FirmRCA: Towards Post-Fuzzing Analysis on ARM Embedded Firmware with Efficient Event-based Fault Localization

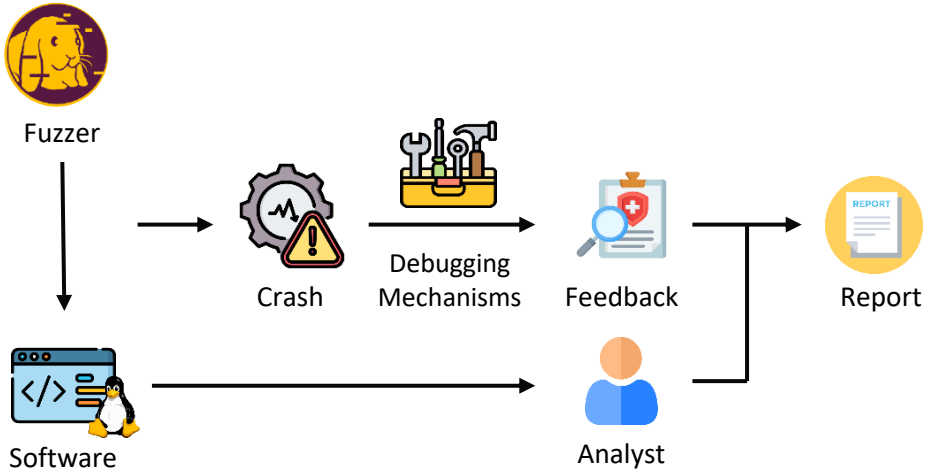
Boyu Chang, Binbin Zhao, Qiao Zhang, Peiyu Liu, Yuan Tian, Raheem Beyah, Shouling Ji

bychang@zju.edu.cn

浙江大学网络系统安全与隐私实验室
NETWORK SYSTEM SECURITY & PRIVACY LAB

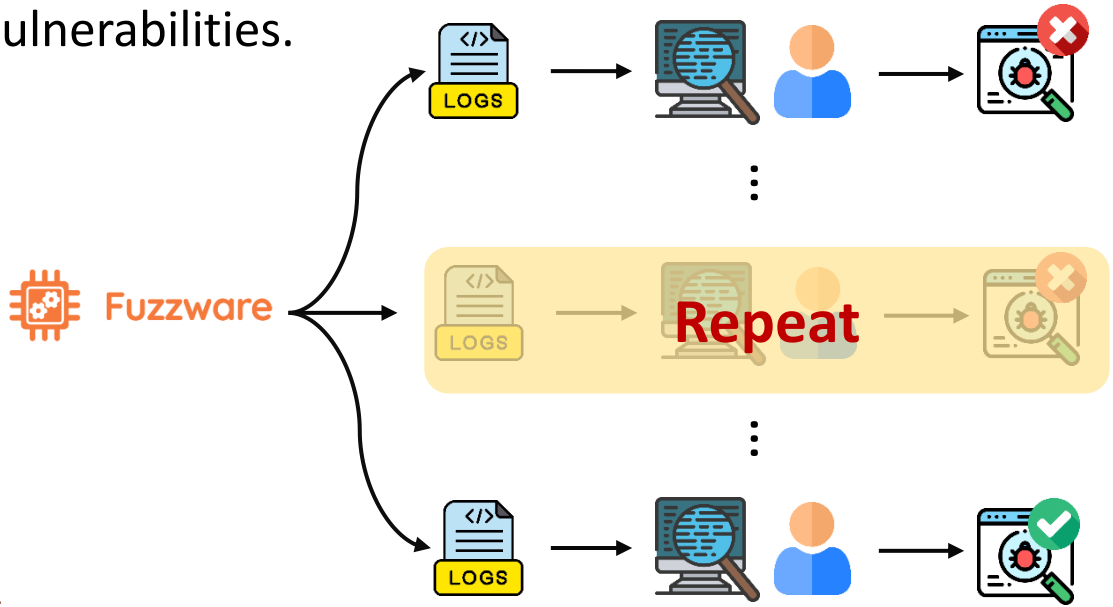
Motivation

- Firmware vulnerabilities threaten IoT devices.
- Fuzzing finds crashes, but fault localization (FL) is tedious and error-prone.



