Cyberscope

Audit Report

PancakeRouter

February 2023

SHA256 49f8aca785440e35e834f02d21ddd8701fc3df5f4425c9aab33311530962affa

Audited by © cyberscope

Table of Contents

Table of Contents 1
Review 1
Audit Updates 1
Source Files 1
Introduction 4
PancakeRouter 4
Roles 4
Diagnostics 5
L01 - Public Function could be Declared External 5
Description 6
Recommendation 8
L04 - Conformance to Solidity Naming Conventions 8
Description 8
Recommendation 9
L14 - Uninitialized Variables in Local Scope 9
Description 9
Recommendation 9
L16 - Validate Variable Setters 9
Description 10
Recommendation 11
L20 - Succeeded Transfer Check 11
Description 12
Recommendation 12
Functions Analysis 12
Inheritance Graph 16
Flow Graph 16
Summary 16
Disclaimer 16
About Cyberscope 21

Review

Audit Updates

Initial Audit	06 Mar 2023

Source Files

Filename	SHA256
interfaces/IERC20.sol	9c75cbedd4aa49570bfc4ca4a8da250adb1e1e6158ad2c2c5a230ce218adc033
interfaces/IPancakeCallee.sol	a95cc49d2a108030491f500dcfaa196926a28915ee8ec3bce7ddc2a823e033ec
interfaces/IPancakeERC20.sol	92647340818c895d5b716b97cf6a022694347309ea5934787a398e104ed1d441
interfaces/IPancakeFactory.sol	18ff5ffb0e39fca37091ed77356e964fb42dc6b1f699f6190eaa797dd7b7a23c
interfaces/IPancakeMigrator.sol	e3241d632b4599c3c02bb42212294f85feea78b3d5f5fffba5e7f0950ae9c764
interfaces/IPancakePair.sol	3411df2a3f50c805a90e84ed978a65bbce73a06938f174fc65670dd0628d6534
interfaces/IPancakeRouter01.sol	49bc7f8b099d3acd5680eef9cbd8d3fa681bbd99a32150fde4a0caedd07ef7b2
interfaces/IPancakeRouter02.sol	58aa9e0b66706c39012a797fc5744f64319e7588d323f87a634e4b17e9ea8059

interfaces/IWETH.sol	893803239b1e6893c19aff681e254fe798800ba1e22b543a12b832a1a527051d
libraries/Babylonian.sol	3c4e00535941e39acabfb4b0dc729a6420e34535b1d01657b326ce459fcbee50
libraries/Math.sol	68728e7cd44650b0f823189d89d1febec1b099982dac3edfa6b5745d08d4750e
libraries/PancakeLibrary.sol	a5ebaa2763464728a67f8238d52f5b7910927bb5db7d05a51b67d2e238a9a0f0
libraries/SafeMath.sol	7d1ba5983aed2d4b7598fd04c07e229729b4d5f543b657c5589d3f3bf796baa2
libraries/UQ112x112.sol	b1595a03b3f9f00282b14f3967b26f6463c8e4a40fea1b97c725f222aefffc9e
libraries/WBNB.sol	cd8fd86a1d54512921a653c04ffa824d9ac7c5b855c3487fea9ee9df9ad91d01
PancakeRouter.sol	49f8aca785440e35e834f02d21ddd8701fc3df5f4425c9aab33311530962affa
PancakeRouter01.sol	178c7afad893507567e03df32aefb6737a270b3209ea94a3bcb1d1241824ac28

Introduction

This audit is focused on the PancakeRouter contract. The PancakeRouter contract is forked from Pancake Swap. It implements the same functionality as the PancakeRouter contract.

PancakeRouter

The PancakeRouter contract facilitates trades and swaps tokens on a DEX. It uses an Automated Market Maker algorithm and liquidity pools to determine token prices and enables users to provide liquidity and earn rewards.

Roles

The PancakeRouter contract does not have Roles.

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	L01	Public Function could be Declared External	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L16	Validate Variable Setters	Unresolved
●	L20	Succeeded Transfer Check	Unresolved

L01 - Public Function could be Declared External

Criticality	Minor / Informative
Location	PancakeRouter01.sol#L274,278 PancakeRouter.sol#L479,489
Status	Unresolved

Description

A public function is a function that can be called from external contracts or from within the contract itself. An external function is a function that can only be called from external contracts, and cannot be called from within the contract itself.

