



CyberCrime Shield
cybercrimeshield.org

Smart Contract Audit Report

DERIVE FINANCE

<https://derive.fi>

AUDIT TYPE: **PUBLIC**



<https://cybercrimeshield.org/secure/derive>
ID:2390325

April 20, 2021















TABLE OF CONTENTS

SMART CONTRACTS.....	3
INTRODUCTION.....	4
AUDIT METHODOLOGY.....	5
ISSUES DISCOVERED.....	6
AUDIT SUMMARY.....	6
FINDINGS.....	7
CONCLUSION.....	8



SMART CONTRACTS

<https://github.com/derivefinance/derive-contracts>

-  guarded
-  helper
-  interfaces
-  VirtualSwap
-  LPToken.sol
-  MathUtils.sol
-  OwnerPausableUpgradeable.sol
-  StakeableTokenWrapper.sol
-  Swap.sol
-  SwapDeployer.sol
-  SwapFlashLoan.sol
-  SwapUtils.sol

Mirror: <https://cybercrimeshield.org/secure/uploads/derive.zip>

CRC32: DDFC297F

MD5: 2933ECEE3226AD401A5C6C6684AB119A

SHA-1: 6246532875EDE49B40BF3AB1A9958DB94B5E3070



INTRODUCTION

Blockchain platforms, such as Nakamoto's Bitcoin, enable the trade of cryptocurrencies between mutually mistrusting parties.

To eliminate the need for trust, Nakomoto designed a peer-to-peer network that enables its peers to agree on the trading transactions.

Smart contracts have shown to be applicable in many domains including financial industry, public sector and cross-industry.

The increased adoption of smart contracts demands strong security guarantees. Unfortunately, it is challenging to create smart contracts that are free of security bugs.

As a consequence, critical vulnerabilities in smart contracts are discovered and exploited every few months.

In turn, these exploits have led to losses reaching billions worth of USD in the past few years.

It is apparent that effective security checks for smart contracts are strictly needed.

Our company provides comprehensive, independent smart contract auditing.

We help stakeholders confirm the quality and security of their smart contracts using our standardized audit process.

Derive Finance is a project holding swap exchange and liquidity provider function.

Most contracts operate same as Curve Finance contracts <https://github.com/curvefi>

Derive Finance smart contacts are adapted for Binance Smart Chain in accordance with the recommendations of the developers of the BSC platform.

The scope of this audit was to analyze and document the Derive Finance contracts.

This document is not financial advice, you perform all financial actions on your own responsibility.



AUDIT METHODOLOGY

1. Design Patterns

We inspect the structure of the smart contract, including both manual and automated analysis.

2. Static Analysis

The static analysis is performed using a series of automated tools, purposefully designed to test the security of the contract.

All the issues found by tools were manually checked (rejected or confirmed).

3. Manual Analysis

Contract reviewing to identify common vulnerabilities. Comparing of requirements and implementation. Reviewing of a smart contract for compliance with specified customer requirements. Checking for energy optimization and self-documentation. Running tests of the properties of the smart contract in test net.



ISSUES DISCOVERED

Issues are listed from most critical to least critical. Severity is determined by an assessment of the risk of exploitation or otherwise unsafe behavior.

Severity Levels

Critical - Funds may be allocated incorrectly, lost or otherwise result in a significant loss.

Medium - Affects the ability of the contract to operate.

Low - Minimal impact on operational ability.

Informational - No impact on the contract.

AUDIT SUMMARY

The summary result of the audit performed is presented in the table below

Findings list:

LEVEL	AMOUNT
Critical	0
Medium	0
Low	2
Informational	3



FINDINGS

1. Multiplication after division (2)

Solidity operates only with integers. Thus, if the division is done before the multiplication, the rounding errors can increase dramatically.

contracts/SwapUtils.sol
lines: 305, 414

2. Costly loop

If `array.length` is large enough, the function exceeds the block gas limit, and transactions calling it will never be confirmed

contracts/SwapUtils.sol
lines: 299, 355, 394, 407, 462, 549, 667

3. Overpowered role (3)

Function is callable only from one address. Therefore, the system depends heavily on this address. In this case, there are scenarios that may lead to undesirable consequences for investors, e.g. if the private key of this address becomes compromised.

contracts/SwapFlashLoan.sol
line: 151

contracts/guarded/SwapGuarded.sol
line: 176

contracts/guarded/Allowlist.sol
lines: 44, 45, 122, 136



CONCLUSION

- Contracts have high code readability
- Gas usage is optimal
- Contracts are fully BSC completable
- No backdoors or overflows are present in the contracts