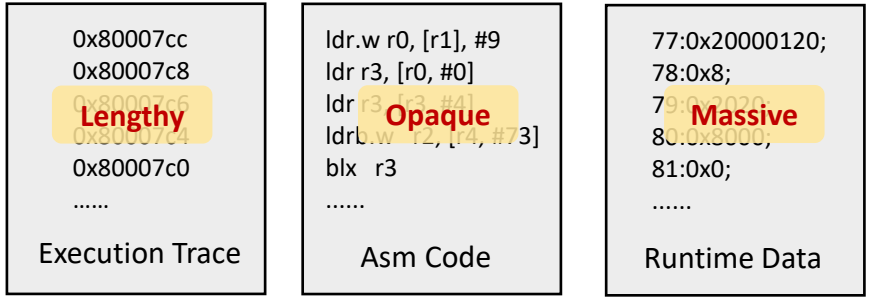
Motivation

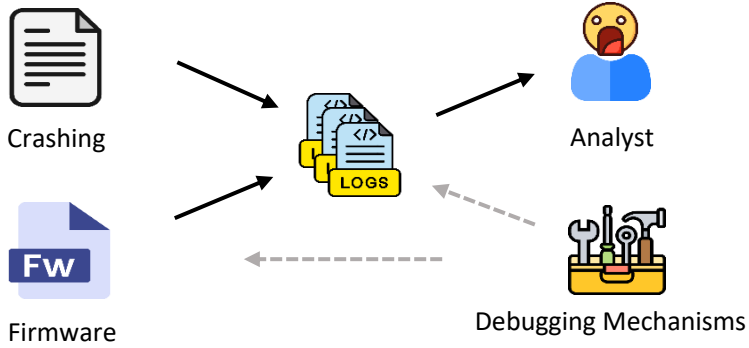
- After fuzzing, analysts face thousands of crashing test cases.
- Manual FL is inefficient, especially on stripped, raw binary firmware.
- Given limited analyst time, efficient FL is essential to prioritize and remediate critical vulnerabilities.



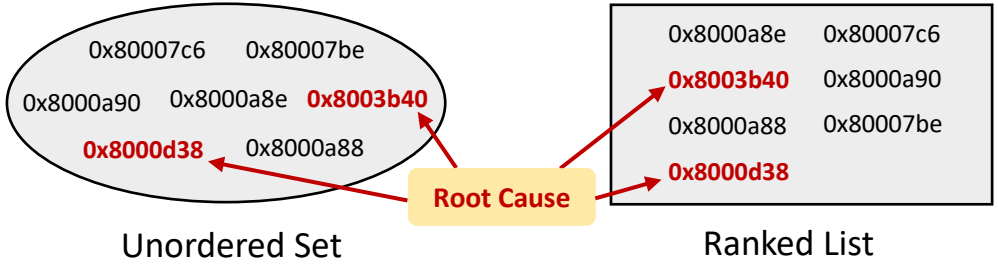
Challenges

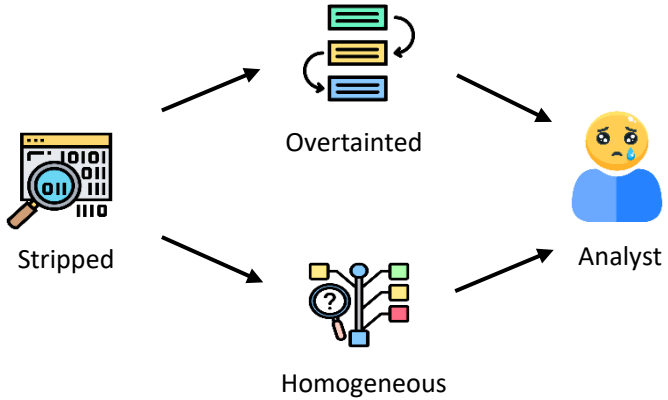
- Challenge 1: Inadequate debugging mechanisms.





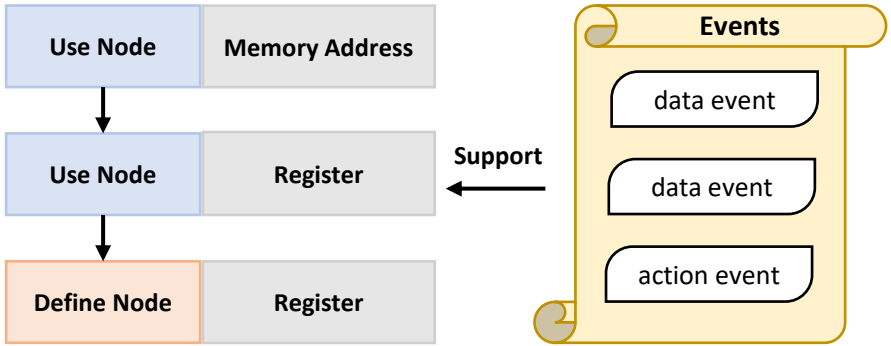
- Challenge 2: Limited investigation guidance.

