



# SOLIDProof

*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC  
Development | Marketing**

MADE IN GERMANY

# LuigiSwap

# AUDIT

SECURITY ASSESSMENT

**18. October, 2023**

FOR



**LUIGISWAP**



**SolidProof\_io**



**@solidproof\_io**



Introduction	3
Disclaimer	3
Project Overview	4
Summary	4
Social Medias	4
Audit Summary	5
File Overview	6
Imported packages	8
Components	9
Exposed Functions	9
Capabilities	10
Inheritance Graph	11
Audit Information	12
Vulnerability and Risk Levels	12
Auditing Strategy and Techniques Applied	13
Methdology	13
Overall Security	14
Upgradeability	14
Ownership	15
Ownership Privileges	16
Minting Tokens	16
Burning Tokens	17
Blacklist Addresses	18
Fees and Tax	19
Lock User Funds	20
Centralization Privileges	21
Audit results	23

## Introduction

[SolidProof.io](#) is a brand of the officially registered company MAKE Network GmbH, based in Germany. We're mainly focused on Blockchain Security such as Smart Contract Audits and KYC verification for project teams.

Solidproof.io assess potential security issues in the smart contracts implementations, review for potential inconsistencies between the code base and the whitepaper/documentation, and provide suggestions for improvement.

## Disclaimer

[SolidProof.io](#) reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc'...)

**SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.**

SolidProof.io Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. SolidProof's position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of the security or functionality of the technology we agree to analyze.

# Project Overview

## Summary

<b>Project Name</b>	<b>LuigiSwap</b>
<b>Website</b>	<a href="https://luigiswap.finance/">https://luigiswap.finance/</a>
<b>About the project</b>	LuigiSwap is your entry into decentralized finance (DeFi) powered by opBNB. It's a peer-to-peer system that makes cryptocurrency trading on the opBNB smooth and secure. No intermediary villains that might limit access.
<b>Chain</b>	TBA
<b>Language</b>	Solidity
<b>Codebase</b>	<a href="https://github.com/LuigiSwap/LuigiSwap-Contracts/">https://github.com/LuigiSwap/LuigiSwap-Contracts/</a>
<b>Commit</b>	N/A
<b>Forked Status</b>	This project is 1:1 forked from Uniswap and pancake swap, Here are the links: v2-core: <a href="https://github.com/Uniswap/v2-core">https://github.com/Uniswap/v2-core</a> v2-periphery: <a href="https://github.com/Uniswap/v2-periphery/tree/master/contracts">https://github.com/Uniswap/v2-periphery/tree/master/contracts</a> Libraries: <a href="https://github.com/Uniswap/solidity-lib/tree/master/contracts/libraries">https://github.com/Uniswap/solidity-lib/tree/master/contracts/libraries</a>
<b>Unit Tests</b>	Not Provided

## Social Medias

<b>Telegram</b>	N/A
<b>Twitter</b>	<a href="https://twitter.com/LuigiSwap">https://twitter.com/LuigiSwap</a>
<b>Facebook</b>	N/A
<b>Instagram</b>	N/A
<b>GitHub</b>	N/A
<b>Reddit</b>	N/A
<b>Medium</b>	N/A
<b>Discord</b>	<a href="https://discord.gg/hCbXdxpnQe">https://discord.gg/hCbXdxpnQe</a>
<b>YouTube</b>	N/A
<b>TikTok</b>	N/A
<b>LinkedIn</b>	N/A

## Audit Summary

Version	Delivery Date	Change Log
v1.0	18. October 2023	<ul style="list-style-type: none"> <li>· Layout Project</li> <li>· Automated/ Manual-Security Testing</li> <li>· Summary</li> </ul>

**Note** – The following audit report presents a comprehensive security analysis of the smart contract utilized in the project that includes outside manipulation of the contract's functions in a malicious way. This analysis did not include functional testing (or unit testing) of the contract/s logic. We cannot guarantee 100% logical correctness of the contract as we did not functionally test it. This includes internal calculations in the formulae used in the contract.



## File Overview

The Team provided us with the files that should be tested in the security assessment. This audit covered the following files listed below with an SHA-1 Hash.

