoC using namespace

Kernel Privileged Operation

- ~~*Releasing occupied resources*~~
- ~~*Mounting filesystem*~~
- *Using modprobe to load kernel modules*

Is modprobe the only way to load kernel modules?

Privilege Downgrading – module loading



Kernel Privileged Operation

- ~~*Releasing occupied resources*~~
- ~~*Mounting filesystem*~~
- *Using modprobe to load kernel modules*

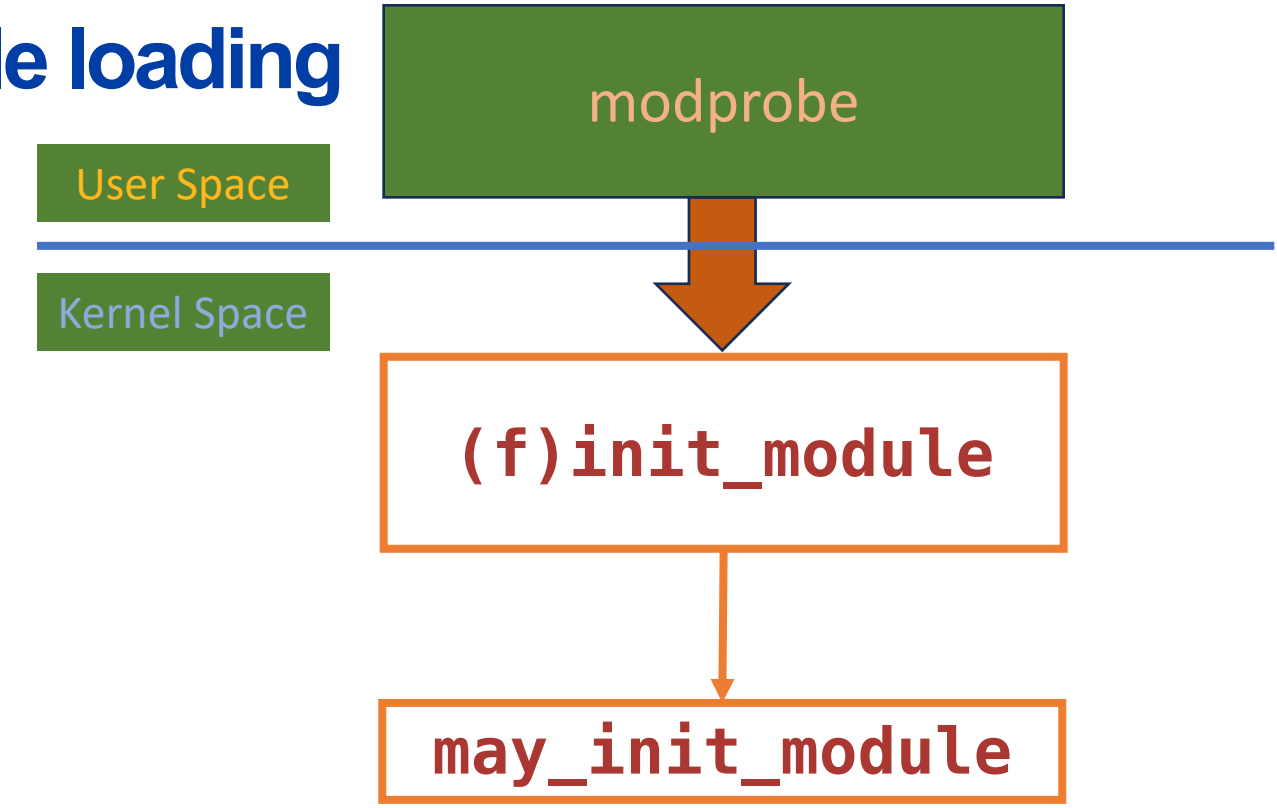
Is modprobe the only way to load kernel modules?

Privilege Downgrading – module loading

Kernel Privileged Operation

- ~~*Releasing occupied resources*~~
- ~~*Mounting filesystem*~~
- *Using modprobe to load kernel modules*

Is modprobe the only way to load kernel modules?