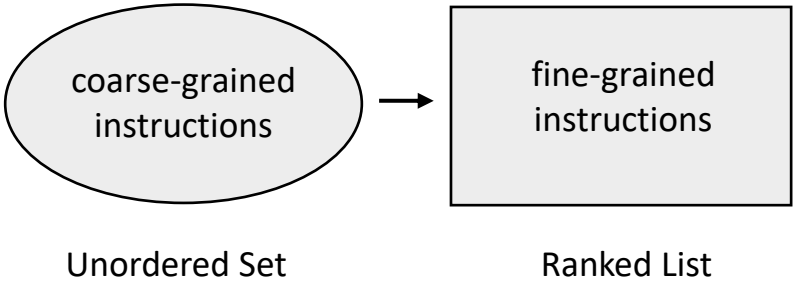




Intuitions

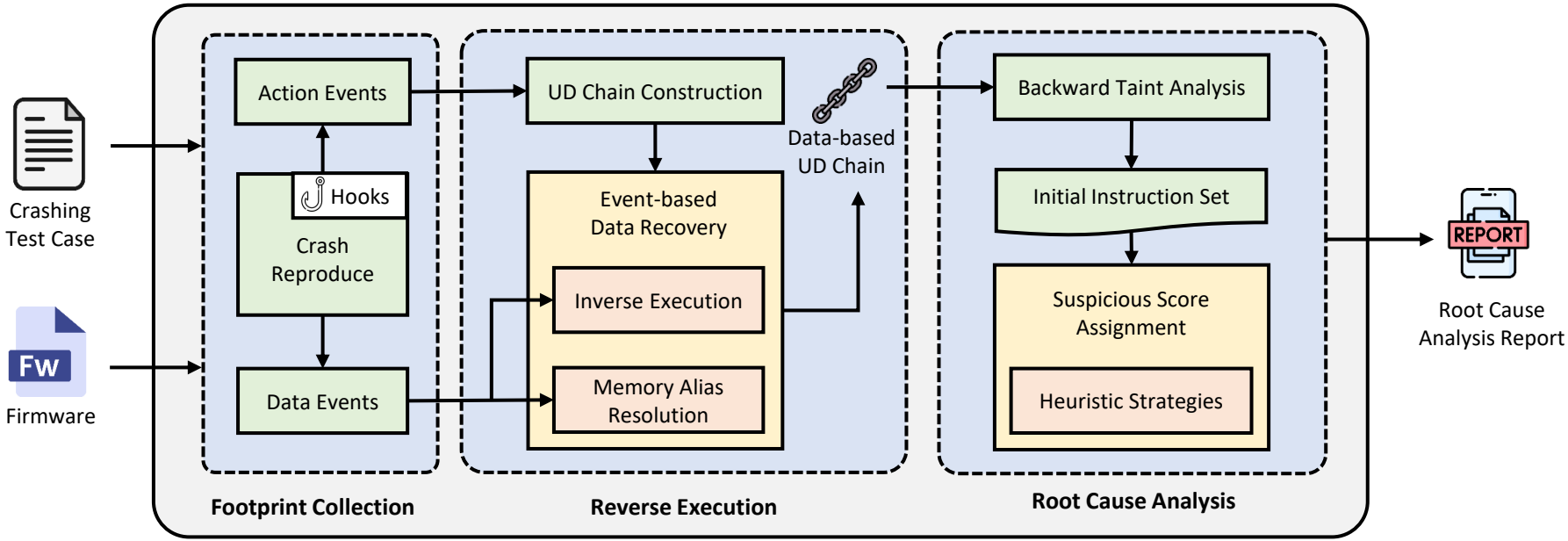
- **Inadequate debugging mechanisms:** reverse taint analysis from the crash site through a data-based use-define chain.
- **Limited investigation guidance:** tainted instructions are not equally important.





FirmRCA Overview

- Event-based data recovery enables precise memory resolution.
- Heuristic strategies prioritize root cause instructions.



FirmRCA – Footprint Collection

- **Problem:** uncertain trace introduces huge time overhead and inaccuracy.
- **Solution:** event-based logging during crash reproduction.

```
net_6lo_uncompress:
  0x406c74 PUSH.W {R3-R11, LR}
  0x406c78 LDR  R6, [R0, #16]
  0x406c7a LDR  R4, [R6, #8]   Crash!
  0x406c7c LDRB R3, [R4, #0]
```

```
ieee802154_recv:
  0x40d126 PUSH.W {R4-R8, LR}
  …
  0x40d1de BL   0x40d0a6      interprocedural
  0x40d1e2 LDR  R0, [R4, #40]
  0x40d1e4 CBZ  R0, 0x40d1f2  intraprocedural
  0x40d1e6 LDRB R3, [R4, #44]
```

