freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

DeltaFlip

$DeltaF

27/12/2021

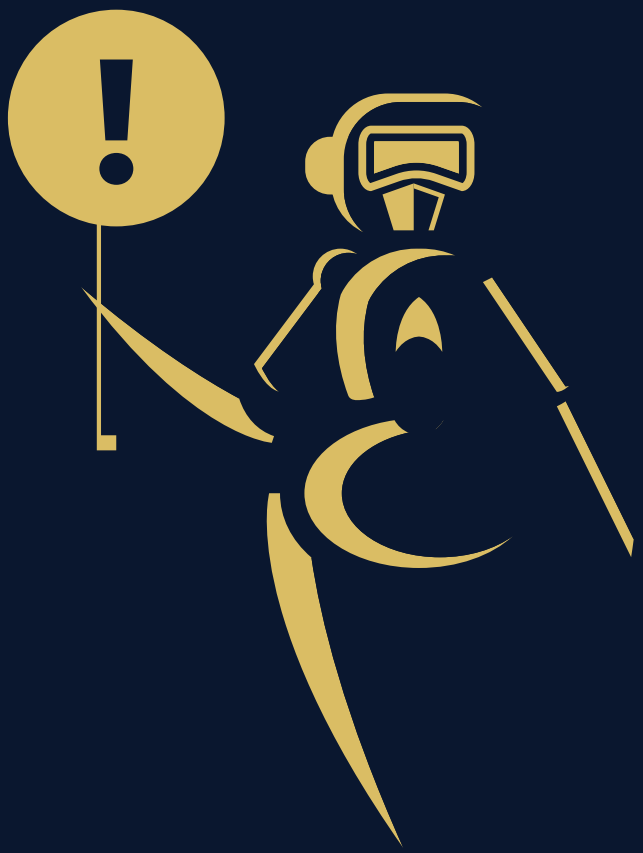
TABLE OF CONTENTS

DISCLAIMER

The information provided on this analysis decoument is only for general information and should not be used as a reason to invest.

FreshCoins Team will take no payment for manipulating the results of this audit.

The score and the result will stay on this project page information on our website https://freshcoins.io
FreshCoins Team does not guarantees that a project will not sell off team supply, or any other scam strategy (RUG or Honeypot etc)



INTRODUCTION

FreshCoins (Consultant) was contracted by DeltaFlip (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x3D06CB9E8Fa1c7638a8B3D8d8B8755f1F6B7164B

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 27/12/2021

WEBSITE DIAGNOSTIC

https://www.deltaflip.net/

0-49 50-89 90-100

Performance	Accessability	Best Practices	SEO	Progressive Web App
66	91	85	92	NA

Metrics

First Contentful Paint
1.7 s

Time to interactive
5.4 s

Speed Index
3.6 s

Total Blocking Time
260 ms

Large Contentful Paint
6.3 s

Cumulative Layout Shift
0.092

Issues found

Use video formats for animated content.

Serve images in next-gen formats.

Eliminate render-blocking resources.

Ensure text remains visible during webfont load.

Image elements do not have explicit width and height.

Avoid enormous network payloads Total size was 4,306 KiB

Background and foreground colors do not have a sufficient contrast ratio.

<html> element does not have a [lang] attribute.

Links do not have a discernible name.

Heading elements are not in a sequentially-descending order.

AUDIT OVERVIEW

98

Security Score

97

Static Scan
Automatic scanning for common vulnerabilities

99

ERC Scan
Automatic checks for ERC's conformance

0 High

0 Medium

1 Low

0 Optimizations

0 Informational

No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passsed
2	Reentrancy and Cross-function	Passsed
3	Front running	Passsed
4	Timestamp dependence	Passsed
5	Integer Overflow and Underflow	Passsed
6	Reverted DoS	Passsed
7	DoS with block gas limit	Low
8	Methods execution permissions	Passsed
9	Exchange rate impact	Passsed
10	Malicious Event	Passsed
11	Scoping and Declarations	Passsed
12	Uninitialized storage pointers	Passsed
13	Design Logic	Passsed
14	Safe Zeppelin module	Passsed

