Smart Contract Security Audit: Best Practices

In the wake of Vitalik Buterin, coming in with the concept of Ethereum, smart contracts became the forefront of blockchain technology. Although smart contracts were deployed on bitcoin as well, their reachability was quite limited. Today, almost every sphere of business can look through a mirror of blockchain, allowing secure transactions in-absence of a third party.

But why should we trust these smart contracts? Are they secure enough to prevent our money from being stolen? If not, is there a way to make them reliable?

In this article, we are going to address all your queries starting from smart contracts, their working, and the need for **smart contract security audits**. Alongside, we will discuss ways to make your blockchain journey a hack-proof experience.

Let’s start then!

What is a smart contract?

The term smart contract was first used by Nick Szabo, a computer scientist and a cryptographer in 1997, long before the birth of bitcoin.

Essentially, Smart contracts are the automated digital counterpart of a physical contract. These contracts are in the form of a computer program or code stored inside a blockchain.

Smart contracts allow trustless transactions without a third party and are commonly used to facilitate transactions on the blockchain.

Being built on blockchain, smart contracts come with certain inherited features.

1. Immutability: Implying, that once a smart contract is deployed on a blockchain it can never be changed.
2. Distributed: This signifies that output is validated by everyone on the network, making tampering an impossible task.

How does a smart contract works?

Smart contracts follow a simple “if/when…then…” statement, written into code on a blockchain. A node network executes the actions on consummating the predetermined conditions. These actions could include transferring funds to the appropriate destination, trading an asset, sending notifications, and much more. The blockchain is then updated upon the completion of a transaction. Implying, that the transaction cannot be forged, and only the parties granted permission can verify the results.

What is a smart contract audit?

A smart contract audit focuses on the analysis of the source code to verify if it follows the predetermined conditions and behaves in the manner as intended by the developer.

Typically, a third party usually a top smart contract auditor is responsible for performing a smart contract audit to ensure a thorough review.

A top smart contract auditor looks for vulnerabilities like:

1. Re-Entrancy
2. Arithmetic Over/Under Flows
3. Unexpected Ether
4. Delegate call
5. Entropy Illusion
6. External Contract
7. Referencing
8. Short Address/Parameter Attack
9. Unchecked CALL Return Values
10. Denial Of Service (DOS)
11. Block Timestamp Manipulation

among others…

Since a number of bugs can hamper the security of your smart contracts, getting them audited is a good idea. Let's talk about other reasons, why should we go for a smart contract audit.

Why do we need a smart contract audit?

Smart contract hacking has become a recurring phenomenon for some time now. A few years back, DAO(decentralized autonomous organization) that intends to democratize how Ethereum projects were funded, was exploited by a hacker stealing 3.6 million ethers. The hacker realized the vulnerability of the fallback option, in the code exposed to re-entrancy. The Ethereum network had to run a hark fork separating Ethereum and Ethereum classic to recover the funds stolen.

Smart contracts are immutable once deployed, you cannot change your code once it has been placed on a blockchain.

Contrary to other software, in smart contracts we usually manipulate money. Signifying, if we make a mistake in a smart contract not only do we can't fix it after deploying but that mistake can allow hackers to steal money. So, we need to have a smart contract with no bugs left.

A top smart contract auditor puts themselves in the shoes of an attacker and tries to find vulnerabilities from an attacker's viewpoint. Making our smart contract prone to hacking after deploying on a blockchain.

Here is a list of best practices that you can maneuver to create a bug-free smart contract.

Best Practices: To make your smart contract hack-proof

1. Focus on designing a secure code

Firstly, articulate what exactly your smart contract intends to achieve and what are its features. Clear Intention is mandatory before proceeding further with the documentation of the code.

2. Do thorough code documentation

Here are a few suggestions that you should keep in mind before documenting

- Beware of the warnings for any programming language that you are working with.
- Start with creating smaller functionalities by splitting the logic of the entire code.
- Document the procedures of migration or upgrading before the deployment.
- Employ well-tested libraries
- Deploy the recommended version of the programming language compiler
- Periodically monitor your code after placing it on the blockchain.

3. Perform a smart contract audit and penetration testing

For this, you need to hire a smart contract auditor to perform a thorough verification of your code.

Pre-requisites for smart contract auditing

- Provide the location of your source code, preferably GitHub with the commit hash to be audited, and access to auditors, including any associated credentials, requirements, or terms.
- Auditors deploy both static and dynamic auditing tools along with manual auditing to perform an in-depth analysis of your code.
- Refer to the recommendations provided by the auditors and refactor your code based on the same.

4. Following a blockchain security checklist

Act in accordance with the well-researched and practically implemented checklists for the security of your smart asset.

You can follow these high-level recommendations to build a secure smart contract. https://github.com/crytic/building-secure-contracts/blob/master/development-guidelines/guidelines.md

5. Deploying automated security vulnerability scanner

Security vulnerability scanners help identify bugs in the code that can lead to security vulnerabilities and prevent a variety of attacks on your smart contract.
Considering the complex nature of blockchain and the growing number of smart contract users. Smart contract auditing has become a mandatory requirement to build a secure asset on the blockchain.

We at **ImmuneBytes** offer comprehensive smart contract auditing solutions for your applications to have a secure commencement.