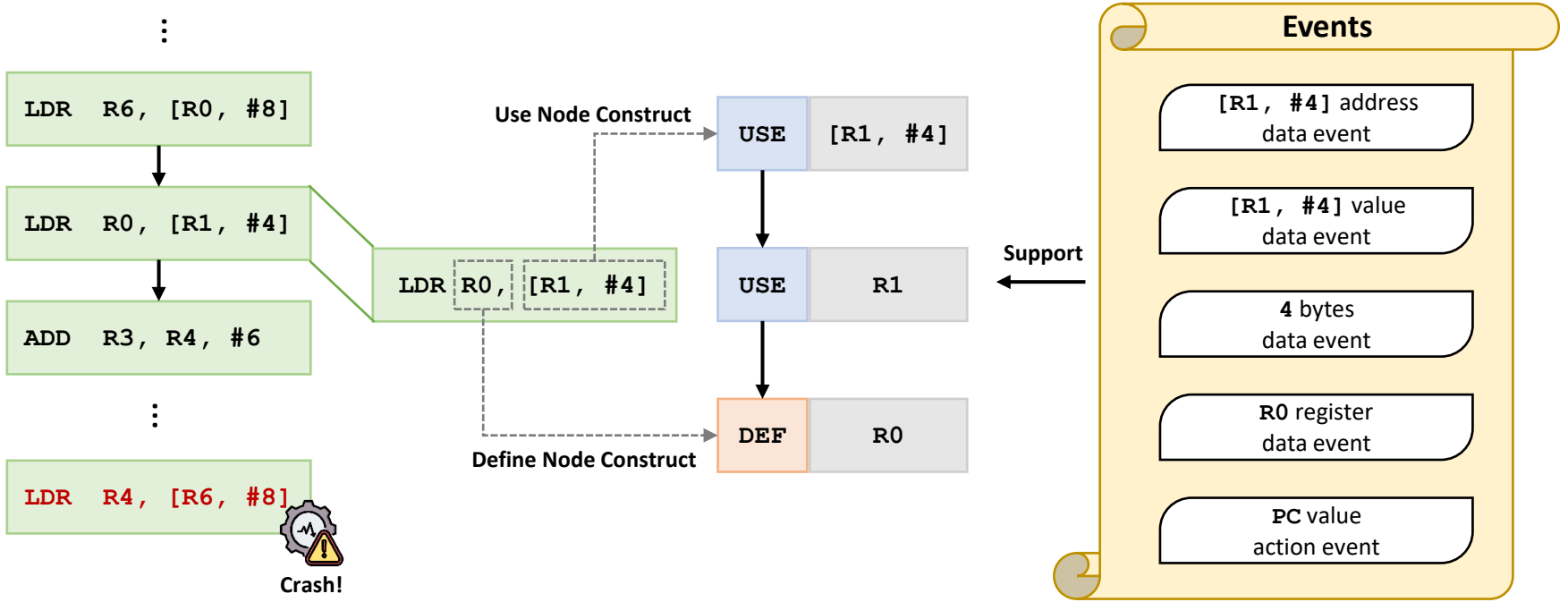
```
struct TraceEvent {
  union {
    instruction @0 :Instruction;
    access @1 :Access;
    dump @2 :Registers;
    crash @3 :Crash;
  }
}
```

```
struct Instruction {
  pc @0 :UInt32;
  lr @1 :UInt32;
}
```

```
struct Access {
  target @0 :AccessTarget;
  type @1 :UInt32;
  size @2 :UInt8;
  pc @3 :UInt32;
  address @4 :UInt32;
  value @5 :UInt32;
}
```

