CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

RISCover

Automatic Discovery of User-exploitable Architectural Security Vulnerabilities in Closed-Source RISC-V CPUs

Fabian Thomas, Eric García Arribas, **Lorenz Hetterich**, Ruiyi Zhang, Daniel Weber, Lukas Gerlach, Michael Schwarz

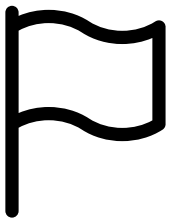
- Free & open-source ISA

- Free & open-source ISA
- Multiple silicon CPUs

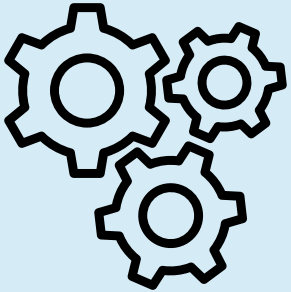
- Free & open-source ISA
- Multiple silicon CPUs
- T-Head: More than 4 billion sold

- Free & open-source ISA
- Multiple silicon CPUs
- T-Head: More than 4 billion sold
- But do they have bugs?

Research Question



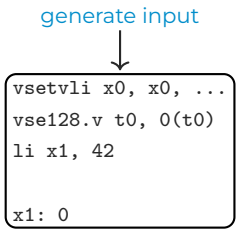
Can we automatically find security vulnerabilities in silicon RISC-V CPUs without source code and without golden model?