It's generally a good idea to declare functions as external if they are only intended to be called from external contracts, as this can help to make the contract's code easier to understand and maintain. Declaring a function as external can also help to improve the contract's performance and gas consumption.

```
function getAmountsOut(uint amountIn, address[] memory path) public
view override returns (uint[] memory amounts) {
        return PancakeLibrary.getAmountsOut(factory, amountIn, path);
    }

function getAmountsIn(uint amountOut, address[] memory path) public
view override returns (uint[] memory amounts) {
        return PancakeLibrary.getAmountsIn(factory, amountOut, path);
    }

...
```

Recommendation

It's important to choose the appropriate visibility for each function based on how it is intended to be used. Declaring a function as external when it should be public, or vice versa can lead to unnecessary gas consumption.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	PancakeRouter01.sol#L14 PancakeRouter.sol#L17 interfaces/IPancakeRouter01.sol#L7
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
address public immutable override WETH
function WETH() external pure returns (address);
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

Find more information on the Solidity documentation
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

L14 - Uninitialized Variables in Local Scope

Criticality	Minor / Informative
Location	PancakeRouter01.sol#L171 PancakeRouter.sol#L251,373 libraries/PancakeLibrary.sol#L98
Status	Unresolved

Description

Using an uninitialized local variable can lead to unpredictable behavior and potentially cause errors in the contract. It's important to always initialize local variables with appropriate values before using them.

```
uint i
uint256 i
```

Recommendation

By initializing local variables before using them, the contract ensures that the functions behave as expected and avoid potential issues.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	PancakeRouter01.sol#L22,23 PancakeRouter.sol#L25,26
Status	Unresolved

Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
factory = _factory
WETH = _WETH
```

Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

L20 - Succeeded Transfer Check

Criticality	Minor / Informative
Location	PancakeRouter01.sol#L110 PancakeRouter.sol#L137
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IPancakePair(pair).transferFrom(msg.sender, pair, liquidity)
```

Recommendation

The contract should check if the result of the transfer methods is successful. The team is advised to check the SafeERC20 library from the Openzeppelin library.

Functions Analysis

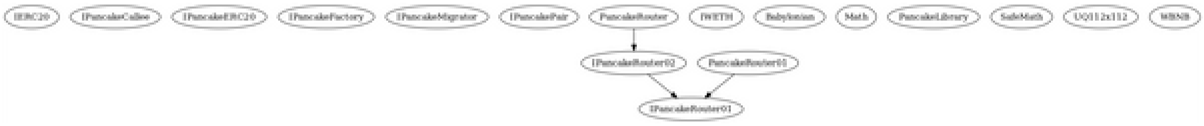
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IPancakeRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
PancakeLibrary	Library			
	sortTokens	Internal		

	pairFor	Internal		
	getReserves	Internal		
	quote	Internal		
	getAmountOut	Internal		
	getAmountIn	Internal		
	getAmountsOut	Internal		
	getAmountsIn	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	mul	Internal		
PancakeRouter	Implementation	IPancakeRouter02		
		Public	✓	-
		External	Payable	-
	_addLiquidity	Internal	✓	
	addLiquidity	External	✓	ensure
	addLiquidityETH	External	Payable	ensure
	removeLiquidity	Public	✓	ensure
	removeLiquidityETH	Public	✓	ensure
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	removeLiquidityETHSupportingFeeOnTransferTokens	Public	✓	ensure
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	_swap	Internal	✓	
	swapExactTokensForTokens	External	✓	ensure
	swapTokensForExactTokens	External	✓	ensure