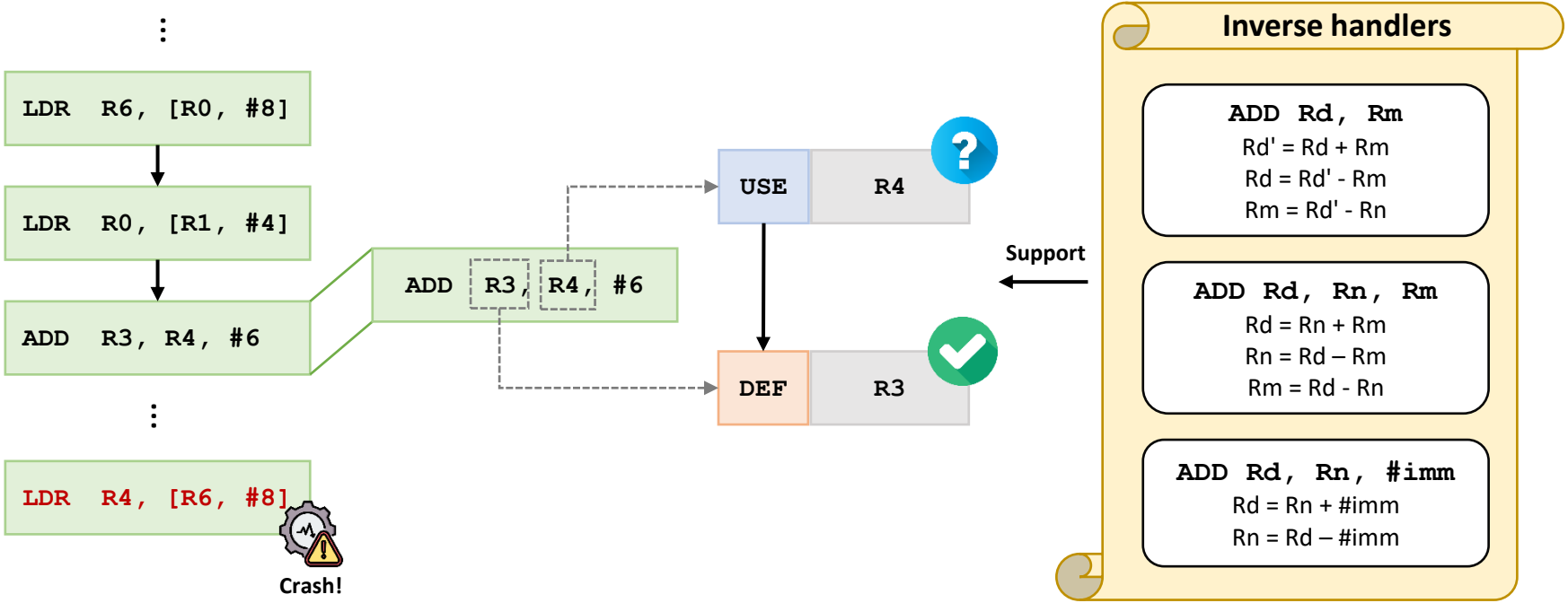
FirmRCA – Reverse Execution

- Data-based use-define chain holds data flow.
- Inverse execution resolves unknown registers.



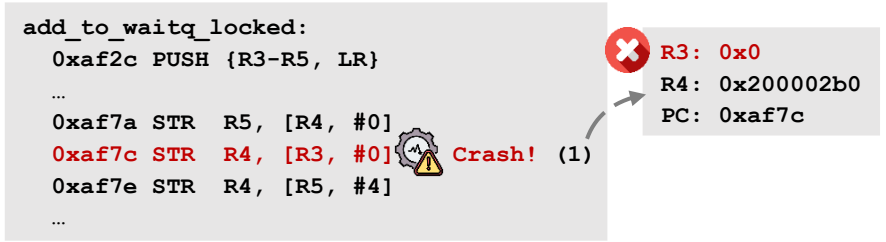
FirmRCA – Reverse Execution

- Data-based use-define chain holds data flow.
- Inverse execution resolves unknown registers.

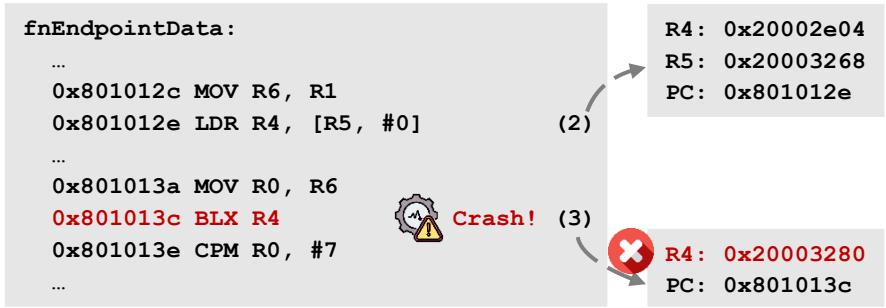


FirmRCA – Root Cause Analysis

- Two kinds of direct crash sites serve as the taint sink.
- Two kinds of heuristic ranking strategies for investigation guidance.