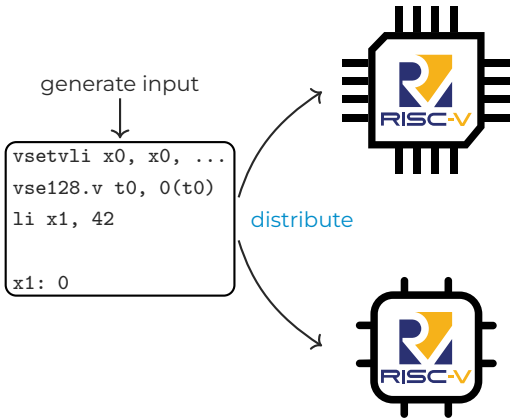
RISCover

Finding Bugs in RISC-V CPUs

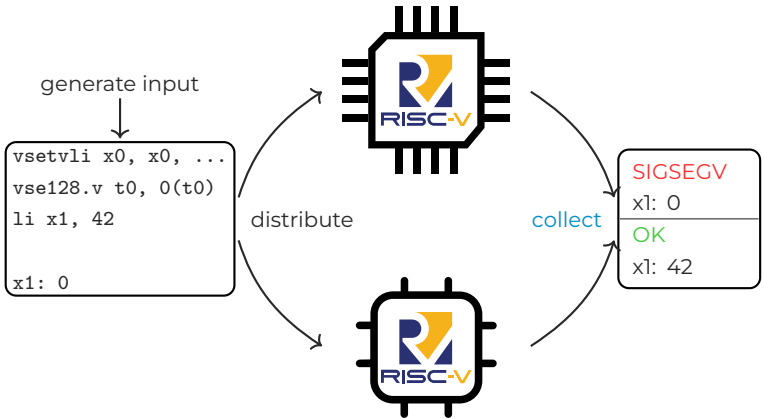
RISCover: Differential CPU Fuzzing



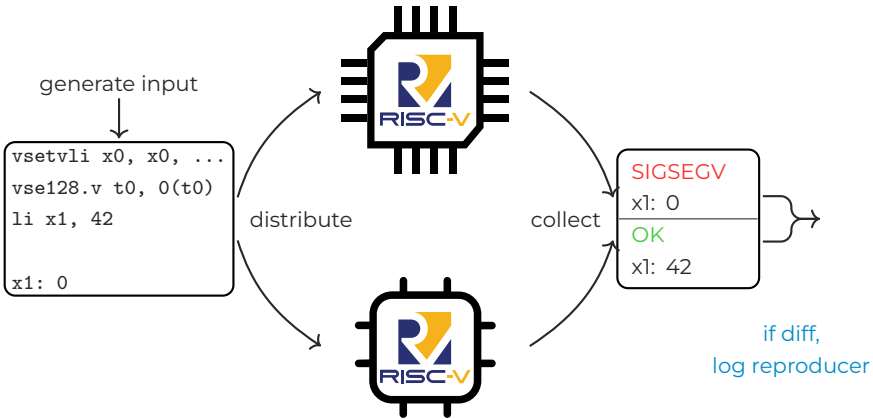
RISCover: Differential CPU Fuzzing



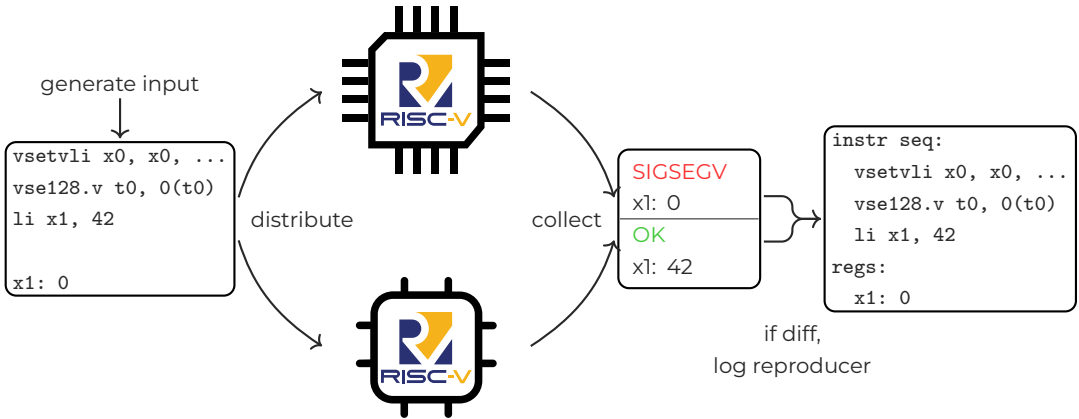
RISCover: Differential CPU Fuzzing



RISCover: Differential CPU Fuzzing

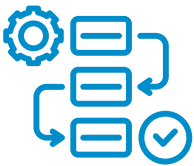


RISCover: Differential CPU Fuzzing



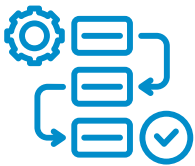
Challenges

Sequence Generation



Challenges

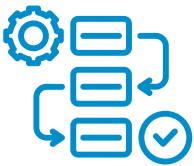
Sequence Generation



→ prefer rare instructions

Challenges

Sequence Generation

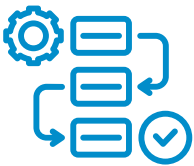


→ prefer rare instructions

Non-determinism

Challenges

Sequence Generation



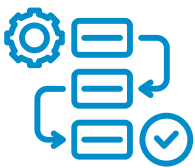
→ prefer rare instructions

Non-determinism

→ exclude instructions

Challenges

Sequence Generation

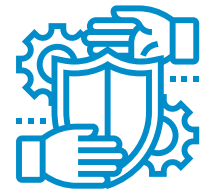


→ prefer rare instructions

Non-determinism

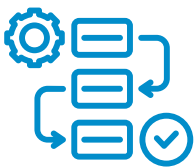
→ exclude instructions

Framework Integrity



Challenges

Sequence Generation

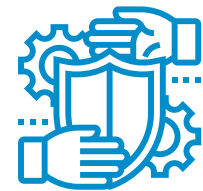


→ prefer rare instructions

Non-determinism

→ exclude instructions

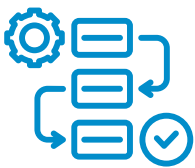
Framework Integrity



→ sandbox

Challenges

Sequence Generation

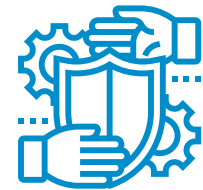


→ prefer rare instructions

Non-determinism

→ exclude instructions

Framework Integrity



→ sandbox
→ handle exceptions

RISCover: 8 Target Silicon CPUs

T-HEAD

C906, C908,
C910, C920

RISCover: 8 Target Silicon CPUs

C906, C908, C910, C920

U54, U74, P550

RISCover: 8 Target Silicon CPUs

C906, C908, C910, C920

U54, U74, P550

X60

RISCover: Findings

Vulnerability	CPUs	Time-to-Bug
GhostWrite	C910, C920	<1 s
Denial-of-service	C906	<2 min
Denial-of-service	C908	<15 min
Denial-of-service	X60	<50 min
Wrong Misaligned Zero-Stores	BeagleV Fire (U54)	<5 s
User-space Hangs	C906, C908, C910, X60	<5 s
Undocumented instructions	P550	<1 s
Decoder Faults	C908, C910	<1 s
ISA Incompatibilities	C908, C910	<2 min
QEMU Segfaults	QEMU 8.2.2, 9.0.0	<30 s