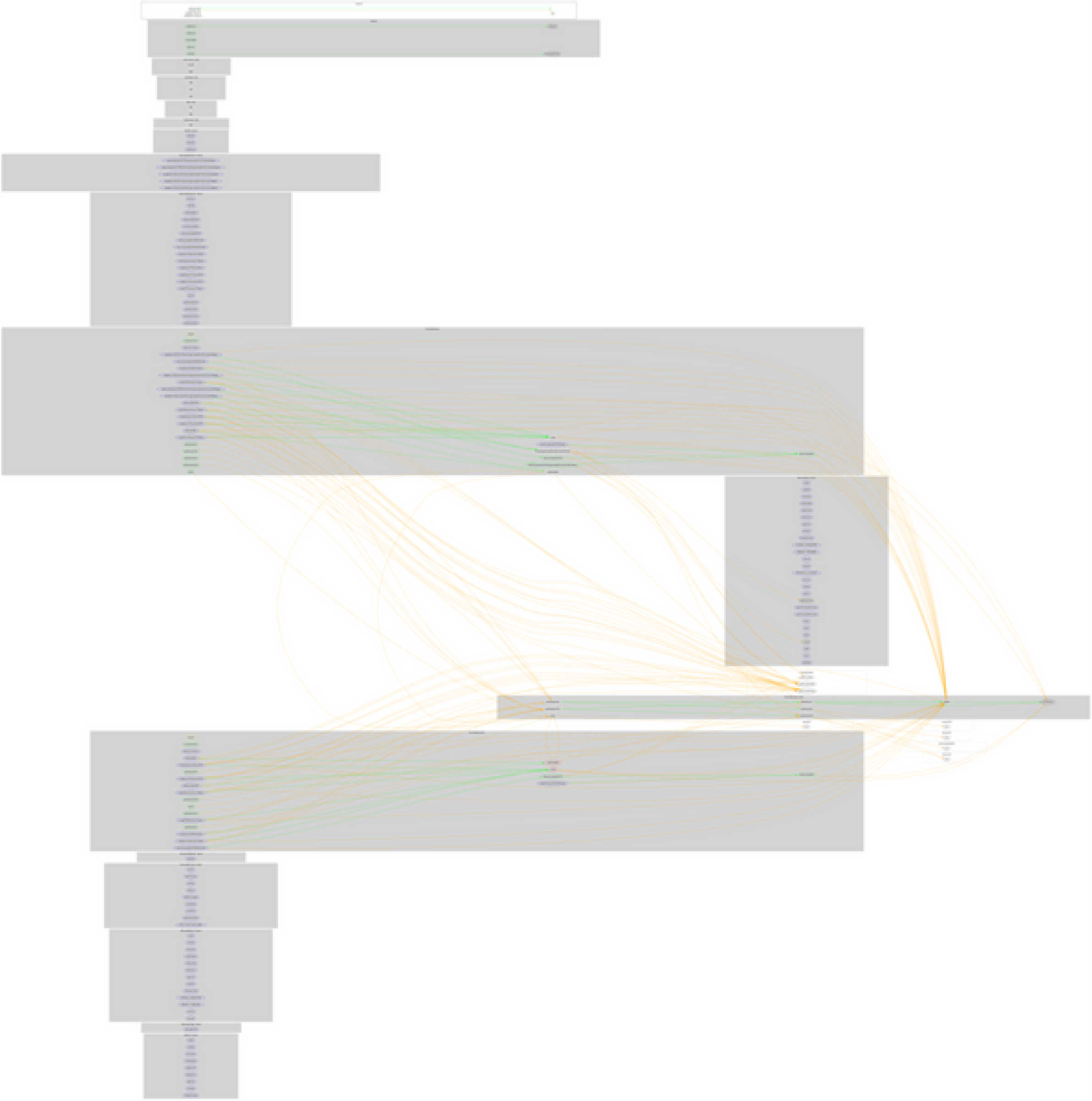
	swapExactETHForTokens	External	Payable	ensure
	swapTokensForExactETH	External	✓	ensure
	swapExactTokensForETH	External	✓	ensure
	swapETHForExactTokens	External	Payable	ensure
	_swapSupportingFeeOnTransferTokens	Internal	✓	
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	ensure
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	ensure
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	ensure
	quote	Public		-
	getAmountOut	Public		-
	getAmountIn	Public		-
	getAmountsOut	Public		-
	getAmountsIn	Public		-
PancakeRouter01	Implementation	IPancakeRouter01		
		Public	✓	-
		External	Payable	-
	_addLiquidity	Private	✓	
	addLiquidity	External	✓	ensure
	addLiquidityETH	External	Payable	ensure
	removeLiquidity	Public	✓	ensure
	removeLiquidityETH	Public	✓	ensure
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	_swap	Private	✓	
	swapExactTokensForTokens	External	✓	ensure
	swapTokensForExactTokens	External	✓	ensure
	swapExactETHForTokens	External	Payable	ensure

	swapTokensForExactETH	External	✓	ensure
	swapExactTokensForETH	External	✓	ensure
	swapETHForExactTokens	External	Payable	ensure
	quote	Public		-
	getAmountOut	Public		-
	getAmountIn	Public		-
	getAmountsOut	Public		-
	getAmountsIn	Public		-

Inheritance Graph



Flow Graph



Summary

PancakeRouter contract implements a financial and utility mechanism. This audit investigates security issues, business logic concerns, and potential improvements.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment 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 Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors’ funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io