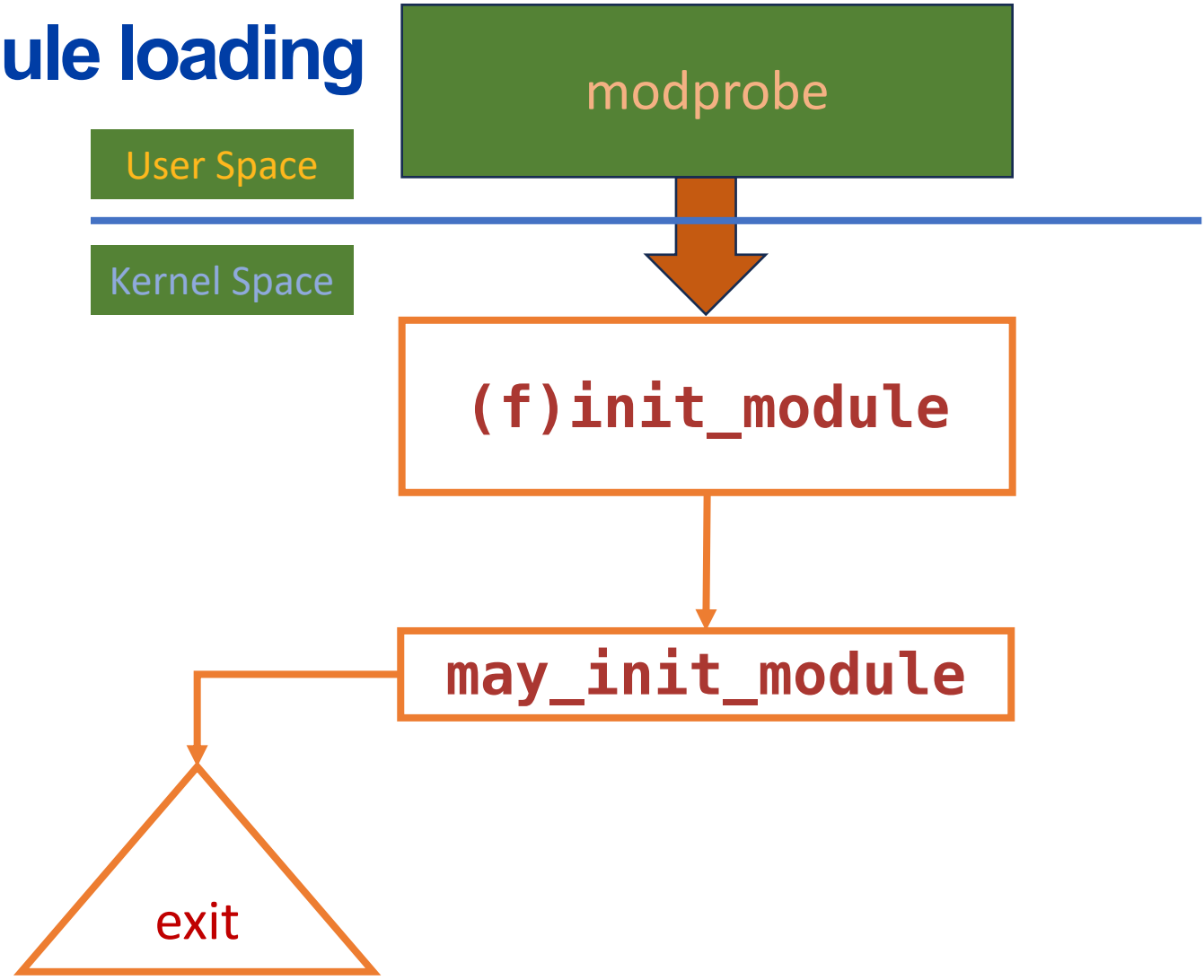


Privilege Downgrading – module loading

Kernel Privileged Operation

- ~~*Releasing occupied resources*~~
- ~~*Mounting filesystem*~~
- *Using modprobe to load kernel modules*

Is modprobe the only way to load kernel modules?

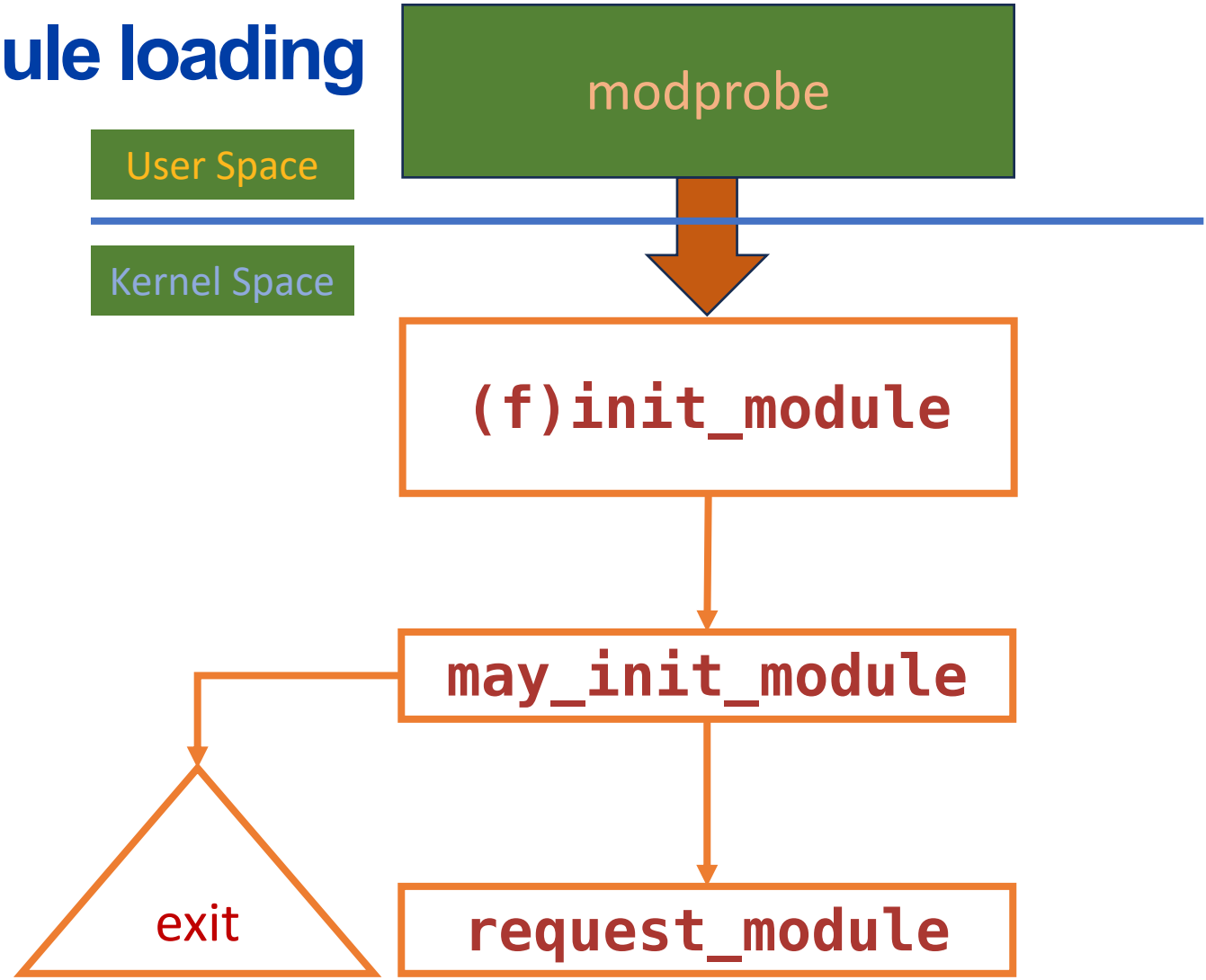


Privilege Downgrading – module loading

Kernel Privileged Operation

- ~~*Releasing occupied resources*~~
- ~~*Mounting filesystem*~~
- *Using modprobe to load kernel modules*