GhostWrite

Writing physical memory

- `vse128.v v0, 0(t0)`

- `vse128.v v0, 0(t0)`
- Writes to physical address

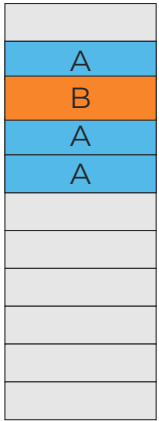
- `vse128.v v0, 0(t0)`
- Writes to physical address
- Deterministic & 100% reliable

- `vse128.v v0, 0(t0)`
- Writes to physical address
- Deterministic & 100% reliable
- Fully compromises system

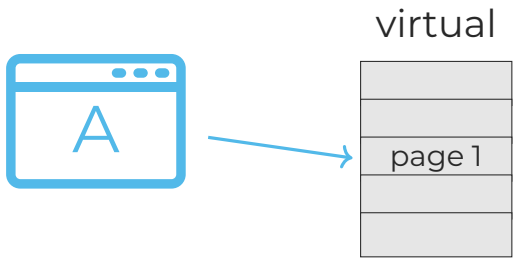
Memory Isolation

virtual

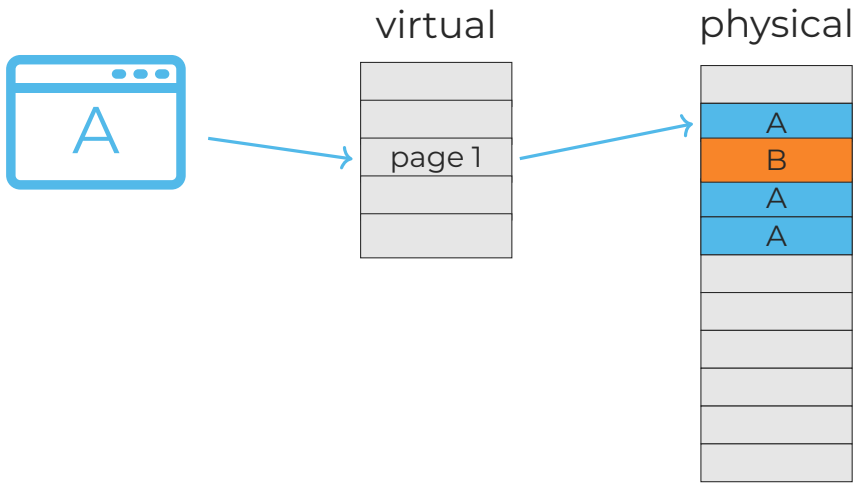
physical



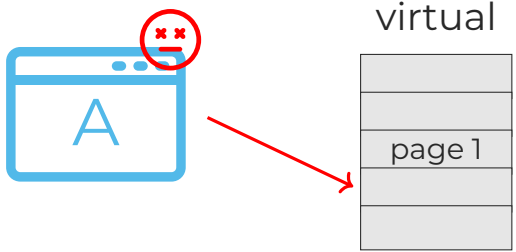
Memory Isolation



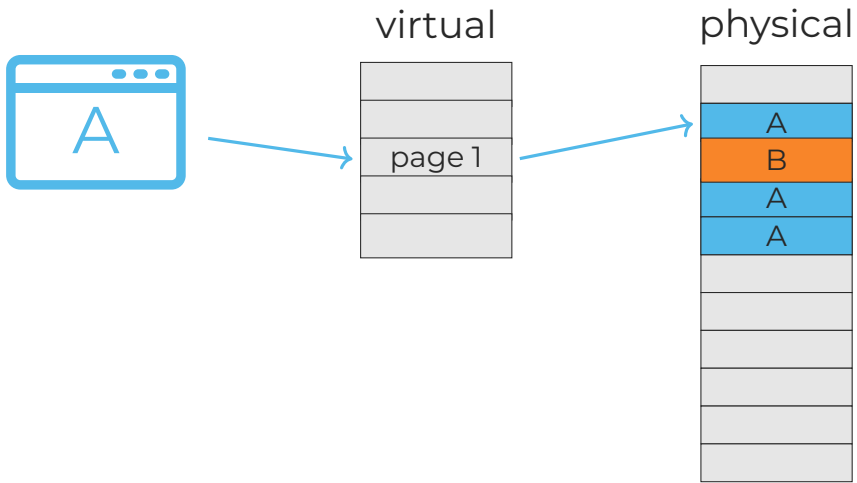
Memory Isolation



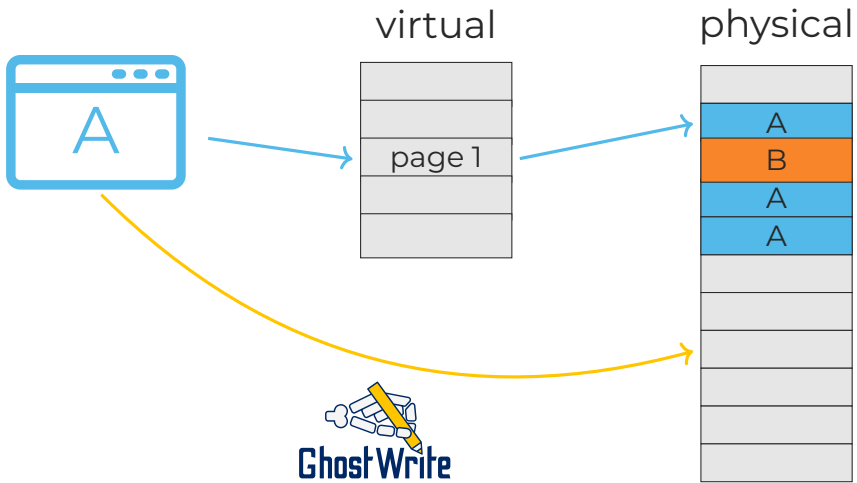
Memory Isolation



Memory Isolation