File Name	SHA-1 Hash
contracts/AMM/Router/contracts/interfaces/ILuigiswapV2Router01.sol	4d407fc383565a0e3f85256437c59c dbf3a5e86a
contracts/AMM/Router/contracts/interfaces/ILuigiswapV2Router02.sol	ebbc7b3dadee43dbb65f705c8255b 689df474ef4
contracts/AMM/Router/contracts/interfaces/IWETH.sol	dc895c5734dd3ed83f324f103b0ab5 a7705f5d9e
contracts/AMM/Router/contracts/interfaces/ILuigiswapV2Migrator.sol	9ef09f6fb629a4a788d0d3cc554d2c 312a9da2ce
contracts/AMM/Router/contracts/interfaces/ILuigiswapV2Pair.sol	e8971603dee4e6000a21d92048b32 3fef68046cd
contracts/AMM/Router/contracts/interfaces/IERC20.sol	b894d83bab70c089ac91499ff60fc 5aafec26a9
contracts/AMM/Router/contracts/interfaces/V1/ILuigiswapV1Factory.sol	36c263ae725a3bddad8ad09366cfe 2e1d5ff7799
contracts/AMM/Router/contracts/interfaces/V1/ILuigiswapV1Exchange.sol	de7ca26ae07db3152d0cd484efc0e 918fe195ca8
contracts/AMM/Router/contracts/interfaces/ILuigiswapV2Factory.sol	6947c0ee6f8223d2b74e813218ea28 89ce54dcd3
contracts/AMM/Router/contracts/LuigiswapV2Migrator.sol	db288d612f322425ff8047b1fd43cfd 007161c89
contracts/AMM/Router/contracts/WETH.sol	a4c558736ed65b29a3e3614d08189 cf6ca9f045c
contracts/AMM/Router/contracts/LuigiswapV2Router01.sol	fd19c7b4a8bee8213ec99288051680 bac3d0e832
contracts/AMM/Router/contracts/libraries/FullMath.sol	20ce5b4e756f9e0f4f90a99d1b5636 919da96b29
contracts/AMM/Router/contracts/libraries/LuigiswapV2LiquididityMathLibrary.sol	0e1da7b5b24f2a7dbeefdad6c7946 b30d4b44111
contracts/AMM/Router/contracts/libraries/TransferHelper.sol	376e11b5effab944e498827f8248293 8ca7860ed
contracts/AMM/Router/contracts/libraries/LuigiswapV2Library.sol	ce927d41a991ec51574058a0d50604 68eb216782



File Name	SHA-1 Hash
contracts/AMM/Router/contracts/libraries/LuigiswapV2OracleLibrary.sol	f6e000d9d780ba5a8e458fa5c5917c49f923c9cb
contracts/AMM/Router/contracts/libraries/Babylonian.sol	45121ffd37d4ef298f90dfb9ea903da19b1afe3b
contracts/AMM/Router/contracts/libraries/SafeMath.sol	434532d4628b835e129b6f3f048a32deee284cae
contracts/AMM/Router/contracts/libraries/FixedPoint.sol	53e38a7d5fea39037738c3b66beb58865bd6d586
contracts/AMM/Router/contracts/libraries/BitMath.sol	bc33abbfe11423bf26778a71dbad2fe9d38d2bf8
contracts/AMM/Router/contracts/LuigiswapV2Router02.sol	81f797cf482d1fa6fa686c9d149d16af259810e2
contracts/AMM/Router/contracts/Multicall.sol	08886ed657dd1dfbdf9e2ce06c6bd2402fe157ef
contracts/AMM/Factory/contracts/interfaces/ILuigiswapV2Pair.sol	e8971603dee4e6000a21d92048b323fef68046cd
contracts/AMM/Factory/contracts/interfaces/ILuigiswapV2Callee.sol	3b858603d58db8ad41bd3c245ba016835d82619b
contracts/AMM/Factory/contracts/interfaces/ILuigiswapV2ERC20.sol	4ea7cd6d03142b65082d3d766926f9348d1d4eff
contracts/AMM/Factory/contracts/interfaces/IERC20.sol	b894d83bab70c089ac91499ff60fc5aafec26a9
contracts/AMM/Factory/contracts/interfaces/ILuigiswapV2Factory.sol	6947c0ee6f8223d2b74e813218ea2889ce54dcd3
contracts/AMM/Factory/contracts/LuigiswapV2Factory.sol	94d84c0a43030c23da58a7094f76fd3deb90e812
contracts/AMM/Factory/contracts/LuigiswapV2ERC20.sol	9dd25bdc725a44b22f75b88d6426706ad3656eb5
contracts/AMM/Factory/contracts/libraries/Math.sol	1348a437cf092bb525e65d5b825ce19e5024ca26
contracts/AMM/Factory/contracts/libraries/UQ112x112.sol	cbb25b70152d23f5845542403004f2cccb992c29
contracts/AMM/Factory/contracts/libraries/SafeMath.sol	074e21997d7ba23f6fc85ae4fe265dafc884f188
contracts/AMM/Factory/contracts/LuigiswapV2Pair.sol	573b73c6df864f7cce275581987c7010a583d224



