March 2024
Nadir Khan (@Nadir_Khan_Sec)

EVALUATING TEST SUITE ROBUSTNESS OF **TOP DEFI PROTOCOLS**

March 2024
Nadir Khan (@Nadir_Khan_Sec)

Executive Summary

This research evaluated the testing suites of top DeFi protocols. 400+ test suites across the top 10 DeFi protocols were analyzed (collectively holding $60B TVL). Tests were flagged by type (dynamic, static, fuzz, unit, invariant and formal methods). The report concludes that most protocols employ only basic testing practices. Overall, top DeFi protocols are not sufficiently testing their code prior to deployment. Security experts must better assist to ensure rigorous testing practices

Findings

- Insufficient testing prior to deployment.
- Top DeFi protocols need to move beyond basic unit and integration testing.
- Stateful fuzzing was rarely used. Protocols with fuzzing have small (stateless) testing suites.
- Formal verification techniques not consistently used.
- Most audit reports do not suggest further testing, or provide additional test suites.

Introduction + Methodology

The top 10 DeFi Protocols were analyzed from January-March 2024. Each protocol's website, Twitter, GitHub, available documentation and past audit reports were examined. Next, the testing suite of the protocol was analyzed manually to flag for the following: unit testing, invariant testing, fuzz testing, formal verification techniques, tooling, number of past audits, firms engaged, number of test files, and existence of bug bounty program. Samples of the testing suite (10-20% of files) were also analyzed with GPT4 to clarify uncertainties, confusing tests or long test suites with complex logic. The protocols were rated and a recommendation was provided.

[1] This research topic was very helpfully suggested by Jackson (@sjkelleyjr) and Stephan Johnson(@StephanJohnson_). Jackson has an excellent resource and community for aspiring web3 security researchers available at https://jacksonkelley.gumroad.com/l/how-to-become-a-smart-contract-auditor.

[2] https://defillama.com/. Note that due to fluctuating TVL the top 10 protocols may change frequently.

Protocol	Use Case	TVL	Unit?	Invariant?	Fuzz?	FV?	Tools	Past audits?	Firm Engaged	Test files Anlyzed	Bug Bounty?
Lido	Liquid Staking	$38.5b	Yes	Yes	Yes	No	HH, Foundry	15+	Oxorio, OpenZeppelin, ChainSecurity, Statemind, etc	83	Yes
EigeLayer	Restaking	$12.3b	Yes	Yes	Yes	SE	Foundry, Certora, Mythril, Slithe	2	Consensys (2023), SigmaPrime (2023	56	Yes
AAVE	Lending	$11.5b	Yes	Yes	Yes	No	HH, Certora	8	OpenZeppelin, ToB, PeckShield, SigmaPrime, etc.	78	Yes
Maker	DAI Stablecoin, DAO	$9.5b	Yes	Yes	Yes	No	Foundry, Certora	3	PechShield, Trail of Bits, Runtime Verification, etc.	14	Yes
JustLend	Lending	$7.9b	No	No	No	No	None	2	CertiK (2022), Slowmist (2023)	N/A	Yes

Protocol	Use Case	TVL	Unit?	Invariant?	Fuzz?	FV?	Tools	Past audits?	Firm Engaged	Test files Anlyzed	Bug Bounty?
Uniswap v3	DEX	$6b	Yes	Yes	Yes	SE	HH, Slither, Echidna, Manticore	2	ToB (2021), ABDK (2021)	57	Yes
Summer.fi	Services	$6b	Yes	Yes	Yes	No	HH	4	ChainSecurity (2021, 2022)	19	Yes
Rocket Pool	Liquid Staking	$4.8b	Yes	Yes	Only by ToB	SE	HH, Slither, Echidna, Manticore	8	ToB (2021), Consensys (2021, 2022, 2023), Sigma Prime (2021, 2022)	62	Yes
Spark	Lending	$3.7b	Yes	No	Yes	No	Foundry	3	ChainSecurity (2023)	9	Yes
Curve Finance	DEX	$2.59b	No	No	No	No	Brownie	12	ToB,ChainSecurity, MixBytes, QuantStamp (2020-2024)	63	Yes

Lido, Rating : B+

- Relatively complete testing suite, though it is centered on unit and integration tests.
- Does not employ stateful fuzzing or formal verification techniques. Basic invariant testing. Very extensive auditing record.
- Recommendation: Incorporate formal techniques and conduct stateful fuzzing

EigenLayer, Rating: A-

- Focus on integration testing and unit testing.
- Good use of fuzzing and various scenarios; first-class documentation and
- organization; Symbolic execution technique is employed.
- Recommendation: Incorporate stateful fuzzing, more deeply test invariants.

AAVE, Rating: A-

- Predominantly unit testing and integration; No indication of fuzz testing.
- Excellent use of formal verification techniques.
- Recommendation: Employ fuzzing techniques, continue regular audits.

Maker DAO, Rating: B+

- Protocol's design was made to be formally verified; extensive formal verification by Certora team.
- Recommendation: Incorporate greater fuzz testing.

JustLend, Rating: F

- Audits conducted 2 years after deployment; Audit firm used only static analysis and manual review.
- Protocol's response to several findings was that the contract had already been deployed
- No test suite; No unit, invariant, fuzz, or formal verification. Protocol currently holds $7.9 billion in TVL.
- Recommendation: Implement basic security practices to protect user funds. Review all contracts with a well regarded auditing firm.

UniswapV3, Rating: B+

- Significant invariant and unit testing
- Limited fuzz testing; 2 past audits
- Formal Verification: Symbolic execution used by Trail of Bits
- Recommendation: Deeper fuzz testing and stateful fuzz testing

Summer.fi, Rating: D-

- Clear documentation; includes specific documentation for test suite.
- Only 1 contract (Multiply Proxy Actions) of Summer.fi's five layer architecture has a test suite.
- 4 audit reports by ChainSecurity on the formerly named Oasis protocol. Has not been audited in 2 years.
- No invariant and fuzz testing.
- Recommendation: conduct a current audit, ensure testing coverage across all contracts, implement invariant and fuzz testing.

RocketPool, Rating: B

- Although several audits listed on website, most cannot be accessed/downloaded
- Test suite focused on integration, unit testing and basic scenarios. No fuzzing done by the protocol.
- Recommendation: Integrate ToB Audit suggestion for deeper fuzzing, all audits should be clearly accessible.

Spark, Rating: D-

- Spark claims to follow security standards of Maker DAO, but has not implemented invariant testing and FV.
- Github is not well organized and the core contracts cannot be easily found.
- Very limited and basic testing suite. No invariant testing.

Curve Finance, Rating: B+

- Basic unit and integration testing; some edge case testing; some basic fuzzing with limited randomness.
- Employed some stateful fuzzing; extensive and frequent auditing.
- Recommendation: Can benefit from formal verification.

Conclusion

Beyond Basic Testing: Top DeFi protocols need to move beyond basic unit and integration testing.Protocols must more deeply test invariants and fuzz their codes. They must test for a variety of scenarios and attack vectors. Currently, insufficient testing is occurring prior to deployment among most top protocols.

Inadequate Invariant Testing: In almost all protocols, invariant testing was either not present or partially/superficially tested.

Lack of Stateful fuzzing: Stateful fuzzing was rarely used. Protocols with fuzz testing have small (stateless) testing suites. Several high quality fuzzers exist that can be used to better secure codebases.

Inconsistent Formal Verification: Formal verification techniques not consistently used.

Service of Security Researchers: Most audit reports do not suggest further testing, or provide additional test suites.