



Approved

SMART CONTRACT FREE AI-BASED AUDIT

SchnitZeLcoin

Scan and check this report
was posted at Approved Github



June, 2024

Website: Approved.ltd

Table of Contents

Table of Contents	2
Disclaimer	3
Procedure	4
Terminology	5
Limitations	5
Basic Security Recommendation	5
Vulnerabilities checking	6
Security Issues	7
Conclusion for project owner	9
Approved Contact Info	10

Disclaimer

This is a comprehensive report based on our automated and manual examination of cybersecurity vulnerabilities and framework flaws of the project's smart contract.

Reading the full analysis report is essential to build your understanding of project's security level. It is crucial to take note, though we have done our best to perform this analysis and report, that you should not rely on the our research and cannot claim what it states or how we created it.

Before making any judgments, you have to conduct your own independent research.

We will discuss this in more depth in the following disclaimer - please read it fully.

DISCLAIMER: You agree to the terms of this disclaimer by reading this report or any portion thereof. Please stop reading this report and remove and delete any copies of this report that you download and/or print if you do not agree to these conditions. Scan and verify report's presence in the GitHub repository by a qr-code at the title page. This report is for non-reliability information only and does not represent investment advice. No one shall be entitled to depend on the report or its contents, and Approved and its affiliates shall not be held responsible to you or anyone else, nor shall Approved provide any guarantee or representation to any person with regard to the accuracy or integrity of the report.

Without any terms, warranties or other conditions other than as set forth in that exclusion and Approved excludes hereby all representations, warrants, conditions and other terms (including, without limitation, guarantees implied by the law of satisfactory quality, fitness for purposes and the use of reasonable care and skills).

The report is provided as "as is" and does not contain any terms and conditions. Except as legally banned, Approved disclaims all responsibility and responsibilities and no claim against Approved is made to any amount or type of loss or damages (without limitation, direct, indirect, special, punitive, consequential or pure economic loses or losses) that may be caused by you or any other person, or any damages or damages, including without limitations (whether innocent or negligent).

Security analysis is based only on the smart contracts. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Our analysis contains following steps:

1. Project Analysis;
2. Unit Testing:
 - Smart contract functions will be unit tested on multiple parameters and under multiple conditions to ensure that all paths of functions are functioning as intended.
 - In this phase intended behaviour of smart contract is verified.
 - In this phase, we would also ensure that smart contract functions are not consuming unnecessary gas.
 - Gas limits of functions will be verified in this stage.
3. Automated Testing:
 - Mythril
 - Oyente
 - Manticore
 - Solgraph
4. Testing code with artificial intelligence

Terminology

We categorize the finding into 4 categories based on their vulnerability:

- Low-severity issue — less important, must be analyzed
- Medium-severity issue — important, needs to be analyzed and fixed
- High-severity issue — important, might cause vulnerabilities, must be analyzed and fixed
- Critical-severity issue — serious bug causes, must be analyzed and fixed.

Limitations

The security audit of Smart Contract cannot cover all vulnerabilities. Even if no vulnerabilities are detected in the audit, there is no guarantee that future smart contracts are safe. Smart contracts are in most cases safeguarded against specific sorts of attacks. In order to find as many flaws as possible, we carried out a comprehensive smart contract audit. Audit is a document that is not legally binding and guarantees nothing.

Basic Security Recommendation

Unlike hardware and paper wallets, hot wallets are connected to the internet and store private keys online, which exposes them to greater risk. If a company or an individual holds significant amounts of cryptocurrency in a hot wallet, they should consider using MultiSig addresses. Wallet security is enhanced when private keys are stored in different locations and are not controlled by a single entity.

Vulnerabilities checking

Issue Description	Checking Status
Compiler Errors	Completed
Delays in Data Delivery	Completed
Re-entrancy	Completed
Transaction-Ordering Dependence	Completed
Timestamp Dependence	Completed
Shadowing State Variables	Completed
DoS with Failed Call	Completed
DoS with Block Gas Limit	Completed
Outdated Compiler Version	Completed
Assert Violation	Completed
Use of Deprecated Solidity Functions	Completed
Integer Overflow and Underflow	Completed
Function Default Visibility	Completed
Malicious Event Log	Completed
Math Accuracy	Completed
Design Logic	Completed
Fallback Function Security	Completed
Cross-function Race Conditions	Completed
Safe Zeppelin Module	Completed

Security Issues

1) Issue Type: **APPROVE FRONTRUNNING ATTACK**

Severity: **High**

```
166     function approve(address spender, uint256 amount) public virtual override returns (bool)
167     {
168         address owner = _msgSender();
169         _approve(owner, spender, amount);
170         return true;
171     }
```

```
261     function _spendAllowance(address owner, address spender, uint256 amount) internal
virtual {
262         uint256 currentAllowance = allowance(owner, spender);
263         if (currentAllowance != type(uint256).max) {
264             require(currentAllowance >= amount, "ERC20: insufficient allowance");
265             unchecked {
266                 _approve(owner, spender, currentAllowance - amount);
267             }
268         }
269     }
```

L166 L170, L261 L269

Description:

The method overrides the current allowance regardless of whether the spender already used it or not, so there is no way to increase or decrease allowance by a certain value atomically unless the token owner is a smart contract, not an account. This can be abused by a token receiver when they try to withdraw certain tokens from the sender's account. Meanwhile, if the sender decides to change the amount and sends another `approve` transaction, the receiver can notice this transaction before it's mined and can extract tokens from both transactions, therefore, ending up with tokens from both the transactions. This is a front-running attack affecting the `ERC20 Approve` function.

2) Issue Type: **USE OF FLOATING PRAGMA**

Severity: **Low**

```
3
4   pragma solidity ^0.8.18;
5
6   interface IERC20 {
```

L4

Description:

Solidity source files indicate the versions of the compiler they can be compiled with using a pragma directive at the top of the solidity file. This can either be a floating pragma or a specific compiler version. The contract was found to be using a floating pragma which is not considered safe as it can be compiled with all the versions described.

3) Issue Type: **USE OWNABLE2STEP**

Severity: **Low**

```
277 contract Schnitzelcoin is ERC20, Ownable {
278     constructor() ERC20("Schnitzelcoin", "SZL") {
279         _mint(msg.sender, 100000 * 10 ** decimals());
```

L277

Description:

Ownable2Step is safer than **Ownable** for smart contracts because the owner cannot accidentally transfer the ownership to a mistyped address. Rather than directly transferring to the new owner, the transfer only completes when the new owner accepts ownership.

Conclusion for project owner

High and Low-severity issues exist within smart contracts.

NOTE: Please check the disclaimer above and note, that the audit makes no statements or warranties on the business model, investment attractiveness, or code sustainability. Contract security report for community

Approved Contact Info

Website: <https://approved.ltd>

Telegram: @team_approved

GitHub: https://github.com/Approved-Audits/smart_contracts

