auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Donswap

DATED : 2 September 23'

AUDIT SUMMARY

Project name – Donswap

Date: 2 September 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: **Passed**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	1
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1.Code Comparison:

We used specialized tools to perform a line-by-line comparison between the project's code and that of Uniswap V2 to identify any differences.

2.Differential Analysis:

Our audit team conducted a thorough review of the differentials to assess whether they introduce any security vulnerabilities or logical errors.

3.Additional Modules:

Any additional smart contracts, not part of the original Uniswap V2, were audited as separate entities, following our standard auditing procedures.

Token Information

Router Address :
0x3A6a1316109Bf9ee79877C73FDE2b8132308690A

Factory Address:
0xF5695985CeBD8C9F0650D848aeFb1Cf8AFC3ec7c

Network: opBNB

Contract Type: Decentralized Exchange (DEX)

Deployer:
0xc6f560a7559963a3ca3a479eded4f43b1d08e4d9

Checksum:
481a8c4dcb4665feeac96a69412e38db5afd3ae8

Testnet version:
The tests were performed on a forked version of **opBNB** using Donswap smart contracts in forge (foundry) environment

TOKEN OVERVIEW

Forked Codebase:
This project is an exact fork of Uniswap V2, a well-known and previously audited decentralized exchange. Due to the established reputation and multiple prior audits of Uniswap V2, our audit focused primarily on differences between this project and the original Uniswap V2 codebase.
Limitations

Reduced Depth of Review:
While Uniswap V2's codebase has been audited multiple times, it's important to note that our audit did not re-examine the original code in depth. Our focus was on identifying deviations and ensuring that those changes do not introduce new vulnerabilities.

Contextual Differences:
Even if the codebase is identical, the context in which the fork operates might differ, including user behavior, governance, or tokenomics, which are outside the scope of this audit.

Key Features:
1.Automated Market Making: Donswap utilizes an x * y = k formula for its AMM, where x and y are the amounts of two tokens in a liquidity pool, and k is a constant. This formula allows for efficient and low-slippage trading.

2.Decentralization: Being a DEX, Donswap is entirely decentralized, allowing users to maintain control over their assets at all times. There is no need for KYC (Know Your Customer) checks, and the code is open-source.

TOKEN OVERVIEW

3.Liquidity Provision: Users can become liquidity providers by depositing tokens in pairs, earning a share of the trading fees in return.

4.Token Swaps: Donswap supports direct ERC-20 to ERC-20 swaps

5.Routing: Donswap also offers multi-hop trades, routing through multiple pairs to optimize trading.

AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.

- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.

- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.

- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.

- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.

- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

auditace.tech

VULNERABILITY CHECKLIST

- Return values of low-level calls
- Private modifier
- Multiple Sends
- Using Suicide
- Gas Limitand Loops
- Address hardcoded
- Exception Disorder
- Using inline assembly
- Divide before multiply
- Missing Zero Address Validation
- Compiler version not fixed
- **Gasless Send**
- Using block.timestamp
- Re-entrancy
- Tautology or contradiction
- Timestamp Dependence
- Revert/require functions
- Use of tx.origin
- Integer overflow/underflow
- Dangerous strict equalities
- Using SHA3
- Using throw

CLASSIFICATION OF RISK

Severity	Description
Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

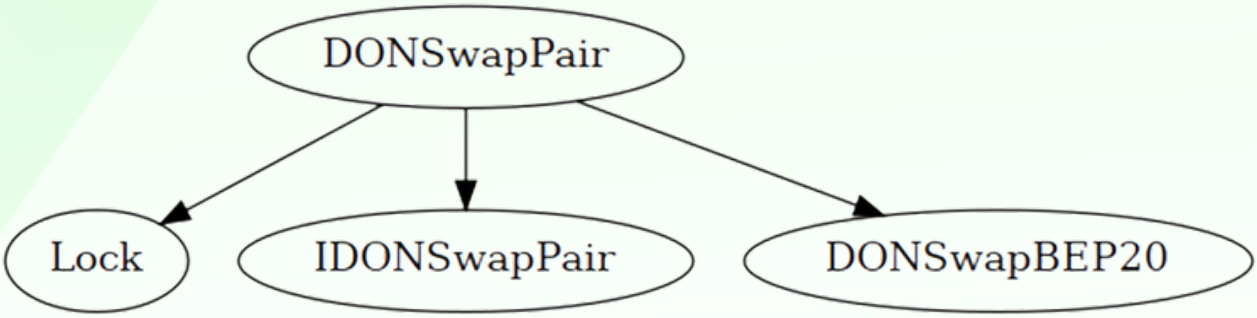
Findings

Severity	Found
Critical	0
High-Risk	0
Medium-Risk	0
Low-Risk	0
Gas Optimization / Suggestions	1