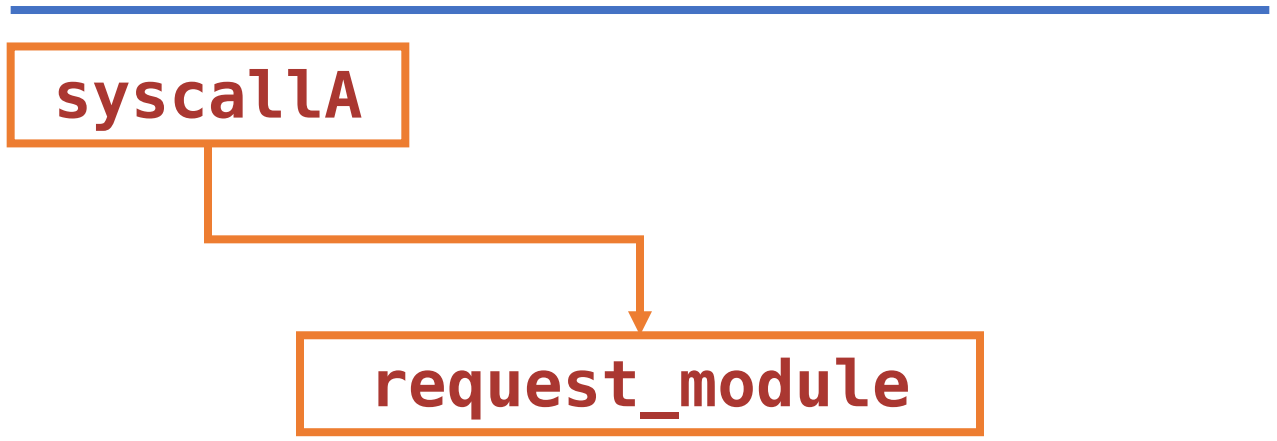
Is modprobe the only way to load kernel modules?



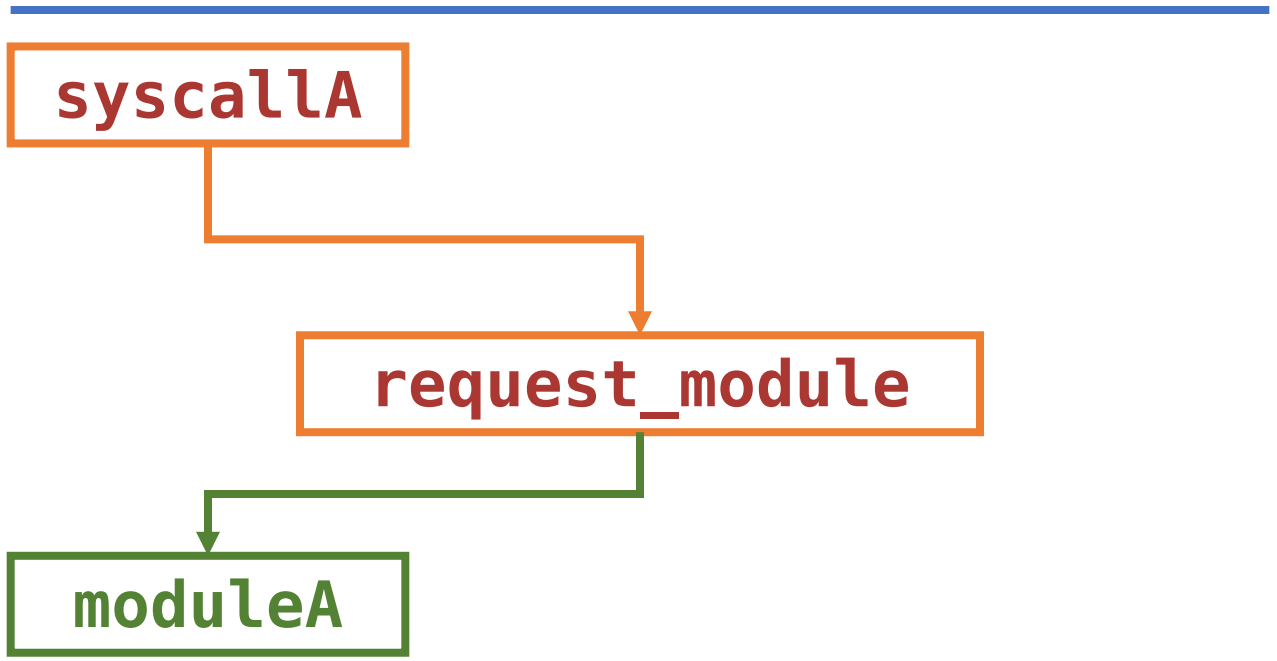
Privilege Downgrading – request_module

```
request_module
```