~~Memory Isolation~~: GhostWrite



GhostWrite: Getting root

```
if get_user() == root:
  run_as_root()
else:
  auth()
```


```
get_user:
  return current_user
```

OS

GhostWrite: Getting root

```
if get_user() == root:
  run_as_root()
else:
  auth()
```

syscall

```
get_user:
  return current_user
```

OS

GhostWrite: Getting root

```
if get_user() == root:
  run_as_root()
else:
  auth()
```

syscall

```
get_user:
  return current_user ← patch
```

OS

GhostWrite: Getting root

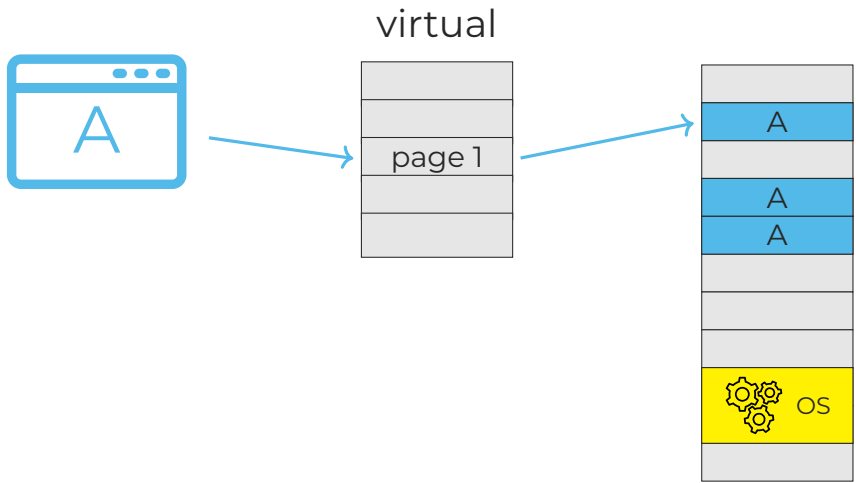
```
if get_user() == root:
  run_as_root()
else:
  auth()
```

syscall

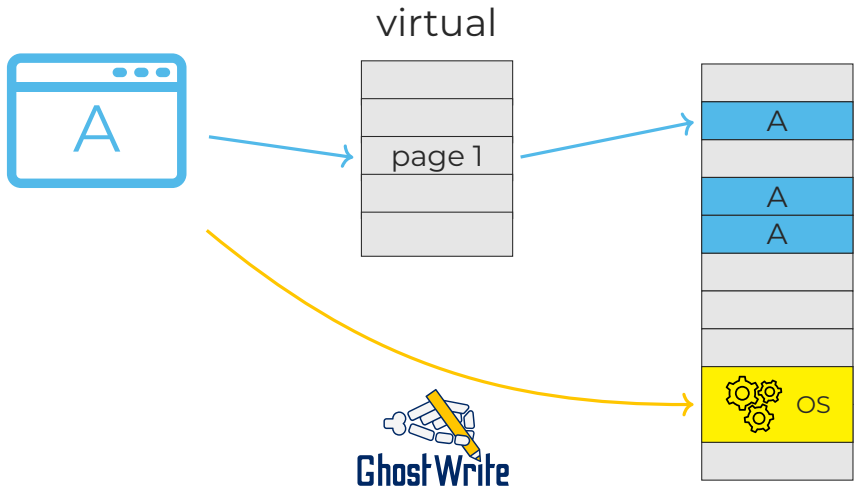

```
get_user:
  return root ← patch
```

