What is a smart contract audit?

A smart contract audit is a security review of blockchain (on-chain) code, usually Solidity, before it is deployed, to find the vulnerabilities that let attackers steal funds or break the protocol. It combines manual line-by-line review, economic and protocol analysis, and automated tooling against known weakness classes (reentrancy, access control, oracle manipulation, and more). Because deployed contracts are public and effectively immutable, the audit is done before launch, when fixes are still cheap. The output is a report with each finding, a proof, and a fix.

HOW IT WORKS

01 Why audits happen before launch

Two properties make on-chain code unforgiving:

- Public: the bytecode (and usually the source) is visible to everyone, so attackers can study it at leisure.
- Immutable: once deployed, a contract generally cannot be patched; fixing a bug means migrating users to a new contract, which is slow and risky.

There is also direct financial value on the line, contracts hold tokens and funds, so a single bug can be drained in one transaction. That combination is why the review happens before deployment, not after.

02 What an audit covers

A thorough audit looks at both layers:

- Code-level bugs: reentrancy, integer overflow-, access control, delegatecall, unchecked calls, and proxy issues.
- Economic and protocol attacks: flash loans-, oracle manipulation, front-running and MEV, and rug pulls.

03 How an audit is done

A credible audit blends methods rather than relying on any one:

- Manual review: experienced auditors read the code line by line, the only way to catch logic and economic flaws.

SOURCES

[1] OWASP Smart Contract Top 10

[2] Ethereum.org: Smart contract security

[3] Solidity docs: Security considerations

- Automated analysis: static analyzers, linters, and symbolic-execution tools surface known weakness patterns.
- Testing and fuzzing: property-based tests and fuzzers probe edge cases.
- Threat modeling: reasoning about incentives, who profits if they break this, and how.

The deliverable is a report grading each finding by severity, with a reproduction and a fix, followed by a re-test once fixes land.

> IMMUTABLE MEANS AUDIT-FIRST
>
> *You cannot patch most deployed contracts, and they hold real funds in public view. That is why the audit comes before launch, when a finding costs a code change instead of the treasury.*

04 What you get from an audit

A good engagement leaves your team with a clear, severity-graded report, a reproduction for every finding, concrete fixes, and a re-test confirming the fixes hold. It should cover both the code and the economics, since many of the largest losses came from economically valid but unintended interactions, not classic memory bugs.