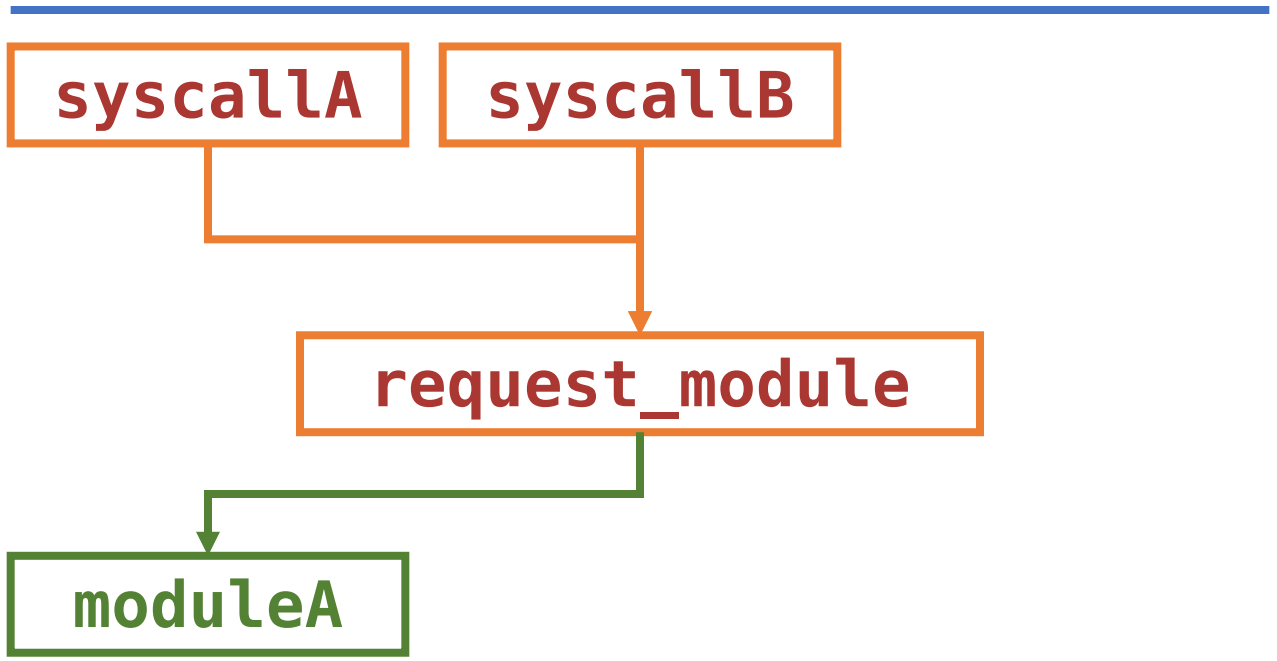
Privilege Downgrading – request_module



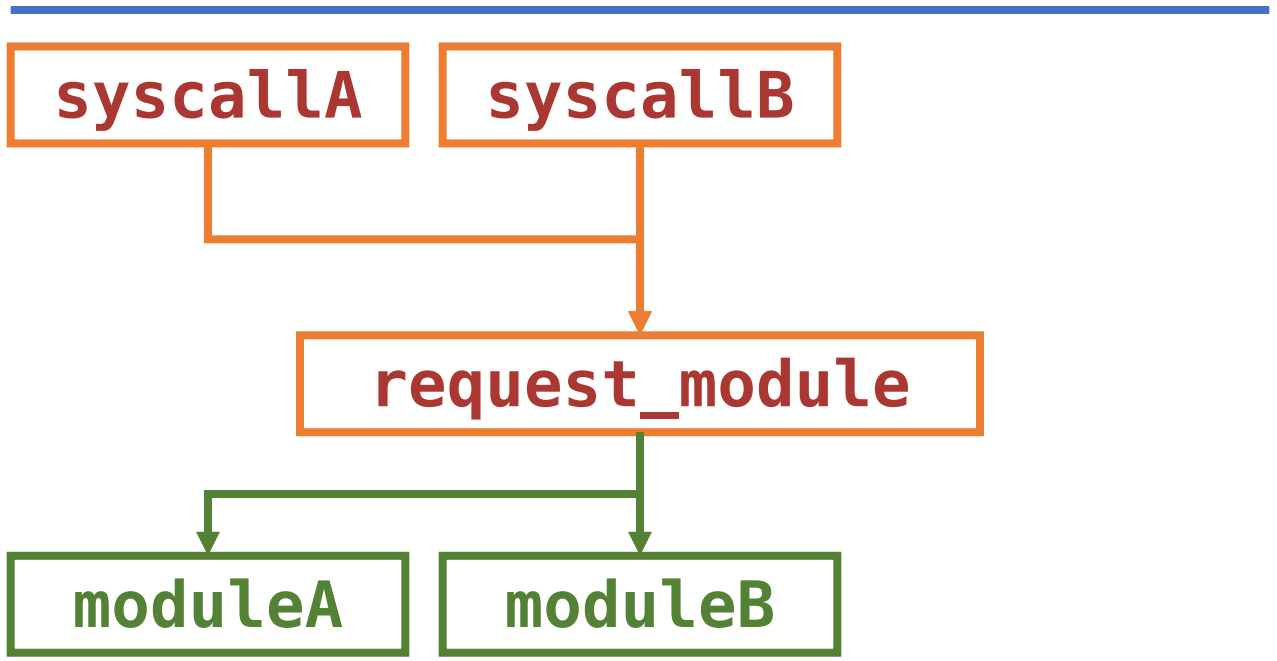
Privilege Downgrading – request_module



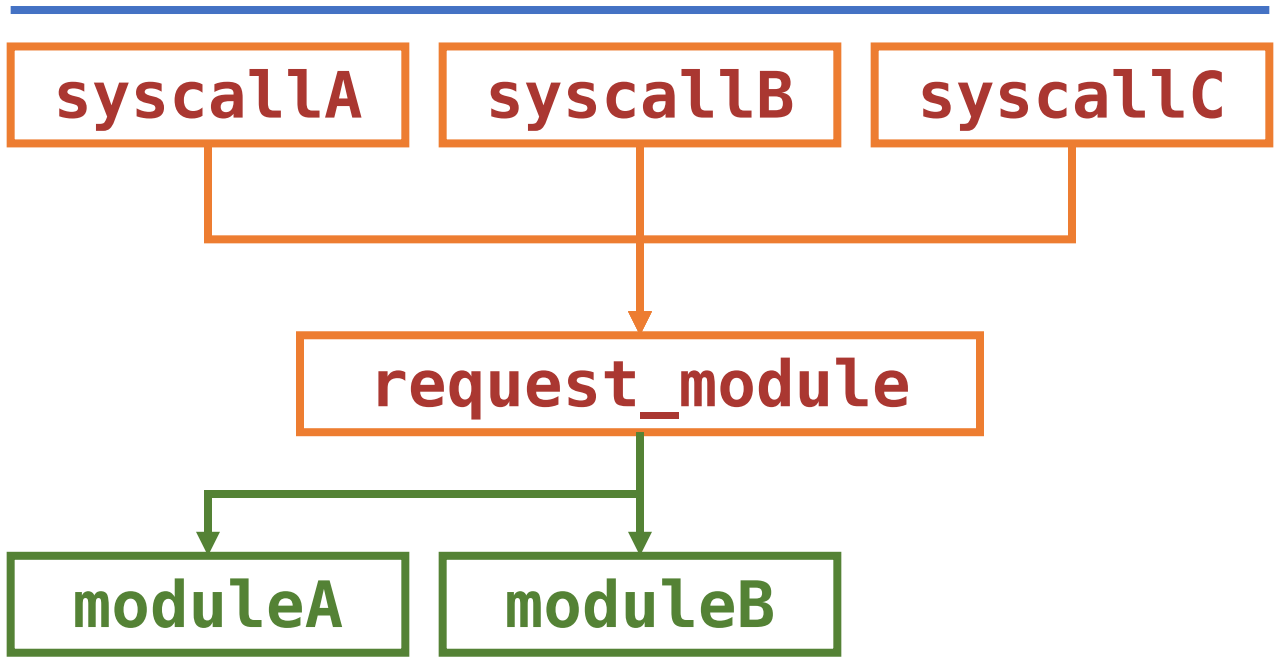
Privilege Downgrading – request_module



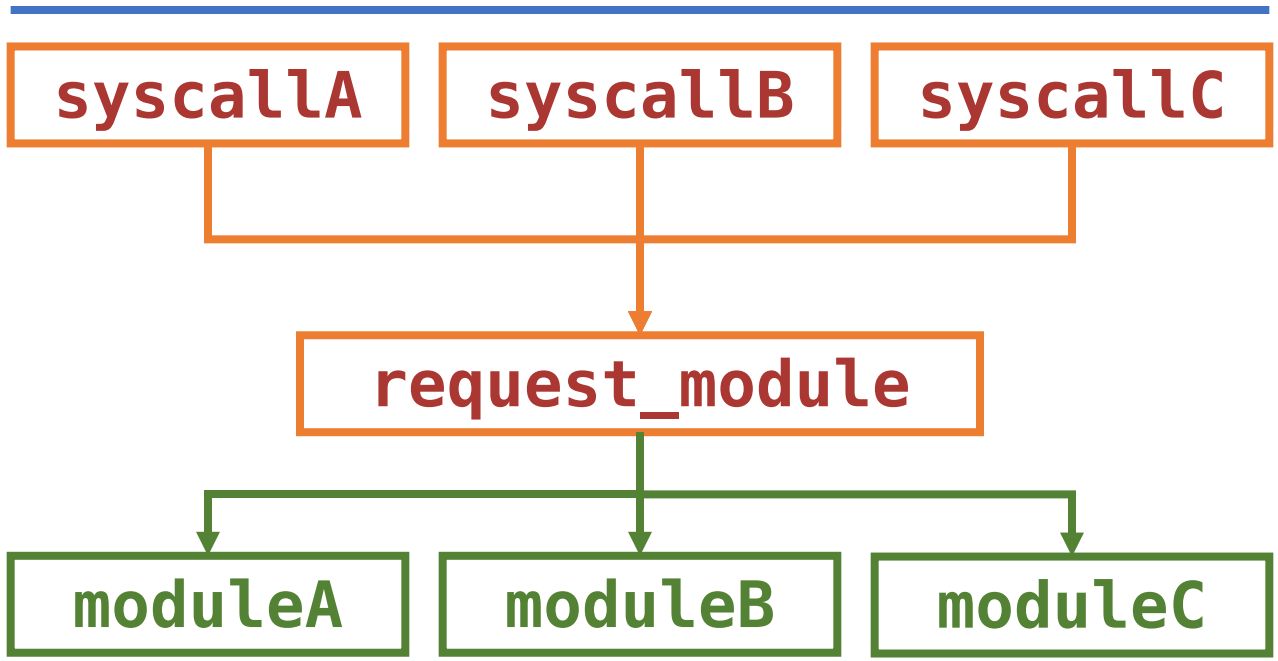
