Smart Contract Security Audits - Why Do You Need Them?

TechRounder PDF Edition

Live article: https://www.techrounder.com/technology/smart-contract-security-audits-why-do-you-need-them/

By Ankit Pahuja | Published October 15, 2021 | Updated January 4, 2026 | Format: Article | 4 min read

In brief

If your firm operates on blockchain technology, you must be familiar with the concept of smart contracts. On the other hand, smart contract security audits may not seem extensively popular due to their complexity and high resource requirements.

If your firm operates on blockchain technology, you must be familiar with the concept of smart contracts. On the other hand, smart contract security audits may not seem extensively popular due to their complexity and high resource requirements. However, smart contract security audits should be necessary for the firm's overall security strategy of protecting sensitive information against hacking attempts.

The occurrence of even one bug in smart contracts can lead to the failure of the entire structure and tremendous revenue losses, proven by historical events time and again. The three most recent examples include the 51% attack on the decentralized network PegNet (April 2020), the Lendf.Me platform that lost ETH valued at USD 25 million, and the Parity wallet (Parity 2 Hack) that lost 650,000 worth ETH (2017).

An introduction to Smart Contract Security Audits

Similar to penetration testing procedures, smart contract security audits involve the detailed examination of all components and features of the smart contract's code, its intended purpose, and analysis of interactions with other cryptocurrencies. The security audit's main purpose is to analyze security issues, hidden vulnerabilities, errors, and misconfigurations and suggest the best remediation measures.

It's important to deal with the security aspect of smart contracts because they usually deal with sensitive customer information or financial data, as was seen in the real-life examples above. Therefore, smart contract security audits are complicated because tests are conducted to detect vulnerabilities in individual smart contracts and those contracts that interact with each other, and existent integrations with other third-party software that may introduce foreign vulnerabilities into the system. This is also why smart contract security audits contain both running tests and manual code analysis to cover all security perspectives.

What kind of projects require Smart Contract Security Audits?

Any firm that uses blockchain technology will benefit from smart contract security audits, but let's have an in-depth look at the specific kinds of projects that necessarily require such security testing:

DeFi projects

Smart contracts used in DeFi projects have become increasingly complicated and can benefit from a fully encompassing security audit. DeFi, or decentralized finance, usually refers to a collection of financial applications connected via blockchain technology. Banks use this service to offer traditional banking facilities added security, privacy, and other provisions for both lenders and borrowers.

Token contracts (Crowdsales)

Smart contract security audits should also be conducted across major protocols and with the help of different programming languages such as JavaScript, C++, etc., for capturing all possible vulnerabilities in different applications. Crowdsales usually include the sale of token contracts by forming a base contract that dictates the rules and regulations. The activity is taken up to meet the financial requirements of a firm’s project, following which token providers are shareholders of the project.

Wallets (dApps)

A decentralized application, or dApp, works as a wallet and some ETH for any transaction fees. Their main feature is that they are operated and controlled by decentralized protocols such as Ethereum. They also involve complex smart contracts that need proper auditing practices and security measures to prevent financial losses.

4 Types of Smart Contract Audit services

Smart contracts can vary among the decentralized applications that use them, so it’s important to understand the unique points of each and design the security audits accordingly for the maximum discovery of vulnerabilities.

1. Full Security Audit

This covers all aspects of the smart contract, including its interaction with other smart contracts and third-party applications. First, a combination of automated and manual testing tools is used to uncover potential vulnerabilities for basic exploitation, followed by more in-depth checking. Manual testing techniques are important here. They help understand the context in which the smart contract operates and its intended purposes, which must be kept in mind before testing for security issues. Otherwise, simply using automated testing tools showcase the risk of generating ’false positives’.

2. Basic Security Audit

This type of audit takes the least time and is led by a single tester as it was designed keeping in mind the prerequisites of standard token contracts such as ERC20 and ERC721. It doesn’t go into the contract extensively and covers the basic aspects of operational needs. Firms with low involvement in blockchain technology-based applications can choose this kind of testing procedure.

3. Interim Audit

Usually used for DeFi projects, it’s majorly used to review the complexities involved in its smart contracts and ensure that the right levels of protection are implemented for customer data and their finances.

4. Round-the-clock Audit

If your project is still running on the development cycle, has a set map of milestones, and requires multiple iterations to work through its roadblocks, this kind of audit fits your requirements best. Testers will accompany your application throughout its development cycle for periodic reviews and security recommendations before moving forward.

With this, you should have a fair idea of the kind of projects that can benefit from smart contract security audits and the kinds of tests that need to be designed to fit the appropriate security requirements.

References

1. getastra.com - blog / security-audit - https://www.getastra.com/blog/security-audit/an-introduction-to-smart-contract-security/
2. getastra.com - blog / security-audit - https://www.getastra.com/blog/security-audit/penetration-testing/
3. wirexapp.com - blog / post - https://wirexapp.com/blog/post/erc20-vs-erc721-whats-the-difference-0341