



expelee

A Secure Place For Web3

SMART CONTRACT AUDIT OF

Son of Doge Final



Contract Address

0x0ca553DE0599a79Aaf43943cf254E60183a0cCA3

www.expelee.com Page 1 |





Audit Summary

Expelee team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analysed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

Audit Result: PASSED

Ownership: NOT RENOUNCED

KYC Verification: Done

Audit Date: 27/07/2022

Audit Team: EXPELEE

Be aware that smart contracts deployed on the blockchain aren't resistant to internal exploit, external vulnerability, or hack. For a detailed understanding of risk severity, source code vulnerability, functional hack, and audit disclaimer, kindly refer to the audit.

www.expelee.com Page 2 |





DISCLAMER

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 Expelee 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 Expelee 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 Expelee team disclaims any liability for the resulting losses.

www.expelee.com Page 3 |



Contract Review

Contract Name	SonOfDogeFinal
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	Yes with 200 runs
License	NO License
Explorer	https://bscscan.com/address/0x0ca55 3DE0599a79Aaf43943cf254E60183a0cC A3#code
Symbol	SODTFC
Decimals	18
Total Supply	1,000,000,000
Domain	https://sonofdogefinal.com/

www.expelee.com | Page 4 |





Project Review

Token Name: ArtexToken

Web Site: https://sonofdogefinal.com/

Twitter: https://twitter.com/sodfinalchapter

Telegram: https://t.me/sodfinalchapter

Contract Address:

0x0ca553DE0599a79Aaf43943cf254E60183a0cCA3

Platform: Binance Smart Chain

Token Type: BEP 20

Language: SOLIDITY

www.expelee.com Page 5 |





Audit Methodology

The scope of this report is to audit the smart contract source code. We have scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category

Smart Contract Vulnerabilities

- Unhandled Exceptions
- Transaction Order Dependency
- Integer Overflow
- Unrestricted Action
- Incorrect Inheritance Order
- Typographical Errors
- Requirement Violation

Source Code Review

- Gas Limit and Loops
- Deployment Consistency
- Repository Consistency
- Data Consistency
- Token Supply Manipulation

Functional Assessment

- Operations Trail & Event Generation
- Assets Manipulation
- Liquidity Access

www.expelee.com | Page 6 |





Vulnerability Checklist

Nō	Description.	Result
1	Compiler warnings.	Passed
2	Race conditions and Re-entrancy. Cross-function raceconditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module.	Passed
21	Fallback function security.	Passed

www.expelee.com | Page 7 |

Manual Audit

- Low-Risk
- 3 low-risk code issues found
- Medium-Risk0 medium-risk code issues found
 - High-Risk

 O high-risk code issues found

www.expelee.com Page 8 |



Audit Summary

Number of lines: 1660 (+ 0 in dependencies, + 0 in tests)

Number of assembly lines: 0

Number of contracts: 10 (+ 0 in dependencies, + 0 tests)

Number of optimization issues: 10 Number of informational issues: 104

Number of low issues: 3 Number of medium issues: 0 Number of high issues: 0 ERCs: ERC20, ERC2612

+	+ <u></u>	+ <u></u> -	<u></u>	+ <u></u> -	+ <u></u>
Name	# functions	ERCS	ERC20 info	Complex code	Features
SafeMath	+ 8	+ 		+ No	+
Address	7			No	Send ETH
L					Assembly
IUniswapV2Factory	8		1	No	ı
IUniswapV2Pair	26	ERC20,ERC2612	No Minting	No	ı
1	l		Approve Race Cond.		ı
1	l		1	l	1
IUniswapV2Router02	24		1	No	Receive ETH
SonOfDogeFinal	81	ERC20	No Minting	Yes	Receive ETH
1			Approve Race Cond.		Send ETH
$\langle \cdot \rangle$				1	Tokens interaction

www.expelee.com | Page 9 |







1) Functions that send Ether to arbitrary destinations

Unprotected call to a function sending Ether to an arbitrary address.

```
(success, ) = address(marketingAddress).call{
     value: address(this).balance
}("");
```

Recommendation

Ensure that an arbitrary user cannot withdraw unauthorized funds.

www.expelee.com | Page 10 |





2) Reentrancy vulnerabilities

Detection of the reentrancy bug. Do not report reentrancies that don't involve Ether (see reentrancy-no-eth)

Recommendation

Apply the check-effects-interactions pattern.

www.expelee.com | Page 11 |





3) Dangerous usage of tx.origin

tx.origin-based protection can be abused by a malicious contract if a legitimate user interacts with the malicious contract.

Recommendation

Do not use tx.origin for authorization.

www.expelee.com | Page 13 |





Manual Audit (Contract Function)

```
contract SonOfDogeFinal is Context, IERC20, Ownable {
        using SafeMath for uint256;
        using Address for address;
        address payable public marketingAddress;
        address payable public devAddress;
        address payable public liquidityAddress;
        mapping(address => uint256) private _rOwned;
        mapping(address => uint256) private tOwned;
        mapping(address => mapping(address => uint256)) private _allowances;
        // Anti-bot and anti-whale mappings and variables
        mapping(address => uint256) private _holderLastTransferTimestamp; // to hold last
Transfers temporarily during launch
        bool public transferDelayEnabled = true;
        bool public limitsInEffect = true;
        mapping(address => bool) private _isExcludedFromFee;
        mapping(address => bool) private _isExcluded;
        address[] private _excluded;
        uint256 private constant MAX = ~uint256(0);
        uint256 private constant _tTotal = 1 * 1e12 * 1e18;
        uint256 private _rTotal = (MAX - (MAX % _tTotal));
        uint256 private _tFeeTotal;
        string private constant _name = "Son of Doge Final";
        string private constant _symbol = "SODTFC";
        uint8 private constant _decimals = 18;
        // these values are pretty much arbitrary since they get overwritten for every txn, but
the placeholders make it easier to work with current contract.
        uint256 private _taxFee;
        uint256 private previousTaxFee = taxFee;
        uint256 private _liquidityFee;
        uint256 private _previousLiquidityFee = _liquidityFee;
        uint256 private constant BUY = 1;
        uint256 private constant SELL = 2;
        uint256 private constant TRANSFER = 3;
        uint256 private buyOrSellSwitch;
```

www.expelee.com | Page 13 |





Important Points To Consider

✓ Verified contract source

Token is sellable (not a honeypot) at this time

- X Ownership renounced or source does not contain an owner contract
 - X Source does not contain a fee modifier
- ✓ Owner/creator wallet contains less than 5% of circulating token supply (0%)
- X All other holders possess less than 5% of circulating token supply.

www.expelee.com | Page 14 |





About Expelee

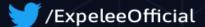
Expelee is a community driven organisation dedicated to fostering an antirug movement. We're here to keep investment safe from fraudsters. We've encountered several rug pulls and know how it feels to be duped, which is why we don't want anybody else to go through the same experience. We are here to raise awareness through our services so that the future of cryptocurrency can be rug-free.

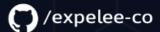
The auditing process focuses to the following considerations with collaboration of an expert team:

- Functionality test of the Smart Contract to determine if proper logic has been followed throughout the whole process.
- Manually detailed examination of the code line by line by experts.
- Live test by multiple clients using Test net.
- Analysing failure preparations to check how the Smart
- Contract performs in case of any bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analysing the security of the on-chain data.

Social Media







www.expelee.com Page 15 |