soken

SMART CONTRACT SECURITY AUDIT

DOGETTI

Scan and check this report was posted at Soken Github

March, 2023

Website: soken.io

Table of Contents

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 Soken and its affiliates shall not be held responsible to you or anyone else, nor shall Soken 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 Soken 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, Soken disclaims all responsibility and responsibilities and no claim against Soken 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. Manual analysis of smart contracts:
   - Deploying smart contracts on any of the network(Ropsten/Rinkeby) using Remix IDE
   - Hashes of all transaction will be recorded
   - Behaviour of functions and gas consumption is noted, as well.

3. 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.

4. Automated Testing:
   - Mythril
   - Oyente
   - Manticore
   - Solgraph

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.
More info: **https://blog.soken.io/how-to-gnosis-multisig-46b1386ba8e5**

Token Contract Details for 02.03.2023

*

To be added after contract deployment

*

Audit Details

Project Name: **DOGETTI**

Language: **Solidity**

Compiler Version: **v0.8.17**

Blockchain:

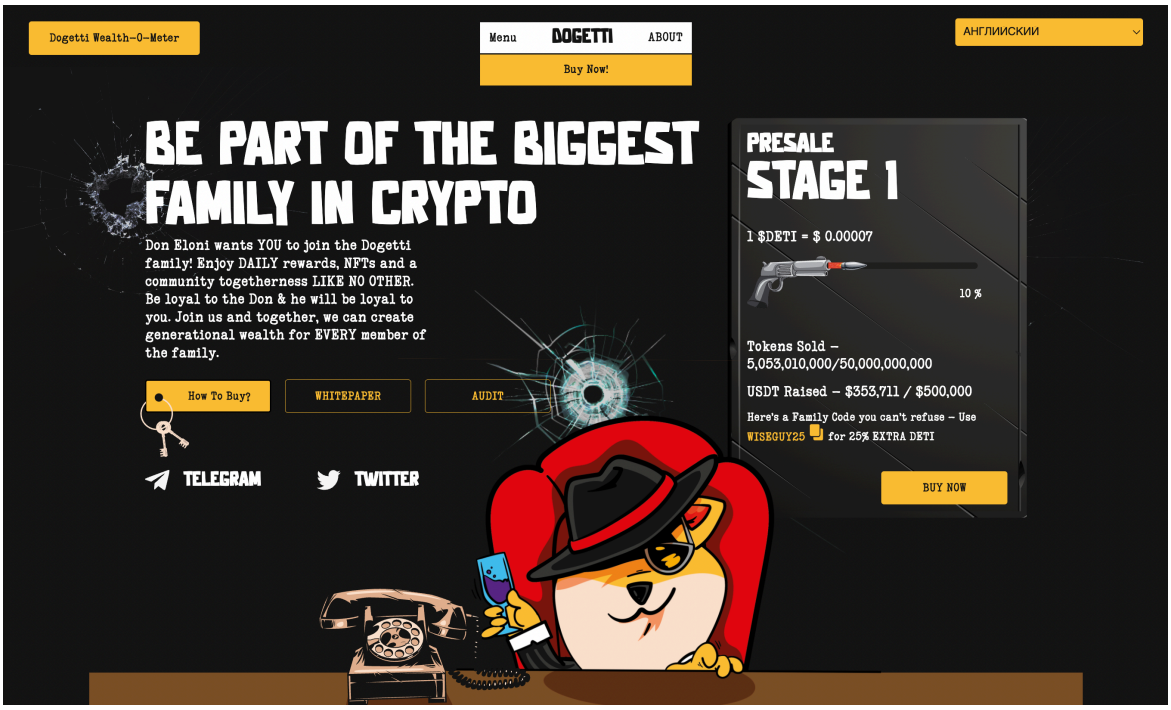
Social Profiles

Project Website: **https://www.dogetti.io/**

Project Twitter: **https://twitter.com/_Dogetti_**

Project Telegram: **https://t.me/Dogetti**

Project Website Overview



✓JavaScript errors hasn't been found.
✓Malware pop-up windows hasn't been detected.
✓No issues with loading elements, code, or stylesheets.

Project Website SSL Certification

Issued To

Common Name (CN)	fwclegends.games
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

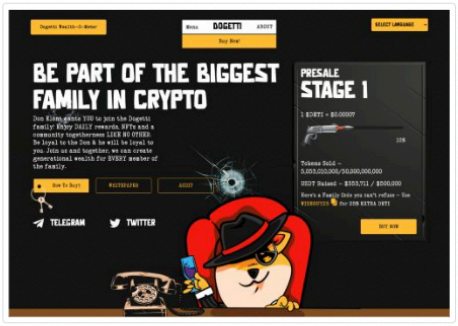
Common Name (CN)	R3
Organization (O)	Let's Encrypt
Organizational Unit (OU)	<Not Part Of Certificate>

Project Website Optimization for Desktop

Performance

Values are approximate and subject to change. The performance level is calculated directly from these metrics. Show calculator

▲ 0–49 ■ 50–89 ● 90–100



INDICATORS Expand

- First Contentful Paint 0.4 sec.
- Time to Interactive 2.8 sec.
- Speed Index 1.2 sec.
- Total Blocking Time 160ms
- Largest Contentful Paint 1.0 sec
- Cumulative Layout Shift 0.049

Project Website Optimization for Mobile

Performance

Values are approximate and subject to change. The performance level is calculated directly from these metrics. Show calculator