Privilege Downgrading – request_module



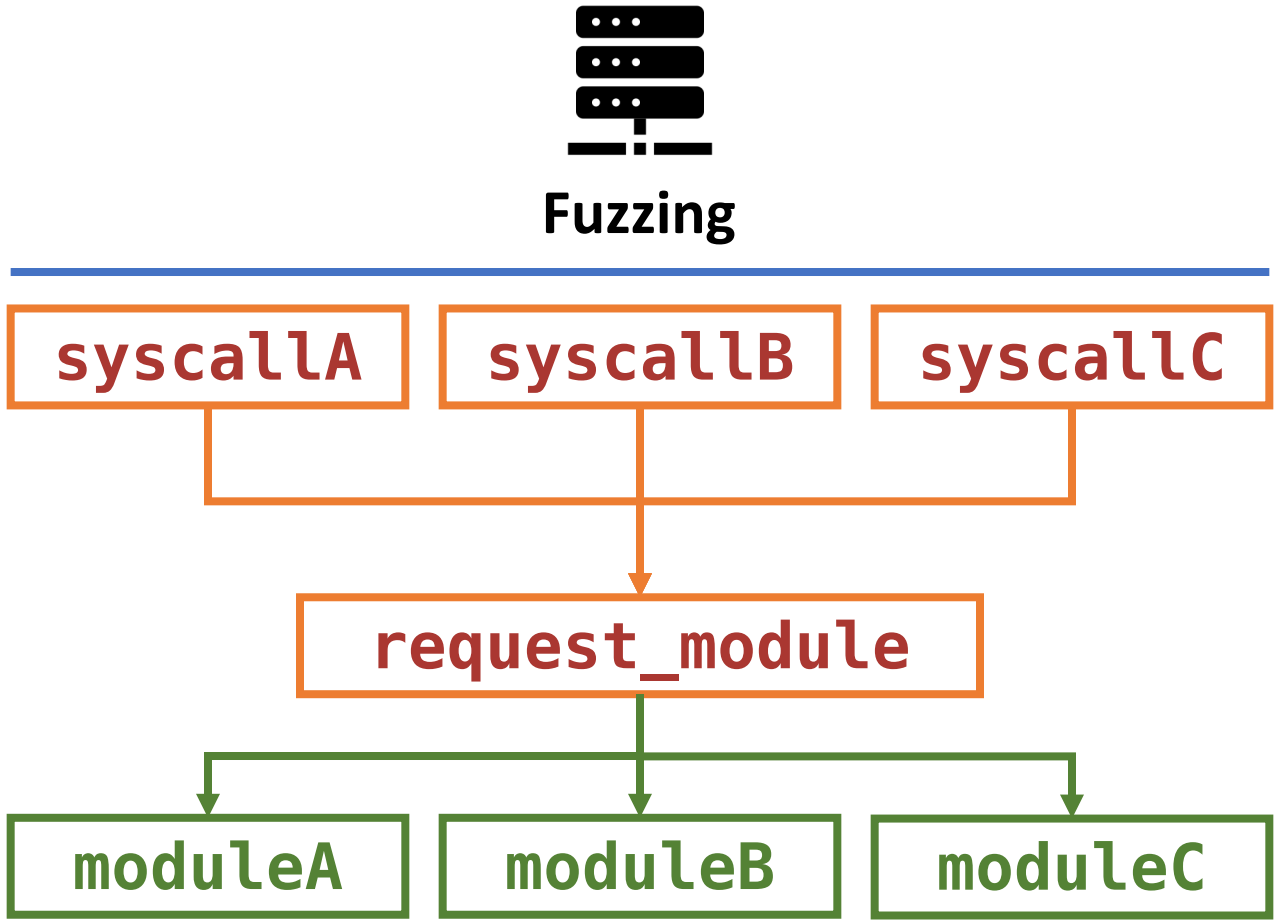
Privilege Downgrading – request_module



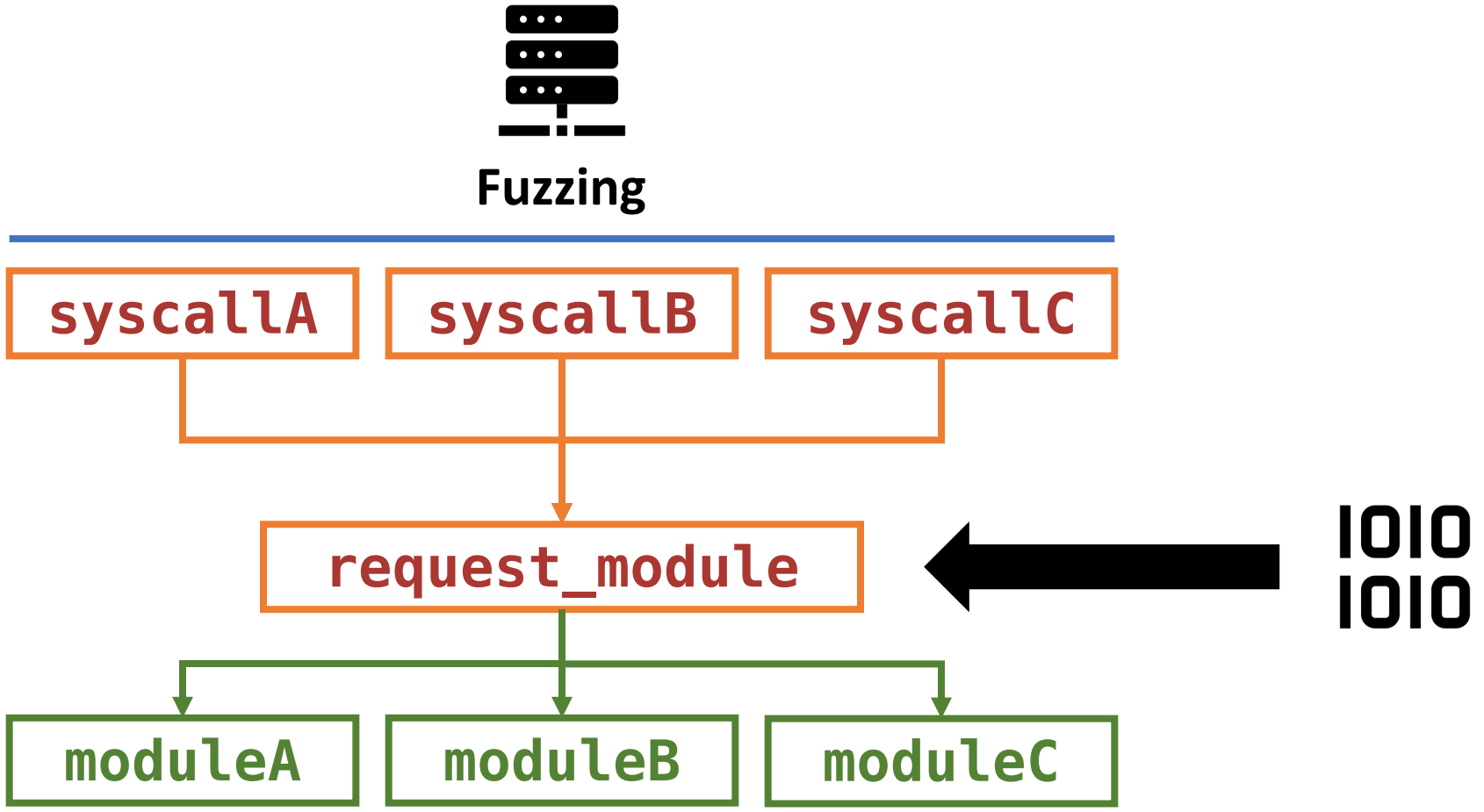
Privilege Downgrading – request_module



Privilege Downgrading – request_module



Privilege Downgrading – request_module



Privilege Downgrading – request_module

Table III: Breakdown of module loading fuzzing results

Distro	crypto	fs	net	tty	other	Sum	Total
Ubuntu	85	58	157	13	3	316	371
Fedora	69	47	106	11	3	236	272
Debian	97	52	132	11	7	299	361
Suse	99	52	145	15	0	311	465