OWNER PRIVILEGES

Contract owner has the authority to increase fees up to 25%. The contract's function that can update fees have limitations to what the fees can be up to a total of 25%.

```
function setTokenRewardsFee(uint256 value) external onlyOwner {
    tokenRewardsFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setLiquiditFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee);
    require(totalFees <= 25, "Total fee is over 25%");
}
```

Contract owner can change wallet address associated with marketing wallet - **setMarketingWallet** function

```
function setMarketingWallet(address payable wallet) external onlyOwner {
    _marketingWalletAddress = wallet;
  }
```

Contract owner has the authority to change tx amount. The owner may take advantage of it by setting the **swapTokensAtAmount** to a very small number.

```
function setSwapTokensAtAmount(uint256 amount) external onlyOwner {
    swapTokensAtAmount = amount;
  }
```

Contract owner has the authority to transfer ownership of the contract to a new account.

```
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    _setOwner(newOwner);
  }
```

```
function _setOwner(address newOwner) private {
    address oldOwner = _owner;
    _owner = newOwner;
    emit OwnershipTransferred(oldOwner, newOwner);
  }
```

Recomandation:

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. The risk can be prevented by temporarily locking the contract or renouncing ownership.

CONCLUSION AND ANALYSIS

Smart Contracts within the scope were manually reviewd and analyzed with static tools.

Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.

Found 1 low issue during the first review.

TOKEN DETAILS

Details

Buy fees:	10%
Sell fees:	10%
Max TX:	N/A
Max Sell:	N/A

Honeypot Risk

Ownership:	Owned
Blacklist:	Not detected
Modify Max TX:	setSwapTokensAtAmount
Modify Max Sell:	Not detected
Disable Trading:	Not detected

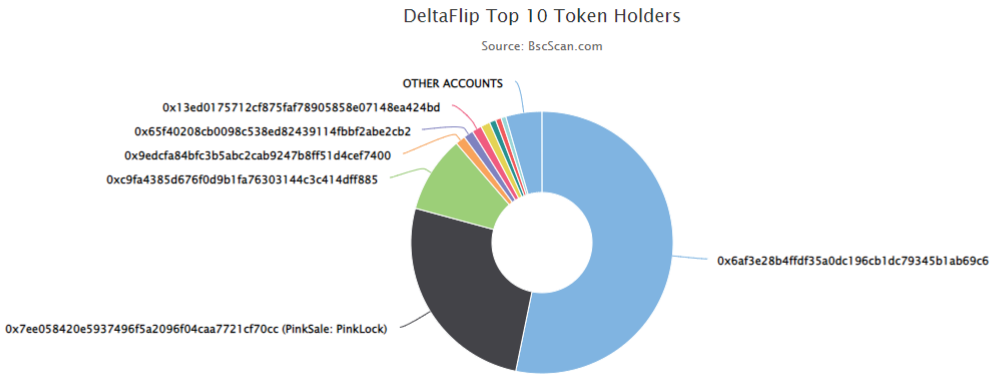
Rug Pull Risk

Liquidity:	26% locked
Holders:	Clean

DELTAFLIP TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 95.55% (955,467,996.00 Tokens) of DeltaFlip

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 42



(A total of 955,467,996.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x6af3e28b4ffdf35a0dc196cb1dc79345b1ab69c6	532,200,000	53.2200%
2	PinkSale: PinkLock	260,000,000	26.0000%
3	0xc9fa4385d676f0d9b1fa76303144c3c414dff885	94,567,996	9.4568%
4	0x9edcfa84bfc3b5abc2cab9247b8ff51d4cef7400	12,300,000	1.2300%
5	0x65f40208cb0098c538ed82439114fbbf2abe2cb2	12,000,000	1.2000%
6	0x13ed0175712cf875faf78905858e07148ea424bd	12,000,000	1.2000%
7	0xf4e1d98494dbc9379449801abf409de51d80b474	11,500,000	1.1500%
8	0xd6c25b8626ba28a08c669e40a55b10f8cb936eb4	7,900,000	0.7900%
9	0x1e1f321450897771fa94c94968568f4e6319e2eb	7,000,000	0.7000%
10	0x90788d27ca8da62f7796221aeb74d3d9acefd726	6,000,000	0.6000%

TECHNICAL DISCLAIMER

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. The audit can't guarantee the explicit security of the audited project / smart contract.