▲ 0–49 ■ 50–89 ● 90–100

INDICATORS

- First Contentful Paint 1.9 sec.
- Time to Interactive 14.1 sec.
- Speed Index 5.4 sec.
- Total Blocking Time 1 310 ms
- Largest Contentful Paint 3.2 sec.
- Cumulative Layout Shift 0.645

Contract Function Details

+ Dogetti.soll
- [Ext] factory
- [Ext] WETH
- [Ext] addLiquidity
- [Ext] addLiquidityETH
- [Ext] removeLiquidity
- [Ext] removeLiquidityETH
- [Ext] removeLiquidityWithPermit
- [Ext] removeLiquidityETHWithPermit
- [Ext] swapExactTokensForTokens
- [Ext] swapTokensForExactTokens
- [Ext] swapExactETHForTokens
- [Ext] swapTokensForExactETH
- [Ext] swapExactTokensForETH
- [Ext] swapETHForExactTokens
- [Ext] quote
- [Ext] getAmountOut
- [Ext] getAmountIn
- [Ext] getAmountsOut
- [Ext] getAmountsIn
- [Ext] removeLiquidityETHSupportingFeeOnTransferTokens
- [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens
- [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens
- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens
- [Ext] feeTo
- [Ext] feeToSetter
- [Ext] getPair
- [Ext] allPairs
- [Ext] allPairsLength
- [Ext] createPair
- [Ext] setFeeTo
- [Ext] setFeeToSetter
- [Ext] INIT_CODE_PAIR_HASH
- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens
- [Ext] updatePancakePair
- [Pub] updatePancakeRouter
- [Prv] _tokenTransfer
- [Prv] _transferStandard
- [Prv] _transferToExcluded
- [Prv] _transferFromExcluded
- [Prv] _transferBothExcluded

- [Prv] _reflectFee
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] removeAllFee
- [Prv] restoreAllFee
- [Prv] _takeLiquidity
- [Ext] name
- [Ext] symbol
- [Ext] decimals
- [Ext] totalSupply
- [Pub] balanceOf
- [Ext] transfer
- [Ext] allowance
- [Pub] approve
- [Ext] transferFrom
- [Ext] increaseAllowance
- [Ext] decreaseAllowance
- [Ext] isExcludedFromReward
- [Ext] totalFees
- [Ext] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward
- [Pub] includeInReward
- [Prv] _approve
- [Ext] updateLiquidityFee
- [Ext] updateCharityFee
- [Ext] updateRewardFee
- [Ext] updateCharityWallet
- [Ext] updateMinAmountToTakeFee
- [Pub] setAutomatedMarketMakerPair
- [Prv] _setAutomatedMarketMakerPair
- [Ext] excludeFromFee
- [Prv] _transfer
- [Prv] takeFee
- [Prv] swapTokensForBaseToken
- [Prv] addLiquidity

+ PancakeCaller.sol
- [Ext] factory
- [Ext] WETH
- [Ext] addLiquidity
- [Ext] addLiquidityETH
- [Ext] removeLiquidity

- [Ext] removeLiquidityETH
- [Ext] removeLiquidityWithPermit
- [Ext] removeLiquidityETHWithPermit
- [Ext] swapExactTokensForTokens
- [Ext] swapTokensForExactTokens
- [Ext] swapExactETHForTokens
- [Ext] swapTokensForExactETH
- [Ext] swapExactTokensForETH
- [Ext] swapETHForExactTokens
- [Ext] quote
- [Ext] getAmountOut
- [Ext] getAmountIn
- [Ext] getAmountsOut
- [Ext] getAmountsIn
- [Ext] removeLiquidityETHSupportingFeeOnTransferTokens
- [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens
- [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens
- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens
- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens
- [Ext] swapExactTokensForTokens

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 Complier 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) Reentrancy risk: High-severity.

PancakeCaller.sol: Line#215-235, 237-243.

In a Re-entrancy attack, a malicious contract calls back into the calling contract before the first invocation of the function is finished. This may cause the different invocations of the function to interact in undesirable ways, especially in cases where the function is updating state variables after the external calls. This may lead to loss of funds, improper value updates, token loss, etc.

Recommendation:

It is recommended to add a [Re-entrancy Guard] (https://docs.openzeppelin.com/contracts/4.x/api/security#ReentrancyGuard) to the functions making external calls. The functions should use a Checks-Effects-Interactions pattern. The external calls should be executed at the end of the function and all the state-changing must happen before the call.

2) Use of Floating Pragma: Informational.

PancakeCaller.sol: Line# 2

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.

The following affected files were found to be using floating pragma: / PancakeCaller.sol - ^0.8.17

Recommendation:

It is recommended to use a fixed pragma version, as future compiler versions may handle certain language constructions in a way the developer did not foresee. Using a floating pragma may introduce several vulnerabilities if compiled with an older version. The developers should always use the exact Solidity compiler version when designing their contracts as it may break the changes in the future. Instead of ^0.8.17 use pragma solidity 0.8.7, which is a stable and recommended version right now.

Conclusion for project owner

High and informational-severity issues exist within smart contracts.

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

SECURITY REPORT
FOR COMMUNITY

Dogetti

soken

Whitepaper of the project

The whitepaper of DOGETTI project has been verified on behalf of Soken team.

Whitepaper link: **https://www.dogetti.io/whitepaper.pdf**

Soken Contact Info

Website: www.soken.io

Mob: (+1)416-875-4174

32 Britain Street, Toronto, Ontario, Canada

Telegram: @team_soken

GitHub: sokenteam

Twitter: @soken_team