Invariant Fuzz Testing

What is it?
and
How can it make your protocol more secure?

Chris Smith

Currently:

Security Researcher and Consultant

- Traditional Audits
- Invariant Security Engineering and Reviews

Protocol Engineering/Technical Advisor

Previously:

- Pre-Endgame MakerDAO Protocol Engineering Core Unit
- Maker Foundation Smart Contract Engineer
- Backend and Smart Contract Engineer with ConsenSys

Presentation and Code:

https://github.com/iamchrissmith/2024-ethdenver-invariant-testing

Invariant Fuzz Testing

a.k.a Invariant Testing, Forge Invariant Testing, Stateful Fuzzing Tests

“Invariant testing allows for a set of invariant expressions to be tested against **randomized sequences** of pre-defined function calls from pre-defined contracts. After each function call is performed, **all defined invariants are asserted**.

Invariant testing is a powerful tool to expose incorrect logic in protocols. Due to the fact that function call sequences are **randomized and have fuzzed inputs**, invariant testing can expose false assumptions and incorrect logic in edge cases and highly complex protocol states.” (emphasis added)

Read more: https://book.getfoundry.sh/forge/invariant-testing branch: main

branch: 1-invariant-setup

branch: 2-handler-setup

branch: 3-add-balance-invariant

Limitations and Next Steps