INHERITANCE TREE

- **Router:**

- **Factory:**



CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IDONSwapBEP20	Interface			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🛑	NO❗
IDONSwapPair	Interface	IDONSwapBEP20		
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	initialize	External ❗	🛑	NO❗
└	getReserves	External ❗		NO❗
└	mint	External ❗	🛑	NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗

CONTRACT ASSESMENT

DONSwapLibrary	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 🔒		
TransferHelper	Library			
└	safeApprove	Internal 🔒	🛑	
└	safeTransfer	Internal 🔒	🛑	
└	safeTransferFrom	Internal 🔒	🛑	
└	safeTransferETH	Internal 🔒	🛑	
IDONSwapRouter	Interface			
└	factory	External ❗		NO❗
└	WBNB	External ❗		NO❗
└	addLiquidity	External ❗	🛑	NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	removeLiquidity	External ❗	🛑	NO❗
└	removeLiquidityETH	External ❗	🛑	NO❗
└	removeLiquidityWithPermit	External ❗	🛑	NO❗

CONTRACT ASSESMENT

└	removeLiquidityETHWithPermit	External ❗	🛑	NO❗
└	swapExactTokensForTokens	External ❗	🛑	NO❗
└	swapTokensForExactTokens	External ❗	🛑	NO❗
└	swapExactETHForTokens	External ❗	💵	NO❗
└	swapTokensForExactETH	External ❗	🛑	NO❗
└	swapExactTokensForETH	External ❗	🛑	NO❗
└	swapETHForExactTokens	External ❗	💵	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑
NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	pairFor	External ❗		NO❗
IDONSwapFactory	Interface			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
└	INIT_CODE_PAIR_HASH	External ❗		NO❗
IBEP20	Interface			
└	totalSupply	External ❗		NO❗
└	decimals	External ❗		NO❗
└	symbol	External ❗		NO❗

CONTRACT ASSESMENT

└	name	External ❗		NO❗
└	getOwner	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
IWBNB	Interface			
└	deposit	External ❗	💵	NO❗
└	transfer	External ❗	🛑	NO❗
└	withdraw	External ❗	🛑	NO❗
DONSwapRouter	Implementation	IDONSwapRouter		
└	<Constructor>	Public ❗	🛑	NO❗
└	<Receive Ether>	External ❗	💵	NO❗
└	_addLiquidity	Internal 🔒	🛑	
└	addLiquidity	External ❗	🛑	ensure
└	addLiquidityETH	External ❗	💵	ensure
└	removeLiquidity	Public ❗	🛑	ensure
└	removeLiquidityETH	Public ❗	🛑	ensure
└	removeLiquidityWithPermit	External ❗	🛑	NO❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO❗
└	removeLiquidityETHSupportingFeeOnTransferTokens	Public ❗	🛑	ensure
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
└	_swap	Internal 🔒	🛑	
└	swapExactTokensForTokens	External ❗	🛑	ensure
└	swapTokensForExactTokens	External ❗	🛑	ensure
└	swapExactETHForTokens	External ❗	💵	ensure
└	swapTokensForExactETH	External ❗	🛑	ensure
└	swapExactTokensForETH	External ❗	🛑	ensure
└	swapETHForExactTokens	External ❗	💵	ensure
└	_swapSupportingFeeOnTransferTokens	Internal 🔒	🛑	
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	ensure
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	ensure

CONTRACT ASSESMENT

└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	ensure
└	quote	Public ❗		NO❗
└	getAmountOut	Public ❗		NO❗
└	getAmountIn	Public ❗		NO❗
└	getAmountsOut	Public ❗		NO❗
└	getAmountsIn	Public ❗		NO❗
└	pairFor	Public ❗		NO❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

MANUAL TESTING

Gas Optimization – String errors

Severity: Informational

function: ---

Status: Open

Overview:

Its suggested to use type “error” for defining error messages instead of using strings, this will reduce contract overall size

Suggestion

Example:

```
error Invaid_K();
if (k2 < k1){
        revert Invalid_K();
}
```

DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.

ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.

https://auditace.tech/

https://t.me/Audit_Ace

https://twitter.com/auditace_

https://github.com/Audit-Ace