File Name	SHA-1 Hash
contracts/LUIGI Token/Lugiswap.sol	fc89cc96dc6e4e9d3e181ca8675c4e203ee46f56
contracts/Farm/contracts/MasterChef.sol	d7d038c3460fef1eecb78b96ed93eb77d8e9c81b
contracts/Farm/contracts/IToken.sol	b97bb833162f87cb9860ffda65a4c8dc1723aab

*Please note: Files with a different hash value than in this table have been modified after the security check, either intentionally or unintentionally. A different hash value may (but need not) be an indication of a changed state or potential vulnerability that was not the subject of this scan.*

## Imported packages.

*Used code from other Frameworks/Smart Contracts.*

Dependency / Import Path	Count
@openzeppelin/contracts/access/Ownable.sol	1
@openzeppelin/contracts/security/ReentrancyGuard.sol	1
@openzeppelin/contracts/token/ERC20/ERC20.sol	2
@openzeppelin/contracts/token/ERC20/IERC20.sol	1
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	1
@openzeppelin/contracts/token/ERC721/IERC721.sol	1
@openzeppelin/contracts/utils/math/SafeMath.sol	1

**Note for Investors:** We only audited contracts mentioned in the scope above. All contracts related to the project apart from that are not a part of the audit, and we cannot comment on its security and are not responsible for it in any way.





## External/Public functions

External/public functions are functions that can be called from outside of a contract, i.e., they can be accessed by other contracts or external accounts on the blockchain. These functions are specified using the function declaration's external or public visibility modifier.

## State variables

State variables are variables that are stored on the blockchain as part of the contract's state. They are declared at the contract level and can be accessed and modified by any function within the contract. State variables can be needed within visibility modifier, such as public, private or internal, which determines the access level of the variable.

## Components

 <b>Contracts</b>	 <b>Libraries</b>	 <b>Interfaces</b>	 <b>Abstract</b>
10	12	15	0

## Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

 <b>Public</b>	 <b>Payable</b>
230	17











<b>External</b>	<b>Internal</b>	<b>Private</b>	<b>Pure</b>	<b>View</b>
186	186	8	60	79

## StateVariables

<b>Total</b>	 <b>Public</b>
53	43



## Capabilities

Solidity Versions observed	 Experimental Features	 Can Receive Funds	 Uses Assembly	 Has Destroyable Contracts	
<div>&gt;=0.6.2</div> <div>&gt;=0.5.0</div> <div>=0.6.6</div> <div>&gt;=0.4.0</div> <div>&gt;=0.6.0</div> <div>=0.5.16</div> <div>^0.8.0</div>	<div>ABIEncoderV2</div>	<div>yes</div>	<div>yes</div> <div>(2 asm blocks)</div>	<div>-----</div>	
 Transfers ETH	 Low-Level Calls	 DelegateCall	 Uses Hash Functions	 ECREcover	 New/Create/Create2
<div>Yes</div>			<div>Yes</div>	<div>Yes</div>	<div>yes</div> <div>→ AssemblyCal</div> <div>1:Name:create2</div>

An inheritance graph is a graphical representation of the inheritance hierarchy among contracts. In object-oriented programming, inheritance is a mechanism that allows one class (or contract, in the case of Solidity) to inherit properties and methods from another class. It shows the relationships between different contracts and how they are related to each other through inheritance.



