What is smart contract security?

Smart contract security is the practice of ensuring blockchain code behaves only as intended, especially around funds, despite being public, immutable, and adversarial by default. It is different from ordinary application security because there is no patching after deployment, every input is potentially hostile and profitable, and the economics of the protocol are part of the attack surface. The main bug families are code-level flaws (reentrancy, overflow, access control) and economic attacks (flash loans, oracle manipulation), and the defense is audits, testing, and safe patterns.

HOW IT WORKS

01 Why it is different from normal appsec

Three properties make smart contract security its own discipline:

- Immutable: most deployed contracts cannot be patched, so bugs are permanent until users migrate.
- Public and adversarial: the code and state are visible, and anyone can call any function, so every path is reachable by an attacker.
- Economically exploitable: a valid sequence of transactions, not a memory-corruption bug, is often the exploit. The attacker uses the protocol exactly as written, in a way the designers did not intend.

02 The main classes of bugs

Vulnerabilities split into two families:

- Code-level: reentrancy, integer overflow, access control, delegatecall, unchecked external calls, and tx.origin auth.
- Economic and protocol: flash loans, oracle manipulation, front-running and MEV, and rug pulls.

03 How it is achieved

Smart contract security relies on layered practice:

- Audits before launch (the main control), see what is a smart contract audit.
- Safe patterns and libraries: checks-effects-interactions, reentrancy guards, vetted standard libraries, and a current compiler.

SOURCES

[1] OWASP Smart Contract Top 10

[2] Ethereum.org: Smart contract security

[3] SWC Registry: Smart Contract Weakness Classification

- Thorough testing and fuzzing, plus formal verification for critical logic.
- Operational controls: timelocks, multisig, monitoring, and a tested incident plan.

> THE EXPLOIT IS OFTEN VALID USAGE
>
> *Many of the biggest losses were not "bugs" in the classic sense, the attacker called the contract exactly as written, in an order the designers never imagined. That is why economic reasoning is core to smart contract security.*