According to Chainalysis, 2023 suffered around $1.7 billion in stolen assets

. Even though the losses have decreased compared to 2022, we still have a long way to go as the industry matures.

Building a more robust and reliable Web3 is hard work, and a big part of getting there is ensuring developers have the tools to build secure smart contracts.

However, prioritizing the security of your codebase as a smart contract engineer and performing security reviews can be a long and difficult task.

That's why we've compiled a list of the top 8 smart contract auditing and security tools

our auditors think every smart contract developer should include in their stack.

Top 9 Industry-leading Smart Contract Auditing and Security Tools

1. Best Smart Contract Fuzzing Tool Overall: Echidna

Fuzz testing is necessary for any blockchain project, and auditors must know how to perform it correctly.

Echidna is a smart contract security tool that usesproperty-based fuzzing to discover vulnerabilities by testing contracts against user-defined predicates

.

Developed by Trail of Bits, Echidna is known for its flexibility and comprehensive toolset, able to break the most difficult assertions. It's an ideal choice for developers to run fuzz tests while seeking to ensure the robustness and security of their contracts.

Price

: Free

Key Features

- Property-Based Fuzzing:

A dynamic testing method that challenges smart contracts with unexpected inputs to ensure they behave as intended under various conditions.

- User-Defined Properties:

Developers can define specific properties or assertions the smart contract should uphold, enabling Echidna to target testing efforts more precisely and uncover vulnerabilities related to these properties.

- Coverage Reporting:

Integrates source code analysis to report which lines of code were covered during the fuzzing campaign, aiding developers in understanding the thoroughness of the tests conducted.

However, the Trail of Bits team didn't stop at Echidna and developed another tool in our list of top smart contract security auditing tools: Medusa

2. Best Experimental Fuzz Testing Tool: Medusa

Medusa is an experimental smart cross-platform go-ethereum-based smart contract fuzzer inspired by Echidna, also built by Trail Of Bits.

It provides parallelized fuzz testing of smart contracts through CLI or its Go API (subject to breaking changes), allowing custom user-extended testing methodology.

Because under development, Medusa couldn't make it to the first position of our top smart contract auditing tools list, but it revealed itself as [one of the most powerful publicly available smart contract fuzzers]

Price:

Free

Key Features

- Parallel fuzzing and testing methodologies

across multiple workers (threads)

- Assertion and property testing:

built-in support for writing basic Solidity property tests and assertion tests

- Mutational value generation:

fed by compilation and runtime values.

- Coverage collecting:

Coverage-increasing call sequences are stored in the corpus

- Coverage guided fuzzing

: Coverage-increasing call sequences from the corpus are mutated to guide further the fuzzing campaign

- Extensible low-level testing API

through events and hooks provided throughout the fuzzer, workers, and test chains.

3. Best Fuzzing as a Service (FaaS): Diligence Fuzzing

[Diligence Fuzzing

](https://consensys.io/diligence/fuzzing/) is another smart contract fuzz testing tool that couldn't be missed on our list.

Built by Consensys, Diligence Fuzzing offers a fully-fledged smart contract fuzzing as a service platform

powered by Harvey

, a powerful fuzzer for Ethereum's bytecode, which delves deep into the contract codes, mutating and testing various inputs to identify potential issues.

Price

: From 0 to $1,999

Key Features

- Harvey is skilled at analyzing Ethereum bytecode, efficiently identifying code anomalies and vulnerabilities.
- Auditors can integrate their existing Foundry

tests with Diligence Fuzzing, streamlining the auditing process and minimizing setup hassles.

- Auditors can use Scribble

to annotate contracts, highlight critical code sections, prepare the testing environment, and initiate in-depth code reviews with Diligence Fuzzing.

4. Best Rust-based Static Analyzer: Cyfrin Aderyn

Another type of smart contract security auditing tool

is “static analyzers

” - Unit and fuzz testing are known as dynamic testing. Dynamic means that you’re doing something, like actually running our code.

Smart contract static analyzers, instead, just look at our code. They don’t run it but try to find logic issues or other potential vulnerabilities

.

Cyfrin’s dedication to advancing smart contract security has createdAderyn - an open-source, Rust-based, static analyzer able to detect and report suspected vulnerabilities in smart contracts written in Solidity. The tool traverses the Abstract Syntax Trees (AST) and identifies potential issues.

Aderyn automatically analyses a smart contract’s codebase and quickly finds possible threats, reporting them in an easy-to-digest markdown format. It also allows developers to build their own detectors through Nyth,

adapting the tool to any codebase.