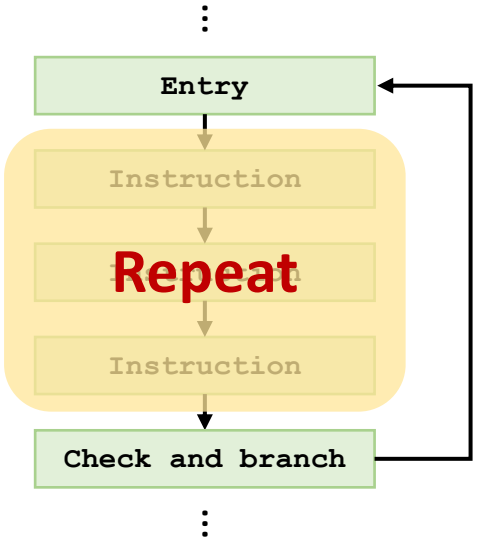
Invalid memory write

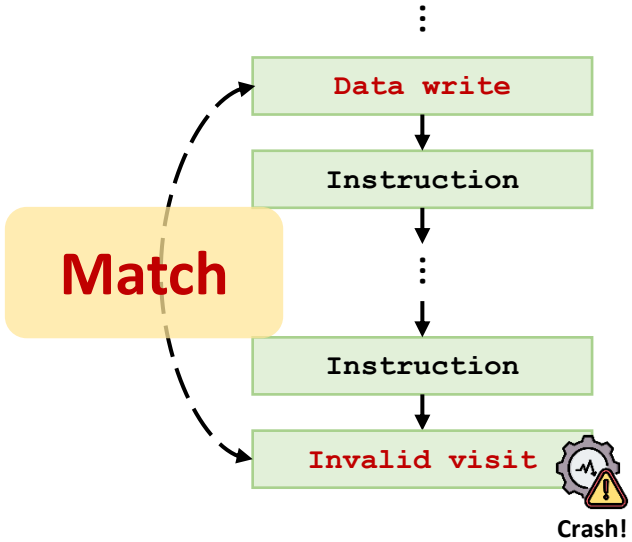


Invalid instruction execution

FirmRCA – Root Cause Analysis

- Two kinds of direct crash sites serve as the taint sink.
- Two kinds of heuristic ranking strategies for investigation guidance.





Redundant loop taint suppression

History write taint prioritization

Evaluation – Effectiveness

- FirmRCA identifies the root cause of 92.7% test cases in top 10 instructions.
- FirmRCA localizes deep root causes of 90.9% test cases.

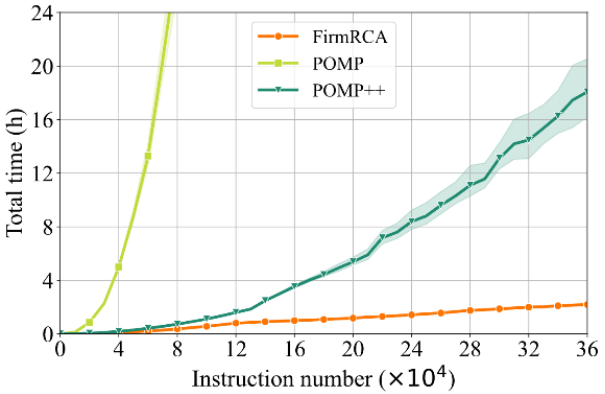
ID	ΔRoot(%)	# Traces	# Ins	Full			Half		
				P	P+	F	P	P+	F
C_1	2 (<0.1%)	52,080	11,869	∅	✗	1	✗	✗	1
C_2	2 (<0.1%)	371,762	13,374	∅	∅	1	∅	∅	1
C_3	2 (<0.1%)	1,003,421	13,374	∅	∅	1	∅	∅	1
C_4	2 (<0.1%)	228,012	5,021	∅	✗	1	∅	✗	1
C_5	2 (<0.1%)	36,271	11,880	✗	✗	1	✗	✗	1
C_6	2 (<0.1%)	108,294	7,126	∅	✗	1	∅	✗	1
C_7	2 (<0.1%)	619,937	15,140	∅	∅	1	∅	∅	1
C_8	2 (<0.1%)	68,894	15,513	∅	✗	1	✗	✗	1
C_9	2 (<0.1%)	524,897	11,931	∅	∅	1	∅	∅	1
C_{10}	20,203 (18.1%)	111,898	18,536	∅	✗	2	✗	✗	2
C_{11}	78 (<0.1%)	168,542	17,165	∅	1	1	1	1	1
C_{12}	120,472 (51.1%)	235,662	9,299	∅	✗	13	✗	✗	8
C_{13}	2,176 (1.4%)	160,721	9,299	∅	∅	21	∅	✗	19
C_{14}	8,155 (39.5%)	20,630	15,428	✗	✗	1	✗	✗	1
C_{15}	8,719 (73.0%)	11,943	17,165	✗	✗	1	✗	✗	✗
C_{16}	215,102 (98.7%)	217,900	17,165	∅	✗	10	∅	✗	✗
C_{17}	5,183 (3.0%)	173,913	17,165	∅	8	1	✗	8	1
C_{18}	2,759 (1.6%)	169,666	17,165	∅	✗	1	✗	✗	1
C_{19}	5,161 (72.2%)	7,151	9,299	✗	✗	1	✗	✗	✗
C_{20}	306 (<0.1%)	523,873	14,325	∅	✗	1	✗	✗	1
C_{21}	2,013 (1.3%)	150,995	19,706	∅	✗	2	✗	✗	2