## Audit Information

### Vulnerability & Risk Level

Risk represents the probability that a certain source threat will exploit the vulnerability and the impact of that event on the organization or system. The risk level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
<b>Critical</b>	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
<b>High</b>	7 - 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
<b>Medium</b>	4 - 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
<b>Low</b>	2 - 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
<b>Informational</b>	0 - 1.9	A vulnerability that have informational character but is not affecting any of the code.	An observation that does not determine a level of risk



## Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to check the repository for security-related issues, code quality, and compliance with specifications and best practices. To this end, our team of experienced pen-testers and smart contract developers reviewed the code line by line and documented any issues discovered.

We check every file manually. We use automated tools only so that they help us achieve faster and better results.

## Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
  - a. Reviewing the specifications, sources, and instructions provided to SolidProof to ensure we understand the size, scope, and functionality of the smart contract.
  - b. Manual review of the code, i.e., reading the source code line by line to identify potential vulnerabilities.
  - c. Comparison to the specification, i.e., verifying that the code does what is described in the specifications, sources, and instructions provided to SolidProof.
2. Testing and automated analysis that includes the following:
  - a. Test coverage analysis determines whether test cases cover code and how much code is executed when those test cases are executed.
  - b. Symbolic execution, which is analysing a program to determine what inputs cause each part of a program to execute.
3. Review best practices, i.e., review smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on best practices, recommendations, and research from industry and academia.
4. Concrete, itemized and actionable recommendations to help you secure your smart contracts.



## Overall Security Upgradeability

### Contract is not an upgradable



**Deployer cannot update the contract with new functionalities.**

Description	The contract is not an upgradeable contract. The Deployer is not able to change or add any functionalities to the contract after deploying.
Comment	N/A





## Ownership

**The ownership is not renounced.**

**✗ The ownership is not renounced.**

Description

The owner has not renounced the ownership that means that the owner retains control over the contract's operations, including the ability to execute functions that may impact the contract's users or stakeholders. This can lead to several potential issues, including:

- Centralizations
- The owner has significant control over contract's operations.

Example	N/A
Comment	N/A

**Note** – *The contract cannot be considered as renounced till it is not deployed or having some functionality that can change the state of the contract.*



## Ownership Privileges

*These functions can be dangerous. Please note that abuse can lead to financial loss. We have a guide where you can learn more about these Functions.*

### Minting tokens

*Minting tokens refer to the process of creating new tokens in a cryptocurrency or blockchain network. This process is typically performed by the project's owner or designated authority, who has the ability to add new tokens to the network's total supply.*

**Contract owner cannot mint new tokens.**

 **The owner cannot mint new tokens.**

Description	The owner is able to mint new tokens once the contract is deployed but it cannot be more than total supply.
Comment	N/A





## Burning tokens

*Burning tokens is the process of permanently destroying a certain number of tokens, reducing the total supply of a cryptocurrency or token. This is usually done to increase the value of the remaining tokens, as the reduced supply can create scarcity and potentially drive up demand.*

### Contract owner cannot burn tokens

 **The owner cannot burn tokens.**

Description	The owner is not able burn tokens without any allowances.
Comment	N/A



## Blacklist addresses

*Blacklisting addresses in smart contracts is the process of adding a certain address to a blacklist, effectively preventing them from accessing or participating in certain functionalities or transactions within the contract. This can be useful in preventing fraudulent or malicious activities, such as hacking attempts or money laundering.*

### Contract owner cannot blacklist addresses.

 **The owner cannot blacklist wallets.**

Description	The owner cannot blacklist addresses for transferring of tokens.
Comment	N/A



## Fees and Tax

*In some smart contracts, the owner or creator of the contract can set fees for certain actions or operations within the contract. These fees can be used to cover the cost of running the contract, such as paying for gas fees or compensating the contract's owner for their time and effort in developing and maintaining the contract.*

**Contract owner cannot set fees more than 25%.**



**The owner cannot set fees more than 25%.**

Description

The owner cannot set fees more than 25%.

Comment

N/A

## Lock User Funds

*In a smart contract, locking refers to the process of restricting access to certain tokens or assets for a specified period of time. When token or assets are locked in a smart contract, they cannot be transferred or used until the lock-up period has expired or certain conditions have been met.*

**The owner can lock functions.**

**✗ The owner can lock functions.**

Description

The owner can lock withdraw and deposit functionality in the contract.

