EEA Primers™
Educating the Business Community About the Power of Ethereum
Version 1, March 23, 2023

Introduction to Smart Contract Security

How can smart contracts be hacked?

Smart contract vulnerabilities arise from various factors including errors in the design or implementation of the contract, the misuse of programming languages, or unforeseen interactions with other smart contracts and the underlying blockchain platform.

Background

Smart contract security is a crucial aspect of blockchain-based applications that rely on these self-executing computer programs to automate transactions and agreements. Smart contracts operate based on predetermined rules and conditions, and once deployed their code cannot be silently updated, making them resistant to fraud and tampering. However, the general immutability and transparency of smart contracts gives attackers the opportunity to look long and hard for vulnerabilities they can exploit to steal or manipulate digital assets.

A primary challenge in smart contract security is avoiding known vulnerabilities that can be exploited by attackers. Common examples of smart contract vulnerabilities include reentrancy attacks, integer overflow and underflow, and unexpected changes in contract ownership. Below are three interesting exploits that illustrate some of these types of attacks:

- **Reentrancy attack: Grim Finance, ~ USD 30m (18.12.2021).** In a reentrancy attack, the attacker exploits a vulnerability in the smart contract code that allows them to call the same function multiple times before it has completed processing. It is analogous to transferring funds from a debit card while the account balance remains unchanged. Re-entrancy attacks are well-known, but numerous contracts are successfully attacked each year for millions of $USD equivalent.

QUICK TAKEAWAYS

Smart contracts are permanently visible on the blockchain. Logic or code errors can be exploited by attackers to steal or manipulate digital assets.

Commonly exploited vulnerabilities include reentrancy, integer overflow and underflow, and poor access control allowing changes in contract ownership.

Following best practices for secure development, with rigorous testing for known vulnerabilities, is vital to prevent the risk of potentially serious hacking.

- **Price oracle manipulation, Vee Finance, USD 35m (21.09.2021).** In blockchain, an oracle provides information from outside the blockchain that a smart contract needs, for example market price data. Hackers discovered that Vee Finance relied on a single oracle, and they could distort the asset prices it showed by manipulative trading. This compromised the system, enabling them to buy and sell at the manipulated prices. (There have been many examples of this type of attack).
- **Rounding Vulnerability: Uniswap - fixed before deployment.** The Uniswap platform was carefully reviewed before deployment. This identified a potential vulnerability to rounding attacks, where swapping large sums from one token to another and then back, could exploit rounding to leak value in every transaction. This could have been exploited (and in other similar cases has) to steal very substantial sums from the platform, rapidly taking a small and barely visible slice at a time.

Prevention. To prevent hacks, it is important to apply best practices for secure development. It is vital to design the code logic carefully, consider the implications of all possible usage, and assign appropriate roles and powers in the smart contract. Smart contracts should provide monitoring, and enable responsible parties to bring about an emergency pause while ensuring that unauthorized parties cannot gain control of the contract, nor manipulate its business logic to work against its intended goals.

Independent code review (audit). A smart contract security review can identify and prevent potential exploits that can lead to financial losses and other negative consequences. This involves a thorough examination of the code to identify potential vulnerabilities. The process typically combines manual and automated analysis, including code reviews, penetration testing, and vulnerability assessments. Because a tiny change to a single line can introduce a vulnerability, it is crucial that the code deployed is the exact code reviewed, and that any changes are subsequently re-checked thoroughly.

HOW DO I FIND OUT MORE?

Read: EEA EthTrust Security Levels Specification, v1. An EEA standard for Smart Contract Security, backed by the expertise of many security experts. Requirements for a security audit, to ensure it tests for known vulnerabilities.

Read: Biggest Crypto Hacks, and Their Causes. Oleh Malanii of Hacken discusses 7 hacks that together enabled the theft of around $3B USD.

Watch: The State of Security for a Decentralized World. An EEA Webinar from 2021: a panel discussion of security in Ethereum, and how to improve it.

The EEA thanks Roman Palamarchuk (Smart Contract Auditor at Hacken) for contributing the core text of this Primer.

About the EEA

The Enterprise Ethereum Alliance (EEA) enables organizations to adopt and use Ethereum technology in their daily business operations. The EEA empowers the Ethereum ecosystem to develop new business opportunities, drive industry adoption, and learn and collaborate.

To learn more about joining the EEA, reach out to james.harsh@entethalliance.org or visit https://entethalliance.org/become-a-member/.

Follow the EEA on Facebook, Twitter, LinkedIn, and YouTube.

ENTERPRISE
ETHEREUM
ALLIANCE®

Produced in conjunction with

HACKEN