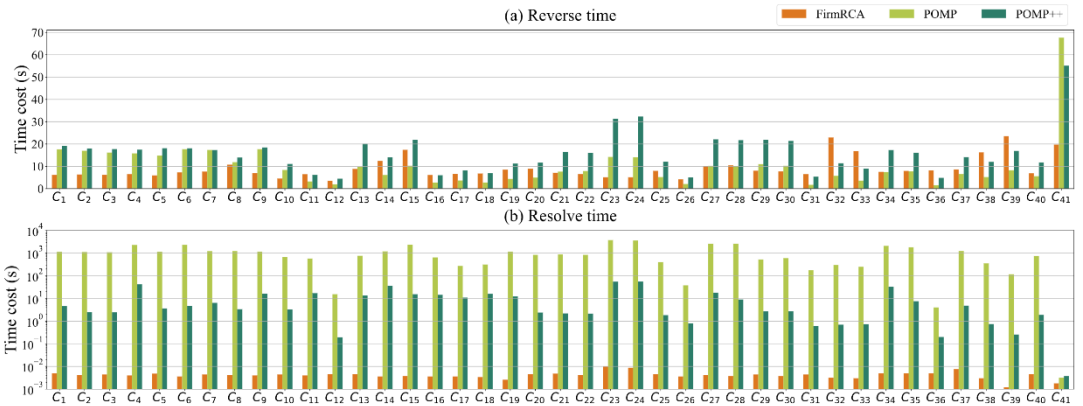
ID	ΔRoot(%)	# Traces	# Ins	Full			Half		
				P	P+	F	P	P+	F
C_{22}	401 (0.2%)	238,470	13,954	∅	✗	2	∅	✗	2
C_{23}	53 (<0.1%)	89,136	32,546	∅	✗	1	∅	✗	1
C_{24}	53 (0.1%)	52,909	32,546	∅	✗	1	✗	✗	1
C_{25}	1,789 (0.4%)	434,172	13,954	∅	1	2	∅	1	2
C_{26}	88,746 (45.2%)	196,313	29,993	∅	3	1	3	3	1
C_{27}	27,411 (58.2%)	47,065	15,529	3	3	3	3	3	4
C_{28}	28,771 (58.9%)	48,880	15,529	2	2	2	2	2	3
C_{29}	55,768 (66.4%)	84,028	23,706	∅	✗	1	✗	✗	✗
C_{30}	59,717 (63.3%)	94,362	23,706	∅	✗	1	✗	✗	✗
C_{31}	402 (<0.1%)	514,772	14,325	✗	✗	1	✗	✗	1
C_{32}	86,956 (8.7%)	994,960	27,664	∅	∅	✗	∅	∅	✗
C_{33}	100,512 (48.2%)	208,669	18,464	∅	2	2	∅	2	3
C_{34}	21 (<0.1%)	160,958	18,475	∅	✗	1	∅	✗	1
C_{35}	337 (<0.1%)	1,960,419	26,165	∅	∅	1	∅	∅	1
C_{36}	274,858 (27.1%)	1,016,078	26,167	∅	∅	1	∅	∅	✗
C_{37}	1,458,974 (99.0%)	1,474,404	18,202	∅	∅	1	∅	∅	✗
C_{38}	44,256 (3.4%)	1,300,470	25,887	∅	∅	1	∅	∅	1
C_{39}	42,177 (53.6%)	78,624	15,605	∅	✗	1	✗	✗	1
C_{40}	891 (1.7%)	52,173	11,952	4	4	5	4	4	5
C_{41}	594,471 (95.1%)	625,021	12,258	∅	∅	1	∅	✗	✗

△Root denotes the distance between the root cause and crash site; ∅ indicates the analysis timed out; ✗ indicates the root cause is not found
Failures in the FirmRCA analysis of full instructions are underlined ; ID is highlighted in GRAY if its △Root is larger than 50%

Evaluation – Efficiency

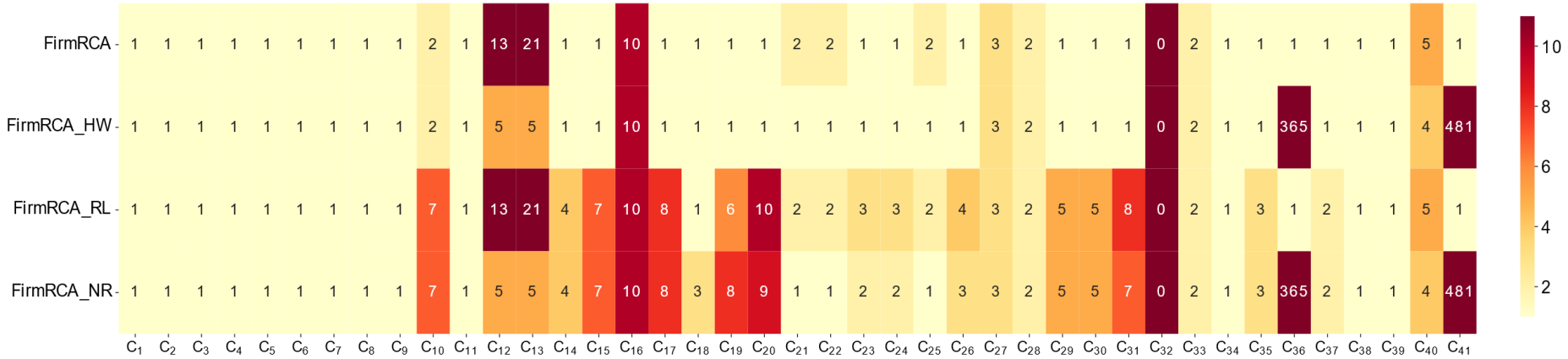
- FirmRCA shows a modest overall time cost as the analysis depth increases.
- The efficient memory alias resolution is a key contributor.