Privilege Downgrading – request_module

Table III: Breakdown of module loading fuzzing results

Distro	crypto	fs	net	tty	other	Sum	Total
Ubuntu	85	58	157	13	3	316	371
Fedora	69	47	106	11	3	236	272
Debian	97	52	132	11	7	299	361
Suse	99	52	145	15	0	311	465

```
modprobe xfrm_user
```

Root

Privilege Downgrading – request_module

Table III: Breakdown of module loading fuzzing results

Distro	crypto	fs	net	tty	other	Sum	Total
Ubuntu	85	58	157	13	3	316	371
Fedora	69	47	106	11	3	236	272
Debian	97	52	132	11	7	299	361
Suse	99	52	145	15	0	311	465

```
modprobe xfrm_user
```

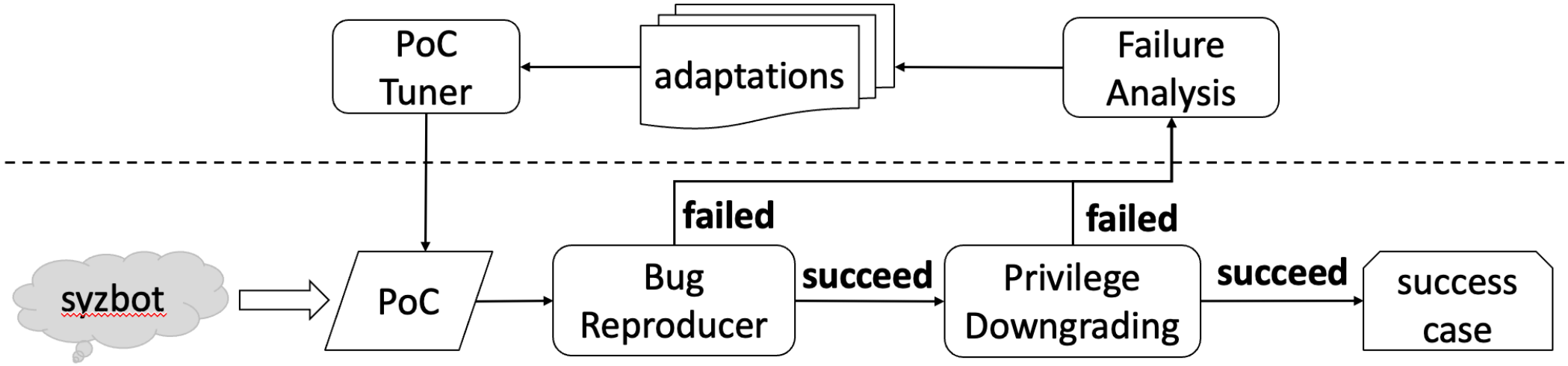
Root

Normal User

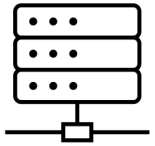
```
res = syscall(__NR_socket, AF_NETLINK, SOCK_RAW, NETLINK_XFRM);
```

UC RIVERSIDE

SyzBridge


PoC
Tuner
adaptations
Failure
Analysis
failed
failed
syzbot
PoC
Bug
Reproducer
succeed
Privilege
Downgrading
succeed
success
case

Evaluation – Upstream PoC Adaptations



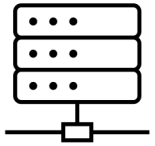
43 distro major releases

230 upstream KASAN bugs

600 seconds timeout

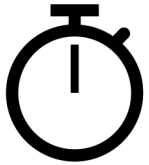
UC RIVERSIDE

Evaluation – Upstream PoC Adaptations



43 distro major releases

230 upstream KASAN bugs

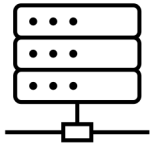


600 seconds timeout

	Root-Triggerable	Normal-User-Triggerable
Ubuntu	55	2
Fedora	58	3
Debian	33	2
Suse	29	1

UC RIVERSIDE

Evaluation – Upstream PoC Adaptations



43 distro major releases

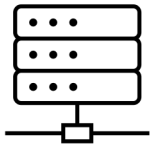
230 upstream KASAN bugs

600 seconds timeout

	Root-Triggerable			Normal-User-Triggerable		
Ubuntu	55	→	86	2	→	27
Fedora	58	→	77	3	→	46
Debian	33	→	63	2	→	21
Suse	29	→	57	1	→	18

UC RIVERSIDE

Evaluation – Upstream PoC Adaptations



43 distro major releases

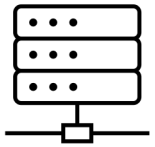
230 upstream KASAN bugs

600 seconds timeout

	Root-Triggerable				Normal-User-Triggerable			
Ubuntu	55	→	86		2	→	27	
Fedora	58	→	77		3	→	46	
Debian	33	→	63	61%	2	→	21	1300%
Suse	29	→	57		1	→	18	

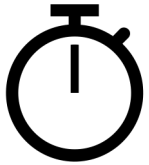
UC RIVERSIDE

Evaluation – Exploitability Assessment Pipeline.



68 distro major releases

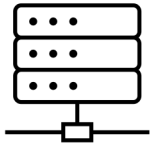
282 upstream high-risk bugs



600 seconds timeout
5 hours SyzScope

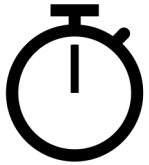
UC RIVERSIDE

Evaluation – Exploitability Assessment Pipeline.



68 distro major releases

282 upstream high-risk bugs

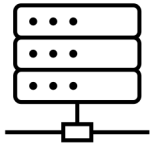


600 seconds timeout
5 hours SyzScope

	Root-Triggerable	Normal-User-Triggerable
Ubuntu	54	1
Fedora	48	3
Debian	47	3
Suse	27	1

UC RIVERSIDE

Evaluation – Exploitability Assessment Pipeline.



