A Systematic Study of Recent Smart Contract Security Vulnerabilities

Zhuo Zhang, Brian Zhang, Wen Xu, Zhiqiang Lin

PURDUE UNIVERSITY

Data Collection

Code4rena Bug Reports

Categories	# Cont	Bounty	# Atten	# Vuln	TVL
Lending	20	$1,145K	180	53	$304.8M
Dexes	13	$1,020K	139	70	$898.9M
Yield	12	$ 970K	193	85	$304.8M
Services	11	$ 532K	123	21	$219.8M
Derivatives	9	$ 525K	123	13	$147.8M
Yield Aggregator	9	$ 365K	124	22	$265.5M
Real World Assets	7	$ 405K	69	10	$ 41.8M
Stablecoins	6	$ 365K	102	7	$364.7M
Indexes	6	$ 215K	101	7	$ 1.0M
Insurance	5	$ 298K	74	19	$ 42.9M
NFT Marketplace	4	$ 266K	126	8	$ 46.6M
NFT Lending	4	$ 230K	108	10	$ 8.2M
Cross Chain	4	$ 250K	81	7	$ 32.0M
Others	3	$ 110K	25	9	$118.3M
Total	113	$6.696M	358	341	$2.797B

Real-world Exploits

Categories	Attacks		Bug Bounties	
	# Bugs	Fund loss	# Bugs	Bounties
Lending	1	$ 5,000K	2	$ 1,630K
Dexes	7	$ 13,950K	3	$ 65K
Yield	6	$ 20,300K	1	$ 10K
Services	3	$ 5,600K	2	$ 610K
Derivatives	-	-	2	$ 200K
Yield Aggregator	1	$ 2,100K	2	$ 300K
Real World Assets	2	$ 1,127K	1	$ 50K
Stablecoins	5	$211,360K	-	-
Indexes	-	-	1	$ 90K
NFT Marketplace	1	$ 20K	-	-
NFT Lending	2	$ 5,800K	-	-
Cross Chain	-	-	1	$10,000K
Others	-	-	1	$ 1,050K
Total	28	$265,257K	16	$14,005K

Finding 1: *Although the DeFi community has heavily in- vested on protecting their products, the current supply of tools and human auditor resources have not met the demand.*

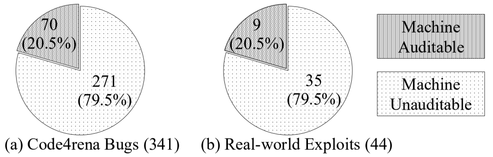
Machine Auditable Bugs & Existing Tools

Assertion Failure (***AF***), Arbitrary Write (***AW***), Block-state Dependency (***BD***), Compiler Error (***CE***), Control-flow Hijack (***CH***), Ether Leak (***EL***), Freezing Ether (***FE***), Gas-related Issue (***GI***), Integer Bug (***IB***), Mishandled Exception (***ME***), Precision Loss (***PL***), Reentrancy (***RE***), Suicidal Contract (***SC***), Transaction-ordering Dependency (***TD***), Transaction Origin Use (***TO***), Uninitialized Variable (***UV***), Weak PRNG (***WP***)

Kind	Tool	Year	Com	Orcl	Machine-auditable Bugs: AF	AW	BD	CE	CH	EL	FE	GI	IB	ME	PL	RE	SC	TD	TO	UV	WP
Fuzzing	ReGuard [46]	18		○			✓		✓		✓	✓		✓		✓					
	ContractFuzzer [47]	18		○					✓		✓	✓		✓		✓					
	ILF [7]	19		○			✓		✓	✓	✓			✓		✓	✓				
	Vultron [48]	19		○								✓	✓	✓		✓					
	sFuzz [49]	20	✓	○	✓	✓	✓	✓	✓	✓		✓	✓								
	Harvey [3]	20	✓	○	✓	✓							✓	✓		✓					
	ContraMaster [50]	20		●								✓	✓	✓		✓					
	ConFuzzius [51]	21		○	✓		✓		✓	✓	✓		✓	✓		✓	✓	✓			
	Smartian [6]	21	✓	○	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓		✓		
	Echidna [52], [53]	21	✓	◐	✓																
	xFuzz [54]	22		○					✓							✓			✓		
Static Analysis	Gasper [55]	17		○								✓									
	Securify [56]	18	✓	○		✓	✓		✓	✓	✓			✓		✓	✓	✓	✓		
	Vandal [57]	18		○						✓	✓			✓		✓	✓		✓		
	MadMax [58]	18		○								✓	✓	✓							
	SASC [59]	18		○			✓	✓								✓			✓		
	SmartCheck [60]	18	✓	○			✓			✓	✓	✓	✓	✓		✓			✓		
	Zeus [61]	18		○			✓						✓	✓		✓	✓	✓	✓		
	Slither [14]	19	✓	○		✓	✓	✓	✓	✓	✓			✓	✓	✓	✓		✓	✓	✓
	Sereum [62]	19		○					✓							✓					
	NPChecker [63]	19		○			✓		✓							✓		✓	✓		
	Sensors [64]	22		○					✓				✓			✓					
	Remix [65]	22	✓	○	✓		✓							✓		✓			✓		
Verification	ECF [66]	17		○												✓					
	Solc-Verify [67]	19		○	✓								✓			✓					
	VeriSol [68]	19	✓	◐	✓																
	VeriSmart [69]	20		○					✓				✓								
	Solid [70]	21		○												✓					
Symbolic Execution	Oyente [8]	16	✓	○	✓		✓				✓		✓			✓	✓	✓			
	Maian [71]	18		○						✓	✓						✓				
	teEther [72]	18		○	✓				✓	✓							✓				
	Osiris [73]	18		○	✓		✓						✓			✓					
	Manticore [74]	19	✓	○	✓		✓		✓	✓			✓	✓		✓	✓		✓		
	sCompile [75]	19		○	✓				✓	✓	✓					✓	✓				
	M-A-R [76]	21		○												✓					
	SmarTest [77]	21		○	✓					✓			✓				✓				
	Mythril [78]	22	✓	○	✓	✓	✓		✓	✓			✓	✓		✓	✓	✓	✓		
	Sailfish [79]	22		○												✓		✓			