Evaluation – Ranking Strategies

- All four prototypes demonstrate high success rates.
- *Firm_RL* helps level up the rank to the top 1.
- *Firm_HW* enhances the ranking results in 36.6% of the test cases.



***FirmRCA_NR**: no ranking strategies enabled; **FirmRCA_RL**: the redundant loop taint suppression strategy enabled*
***FirmRCA_HW**: the history write prioritization strategy enabled; **FirmRCA**: both ranking strategies enabled*

FirmRCA: Towards Post-Fuzzing Analysis on ARM Embedded Firmware with Efficient Event-based Fault Localization

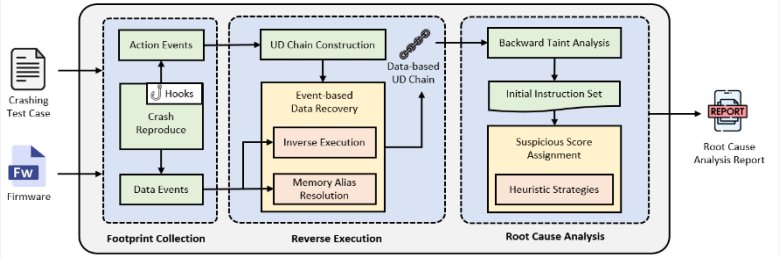
Challenges

Challenge 1: Inadequate debugging mechanisms.

Challenge 2: Limited investigation guidance.

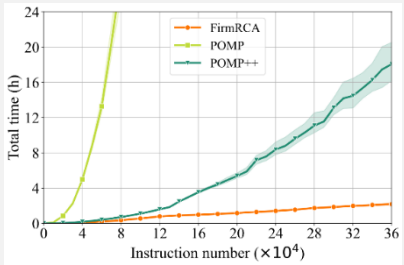
FirmRCA Framwork

An efficient event-based fault localization work.



Effectiveness and Efficiency

ID	ΔRoot(%)	# Traces	# Ins	Full			Half		
				P	P↑	Γ	P	P↑	Γ
C_1	2 (<0.1%)	52,080	11,869	∅	✗	1	✗	✗	1
C_2	2 (<0.1%)	371,762	13,374	∅	∅	1	∅	∅	1
C_3	2 (<0.1%)	1,003,421	13,374	∅	∅	1	∅	∅	1
C_4	2 (<0.1%)	228,012	5,021	∅	✗	1	∅	✗	1
C_5	2 (<0.1%)	36,271	11,880	✗	✗	1	✗	✗	1
C_6	2 (<0.1%)	108,294	7,126	∅	✗	1	∅	✗	1
C_7	2 (<0.1%)	619,937	15,140	∅	∅	1	∅	∅	1
C_8	2 (<0.1%)	68,894	15,513	∅	✗	1	✗	✗	1
C_9	2 (<0.1%)	524,897	11,931	∅	∅	1	∅	∅	1
C_{10}	20,203 (18.1%)	111,898	18,536	∅	✗	2	✗	✗	2
C_{11}	78 (<0.1%)	168,512	17,165	∅	1	1	1	1	1
C_{12}	120,472 (51.1%)	235,662	9,299	∅	✗	13	✗	✗	8
C_{13}	2,176 (1.4%)	160,721	9,299	∅	∅	21	∅	✗	19
C_{14}	8,155 (39.5%)	20,630	15,428	✗	✗	1	✗	✗	1
C_{15}	8,719 (73.0%)	11,943	17,165	✗	✗	1	✗	✗	✗
C_{16}	215,102 (98.7%)	217,900	17,165	∅	✗	10	∅	✗	✗
C_{17}	5,183 (3.0%)	173,913	17,165	∅	8	1	✗	8	1
C_{18}	2,759 (1.6%)	169,666	17,165	∅	✗	1	✗	✗	1
C_{19}	5,161 (72.2%)	7,151	9,299	✗	✗	1	✗	✗	✗
C_{20}	306 (<0.1%)	523,873	14,325	∅	✗	1	✗	✗	1
C_{21}	2,013 (1.3%)	150,995	19,706	∅	✗	2	✗	✗	2



Paper

Code

IEEE S&P 2025
May 12-15

bychang@zju.edu.cn

浙江大学 ZHEJIANG UNIVERSITY
浙江大学网络系统安全与隐私实验室
NETWORK SYSTEM SECURITY & PRIVACY LAB