OS

GhostWrite: Getting root

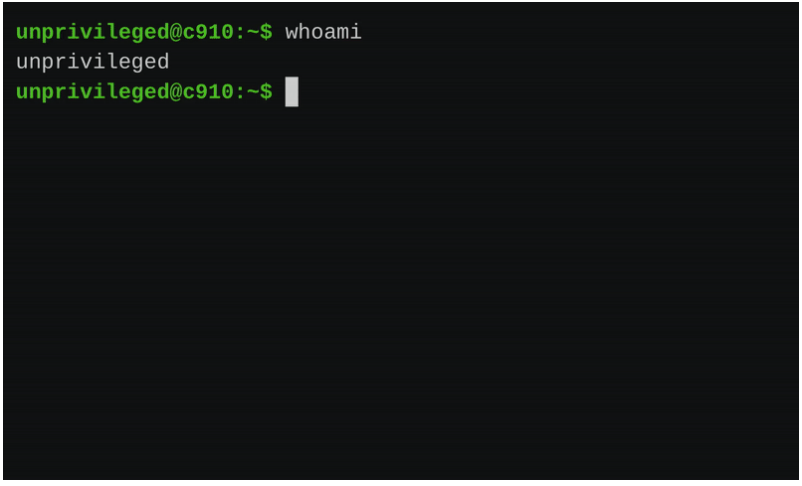


GhostWrite: Getting root



Demo: Getting root

Demo: Getting root



Demo: Getting root

```
unprivileged@c910:~$ whoami
unprivileged
unprivileged@c910:~$ ./sudo-exploit
```

Demo: Getting root

Demo: Getting root

Demo: Getting root

```
unprivileged@c910:~$ whoami
unprivileged
unprivileged@c910:~$ ./sudo-exploit
Current user: unprivileged
Patching getuid at 0x401a3c8f
..........

DONE!

Current user: root
Obtaining shell...
root@c910:~$
```

Demo: Getting root

```
unprivileged@c910:~$ whoami
unprivileged
unprivileged@c910:~$ ./sudo-exploit
Current user: unprivileged
Patching getuid at 0x401a3c8f
..........

DONE!

Current user: root
Obtaining shell...
root@c910:~$ whoami
```

Demo: Getting root

```
unprivileged@c910:~$ whoami
unprivileged
unprivileged@c910:~$ ./sudo-exploit
Current user: unprivileged
Patching getuid at 0x401a3c8f
..........

DONE!

Current user: root
Obtaining shell...
root@c910:~$ whoami
root
root@c910:~$
```

GhostWrite: Mitigation

- C910 has no microcode

GhostWrite: Mitigation

```
Vulnerabilities:
  Gather data sampling:      Not affected
  Ghostwrite:                Mitigation: xtheadvector disabled
  Indirect target selection: Not affected
  Itlb multihit:             Not affected
  L1tf:                      Not affected
  Mds:                       Not affected
  Meltdown:                  Not affected
  Mmio stale data:           Not affected
  Reg file data sampling:    Not affected
  Retbleed:                  Not affected
```

- C910 has no microcode
- OS mitigation: disable vector extension

GhostWrite: Mitigation

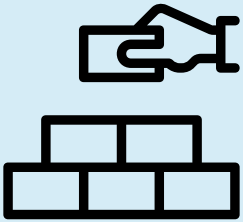
```
Vulnerabilities:
  Gather data sampling:      Not affected
  Ghostwrite:                Mitigation: xtheadvector disabled
  Indirect target selection: Not affected
  Itlb multihit:             Not affected
  L1tf:                      Not affected
  Mds:                       Not affected
  Meltdown:                  Not affected
  Mmio stale data:           Not affected
  Reg file data sampling:    Not affected
  Retbleed:                  Not affected
```