Finding 2: *Existing techniques rely on simple and general oracles or hand-coded ones that are project specific. Such oracles may not be sufficient for functional bugs in general.*

Breakdown of the Bugs

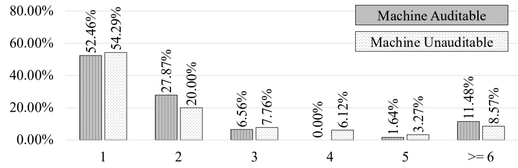


(a) Code4rena Bugs (341) (b) Real-world Exploits (44)

Finding 3: *A large portion of exploitable bugs in the wild (i.e., 80%) are not machine auditable..*

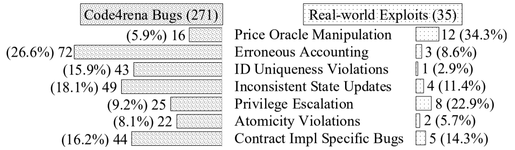
Auditing Difficulty

Overall Auditing Difficulty



Finding 4: *Majority of exploitable bugs are difficult to find.*

Breakdown of Machine Unauditable Bugs (MUB)



Finding 5: *MUBs can be classified to 7 categories, with 85% belonging to first six categories that are not project specific.*

Auditing Difficulty of MUBs

Types	# Auditors: 1	2	3	4	5	>= 6
Price Oracle Manipulation	75.00%	12.50%	0.00%	0.00%	0.00%	12.50%
Erroneous Accounting	59.09%	21.21%	7.58%	6.06%	3.03%	3.03%
ID Uniqueness Violations	42.86%	17.14%	8.57%	11.43%	5.71%	14.29%
Inconsistent State Updates	53.33%	22.22%	2.22%	6.67%	6.67%	8.89%
Privilege Escalation	56.52%	21.74%	8.70%	4.35%	0.00%	8.70%
Atomicity Violations	57.14%	19.05%	4.76%	4.76%	4.76%	9.52%
Contract Impl Specific Bugs	46.15%	20.51%	17.95%	5.13%	0.00%	10.26%

Finding 6: *Different types of MUBs have different auditing difficulties, with price oracle manipulation and ID unique- ness violation bugs the hardest and the easiest, respectively*

Breakdown of the MUBs w.r.t. DeFi Categories

Categories	Code4rena Bugs: C1	C2	C3	C4	C5	C6	C7
Lending	3	6	4	7	9	6	8
Dexes	2	16	8	15	3	1	6
Yield	7	23	17	8	5	6	10
Services	0	4	2	5	0	0	3
Derivatives	1	6	1	0	2	0	1
Yield Aggregator	1	6	0	5	0	1	4
Real world assets	1	0	4	3	0	0	1
Stablecoins	0	2	1	0	0	0	2
Indexes	0	0	1	2	0	2	0
Insurance	0	3	1	3	3	2	4
NFT Marketplace	0	1	1	0	1	2	1
NFT Lending	1	1	1	0	1	1	1
Cross Chain	0	1	1	1	1	0	1
Others	0	3	1	0	0	1	2

Finding 7: *Different kinds of DeFi projects tend to be prone to different types of MUBs.*

Difficulty of Bug Fix

	C1	C2	C3	C4	C5	C6
LoC (+)	9.6	6.0	12.4	14.8	6.6	8.5
LoC (-)	2.6	4.4	8.4	13.2	0.6	5.8

Finding 8: *It is usually not difficult to fix MUBs (i.e., with 15 lines of changes on average).*