Price

: Free

Key Features

- Detects vulnerabilities

with low false positives

- Easy integrates intoCI/CD pipelines
- Hardhat/Foundry

support

- Average execution time of less than 1 second

per contract

- Developer framewor

k to write custom analyses in Python

5. Best Python-based Static Analyzer: Slither

Image showing the aderyn by cyfrin logo a leading smart contract static analysis auditing and security tool.

Another smart contract security tool developed by Trail of Bits

, Slither is a Python-based static analysis tool

that provides an extensive range of vulnerability detectors for Solidity code.

Its fast execution time, low false-positive rate, and ability to integrate into continuous integration (CI) pipelines make it a valuable asset for developers aiming to improve their code's security.

With more than 92 detectors

, Slither can detect a wide range of vulnerabilities with a solid trust score and the ability to fasten up auditors' efficiency, not to mention that it is compatible with a wide range of frameworks like Hardhat

, Dapp Tools, and of course, Foundry.

Price

: Free

Key Features

- Identifies where the error condition occurs in the source code
- Detects vulnerable Solidity code

with low false positives

- Built-in 'printers'

quickly report crucial contract information

- Detector API

to write custom analyses in Python

6. Best formal verification tool: Halmos

Halmos, developed by a16z, emerges as a pioneering open-source smart contract security tool offering formal verification tailored explicitly for Ethereum smart contracts

. It uniquely bridges the gap between traditional unit testing and formal specifications through its innovative use of symbolic testing.

Formal Verification requires advanced mathematical and arithmetic skills, and a writing test with Halmos requires a special setup and reading of the documentation. This rigorous approach ensures high precision in evaluating smart contracts, aiming to iron out potential flaws and guarantee flawless operation.

Price

: free

Key Features

- It is easy to use, and it is good for finding bugs.
- Halmos uses bounded symbolic execution to avoid the halting problem. Bounded symbolic execution limits the number of times a loop can be executed. This allows Halmos to explore all possible paths through a program, even if the program contains unbounded loops.
- It is constantly being improved by the developers.

7. Best smart contract DevOps tool: Foundry

[Foundry

](https://updraft.cyfrin.io/courses/foundry) gets a special mention here, one tool that should never be missed in the smart contract developer stack.

Designed for smart contract development and auditing. It simplifies tasks each task, from managing project dependencies to compiling, testing, and deploying smart contracts, as well as direct blockchain interactions and testing.

Foundry offers features like automatic compiler version detection and efficient caching, and it stands out with its fuzz testing capabilities.

Price

: free

Key Features

- Forge

: An Ethereum application testing framework that supports property-based testing.

- Cast

: Assists users in engaging with and managing smart contracts on the Ethereum blockchain.

- Anvil

: A local Ethereum node, facilitating users in application testing without relying on external networks.

- Chisel

: Solidity REPL tool, enabling users to swiftly test and execute Solidity code, enhancing the development experience.

8. Best for smart contract security research: Solodit

Not an auditing tool per se, but the best place for auditors to learn about vulnerabilities and security breaches.

Solodit, another tool from the Cyfrin ecosystem

, aggregates over 8,000 security vulnerabilities and bounties from various security firms and top researchers worldwide.

The platform aims to strengthen the security of decentralized apps and smart contracts by providing detailed reports on vulnerabilities, including:

- The nature of the vulnerability,
- The contract the vulnerability is present
- The determined severity of the issue,
- Pertinent information to understand and address the related security issue.

Solodit also offers advanced search and filtering tools to help users easily find specific vulnerabilities and bounties.

Price

: free

Key Features

- Aggregates 8000+ smart contract vulnerabilities
- Bug bounties sourced

from the best blockchain bug bounties platforms

- Auditing checklist

with step-by-step guidance on how to find smart contract vulnerabilities

- Smart contract auditing competitions aggregator to monetize your skills

Additional smart contract security auditing tools

We only took 8 of the to blockchain security and auditing tools, but the real behemoth is too big to be shown in just a single article.

Github's user @shanzson gathered an exciting resource with a lot of helpful information, links, and tools any auditor should consider using, and we highly recommend it.

Conclusion

In this list of the top smart contract auditing and security tools, we've seen 8 different tools that shouldn't miss in your toolkit!

Each tool brings unique strengths, catering to different aspects of smart contract security. Remember, the tools are just that, tools.

The most important factor for a successful audit is not the tool but the auditor. Including tools and software in your audit stack will help you perform better, find more vulnerabilities, and improve your workflows.

– If you want to learn how to become a blockchain developer or auditor and learn from the experts to write robust and reliable smart contracts, join Cyfrin Updraft now and start learning for free!