68 distro major releases

282 upstream high-risk bugs

600 seconds timeout
5 hours SyzScope

	Root-Triggerable			Normal-User-Triggerable		
Ubuntu	54	→	82	1	→	35
Fedora	48	→	82	3	→	50
Debian	47	→	84	3	→	31
Suse	27	→	42	1	→	9

UC RIVERSIDE

Evaluation – Exploitability Assessment Pipeline.

Table VI: High-risk bugs from experiment II

Bug	Bug Primitive	Affect	Before Adaptation		Environment Adaptation			Privilege Adaptation		After Adaptation	
			root	normal	EA1	EA2	EA3	PA1	PA2	root	normal
CVE-2022-27666*	OOB W	UFD	F	-	-	UD	UD	UD	UFD	-	UFD
CVE-2022-0185	OOB W	UFD	UFD	-	-	-	-	-	UFD	-	UFD
CVE-2021-22600	DF	UFD	UFD	-	-	-	-	-	UFD	-	UFD
CVE-2021-22555	OOB W	UFD	F	-	-	UD	UD	UD	UFD	-	UFD
CVE-2021-4154	CFH	UFD	UFD	-	-	-	-	-	UFD	-	UFD
CVE-2021-3715	UAF W	U	U	-	-	-	-	-	U	-	U
cf7393b*	UAF W	UFD	UFD	-	-	-	-	-	UF	D	UF
4b0830a	UAF W	UFD	U	D	-	-	F	-	F	U	FD
e67f2fc	UAF W	UFD	-	-	-	UFD	-	-	UFD	-	UFD
2389bfc*	CFH	UFD	UFD	-	-	-	-	-	F	UD	F
403eb21	CFH	UFS	UF	-	S	-	-	-	UF	S	UF
f4c90f2*	OOB W	UFDS	UFDS	-	-	-	-	-	UFS	D	UFS
380acd1*	DF	UF	UF	-	-	-	-	-	UF	-	UF
b53aed2	OOB W	UFD	UFD	-	-	-	-	-	UFD	-	UFD
e2d0f38	CFH	F	-	-	-	F	-	-	F	-	F
d35e6e8	NPD W	UFD	UFD	-	-	-	-	-	UFD	-	UFD
60e3243	CVW	UFDS	UFDS	-	-	-	-	-	UFDS	-	UFDS
a53b68e	OOB W	UF	F	-	-	U	U	U	UF	-	UF
5ad0e07	NPD W	UF	UF	-	-	-	-	-	UF	-	UF

See the complete table in our paper

UC RIVERSIDE

Evaluation – Exploitability Assessment Pipeline.

Table VI: High-risk bugs from experiment II

Bug	Bug Primitive	Affect	Before Adaptation		Environment Adaptation			Privilege Adaptation		After Adaptation	
			root	normal	EA1	EA2	EA3	PA1	PA2	root	normal
CVE-2022-27666*	OOB W	UFD	F	-	-	UD	UD	UD	UFD	-	UFD
CVE-2022-0185	OOB W	UFD	UFD	-	-	-	-	-	UFD	-	UFD
CVE-2021-22600	DF	UFD	UFD	-	-	-	-	-	UFD	-	UFD
CVE-2021-22555	OOB W	UFD	F	-	-	UD	UD	UD	UFD	-	UFD
CVE-2021-4154	CFH	UFD	UFD	-	-	-	-	-	UFD	-	UFD
CVE-2021-3715	UAF W	U	U	-	-	-	-	-	U	-	U
cf7393b*	UAF W	UFD	UFD	-	-	-	-	-	UF	D	UF
4b0830a	UAF W	UFD	U	D	-	-	F	-	F	U	FD
e67f2fc	UAF W	UFD	-	-	-	UFD	-	-	UFD	-	UFD
2389bfc*	CFH	UFD	UFD	-	-	-	-	-	F	UD	F
403eb21	CFH	UFS	UF	-	S	-	-	-	UF	S	UF
f4c90f2*	OOB W	UFDS	UFDS	-	-	-	-	-	UFS	D	UFS
380acd1*	DF	UF	UF	-	-	-	-	-	UF	-	UF
b53aed2	OOB W	UFD	UFD	-	-	-	-	-	UFD	-	UFD
e2d0f38	CFH	F	-	-	-	F	-	-	F	-	F
d35e6e8	NPD W	UFD	UFD	-	-	-	-	-	UFD	-	UFD
60e3243	CVW	UFDS	UFDS	-	-	-	-	-	UFDS	-	UFDS
a53b68e	OOB W	UF	F	-	-	U	U	U	UF	-	UF
5ad0e07	NPD W	UF	UF	-	-	-	-	-	UF	-	UF

See the complete table in our paper

UC RIVERSIDE

Takeaway

Takeaway

Upstream PoCs often fail on downstream OS, but it does not imply the downstream are immune from upstream bugs

Takeaway

Upstream PoCs often fail on downstream OS, but it does not imply the downstream are immune from upstream bugs

Upstream PoCs can be transformed to downstream-valid PoCs through varied adaptations

Takeaway

Upstream PoCs often fail on downstream OS, but it does not imply the downstream are immune from upstream bugs

Upstream PoCs can be transformed to downstream-valid PoCs through varied adaptations

A seemingly high-privilege PoCs might be downgraded to low privilege

Takeaway

Upstream PoCs often fail on downstream OS, but it does not imply the downstream are immune from upstream bugs

Upstream PoCs can be transformed to downstream-valid PoCs through varied adaptations

A seemingly high-privilege PoCs might be downgraded to low privilege

SyzBridge found >50 syzbot bugs are likely exploitable on at least one downstream distro we tested (whereas only 5 were recognized with CVEs in the past 4 years)

Q&A

Thank you for listening

Access my portfolio

SyzBridge has been open source

UC RIVERSIDE