- C910 has no microcode
- OS mitigation: disable vector extension
- Up to 33% overhead

GhostWrite: Mitigation

```
Vulnerabilities:
  Gather data sampling:      Not affected
  Ghostwrite:                Mitigation: xtheadvector disabled
  Indirect target selection: Not affected
  Itlb multihit:             Not affected
  L1tf:                      Not affected
  Mds:                       Not affected
  Meltdown:                  Not affected
  Mmio stale data:           Not affected
  Reg file data sampling:    Not affected
  Retbleed:                  Not affected
```

- C910 has no microcode
- OS mitigation: disable vector extension
- Up to 33% overhead
- Lose ~ 50% instructions



Previous Approaches

Previous Approaches

Pre-silicon

Previous Approaches

Pre-silicon

→ post-silicon

Previous Approaches

Pre-silicon

Golden Model

→ post-silicon

Previous Approaches

Pre-silicon

→ post-silicon

Golden Model

→ differential

Previous Approaches

Pre-silicon

→ post-silicon

Golden Model

→ differential

Non-security Bugs

Previous Approaches

Pre-silicon

→ post-silicon

Golden Model

→ differential

Non-security Bugs

→ user space

Previous Approaches

	Hardware+ user-mode
DIFUZZRTL TheHuzz	✗
Cascade	✗
SiliFuzz	✓
RISCover	✓

[A]Not in disassembler [B]Bug not in RTL [C]RTL not available [D]If T-Head instructions added to Spike [E]If T-Head instructions added to sequence generation

Previous Approaches

	Hardware+ user-mode	GhostWrite
DIFUZZRTL TheHuzz	✗	✗[B]
Cascade	✗	✗[B]
SiliFuzz	✓	✗[A]
RISCover	✓	✓

[A]Not in disassembler [B]Bug not in RTL [C]RTL not available [D]If T-Head instructions added to Spike [E]If T-Head instructions added to sequence generation

Previous Approaches

	Hardware+ user-mode	GhostWrite	C906 hang
DIFUZZRTL TheHuzz	✗	✗[B]	~[D]
Cascade	✗	✗[B]	~[DE]
SiliFuzz	✓	✗[A]	✗[A]
RISCover	✓	✓	✓

[A]Not in disassembler [B]Bug not in RTL [C]RTL not available [D]If T-Head instructions added to Spike
[E]If T-Head instructions added to sequence generation

Previous Approaches

	Hardware+ user-mode	GhostWrite	C906 hang	C908 hang
DIFUZZRTL TheHuzz	✗	✗[B]	~[D]	✗[C]
Cascade	✗	✗[B]	~[DE]	✗[C]
SiliFuzz	✓	✗[A]	✗[A]	✗[A]
RISCover	✓	✓	✓	✓

[A]Not in disassembler [B]Bug not in RTL [C]RTL not available [D]If T-Head instructions added to Spike
[E]If T-Head instructions added to sequence generation

Previous Approaches

	Hardware+ user-mode	GhostWrite	C906 hang	C908 hang	K1 hang
DIFUZZRTL TheHuzz	✗	✗[B]	~[D]	✗[C]	✗[C]
Cascade	✗	✗[B]	~[DE]	✗[C]	✗[C]
SiliFuzz	✓	✗[A]	✗[A]	✗[A]	✗[A]
RISCover	✓	✓	✓	✓	✓

[A]Not in disassembler [B]Bug not in RTL [C]RTL not available [D]If T-Head instructions added to Spike
[E]If T-Head instructions added to sequence generation

Conclusion

Summary

- RISCover: Differential RISC-V CPU fuzzing

fabianthomas.de

lorenz.hetterich@cispa.de

ghostwriteattack.com

Summary

- RISCover: Differential RISC-V CPU fuzzing
- No source code or golden model needed

fabianthomas.de

lorenz.hetterich@cispa.de

ghostwriteattack.com

Summary

- RISCover: Differential RISC-V CPU fuzzing
- No source code or golden model needed
- Many findings within seconds

fabianthomas.de

lorenz.hetterich@cispa.de

ghostwriteattack.com

Summary

- RISCover: Differential RISC-V CPU fuzzing
- No source code or golden model needed
- Many findings within seconds
- GhostWrite (CVE-2024-44067):
  Instruction writing physical memory

fabianthomas.de

lorenz.hetterich@cispa.de

ghostwriteattack.com