Comment

The owner can lock withdraw and deposit functionality in the contract for an unlimited period of time as this can lock user funds as well which is not recommended. Add a functionality where the Locking should only be done for a certain period of time. The user can use emergency withdraw tokens to withdraw their tokens even if the withdrawal is locked but then the user will not receive any rewards.

**File/Line(s): L235-237**

**Codebase: MasterChef.sol**

```
fttrace | funcSig
function setPause(bool _paused) external onlyOwner {
    paused = _paused;
}
```

## Centralization Privileges

*Centralization can arise when one or more parties have privileged access or control over the contract's functionality, data, or decision-making. This can occur, for example, if the contract is controlled by a single entity or if certain participants have special permissions or abilities that others do not.*

In the project, there are authorities that have access to the following functions:

File	Privileges
<b>Lugiswap.sol</b>	<ul style="list-style-type: none"> <li>➤ The minter can mint tokens after the initial deployment not more than the total supply.</li> <li>➤ The minter can update the minter address.</li> </ul>
<b>MasterChef.sol</b>	<ul style="list-style-type: none"> <li>➤ The owner can add a new liquidity pool to the contract.</li> <li>➤ The owner can update the allocation point for the pool.</li> <li>➤ The owner can update any arbitrary value in the token per block.</li> <li>➤ The owner can pause/unpause the deposit and withdraw functionality in the contract for an unlimited period of time.</li> <li>➤ The owner can update any arbitrary value in the startblock in this contract.</li> </ul>

## Recommendations

To avoid potential hacking risks, it is advisable for the client to manage the private key of the privileged account with care. Additionally, we recommend enhancing the security practices of centralized privileges or roles in the protocol through a decentralized mechanism or smart-contract-based accounts, such as multi-signature wallets.

Here are some suggestions of what the client can do:

- Consider using multi-signature wallets: Multi-signature wallets require multiple parties to sign off on a transaction before it can be executed, providing an extra layer of security e.g. Gnosis Safe
- Use of a timelock at least with a latency of e.g. 48-72 hours for awareness of privileged operations
- Introduce a DAO/Governance/Voting module to increase transparency and user involvement



- Consider Renouncing the ownership so that the owner cannot modify any state variables of the contract anymore. Make sure to set up everything before renouncing.



## Audit Result

### Critical Issues

No critical issues

### High Issues

No high issues

### Medium Issue

#### #1 | The owner can lock functions.

File	Severity	Location	Status
MasterChef.sol	Medium	L235-237	ACK

**Description** – The owner can lock withdraw and deposit functionality in the contract for an unlimited period of time as this can lock user funds as well which is not recommended. Add a functionality where the Locking should only be done for a certain period of time.

**Alleviation** – The "pause" function only affects the rewards in case of adverse situations, while users can still withdraw their funds using the "emergencyWithdraw" function. The initial design of the "pause" function was intended to safeguard the project, ensuring that if there were issues with the token, it wouldn't experience hyperinflation.

### Low Issue

#### #1 | Floating pragma solidity version.

File	Severity	Location	Status
Lugiswap.sol	Low	L3	Open
MasterChef.sol	Low	L2	Open

**Description** – Adding the constant version of solidity is recommended, as this prevents the unintentional deployment of a contract with an outdated compiler that contains unresolved bugs.

## #2 | Missing events arithmetic.

File	Severity	Location	Status
Lugiswap.sol	Low	L29-33	Open
MasterChef.sol	Low	L240-242	Open

**Description** – Emit all the critical parameter changes.

## Informational Issue

---

### #1 | NatSpec Documentation missing.

File	Severity	Location	Status
Lugiswap.sol	Informational	--	Open
MasterChef.sol	Informational	--	Open

**Description** – If you started to comment on your code, also comment on all other functions, variables, etc.





## Legend for the Issue Status

Attribute or Symbol	Meaning
<b>Open</b>	The issue is not fixed by the project team.
<b>Fixed</b>	The issue is fixed by the project team.
<b>Acknowledged(ACK)</b>	The issue has been acknowledged or declared as part of business logic.





**Blockchain Security | Smart Contract Audits | KYC  
Development | Marketing**

MADE IN GERMANY