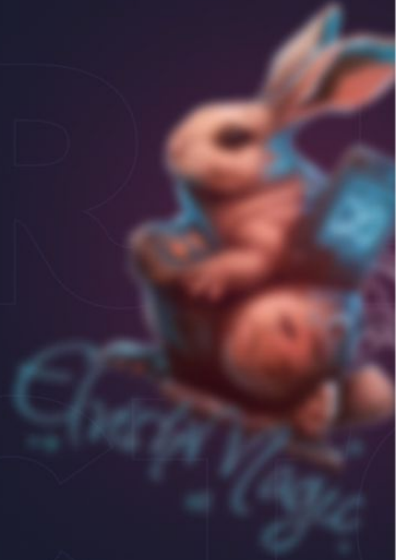




**INSPECTOR
LOVELY**



SMART CONTRACT

SECURITY AUDIT

ELVISH MAGIC



inspector.lovely.finance

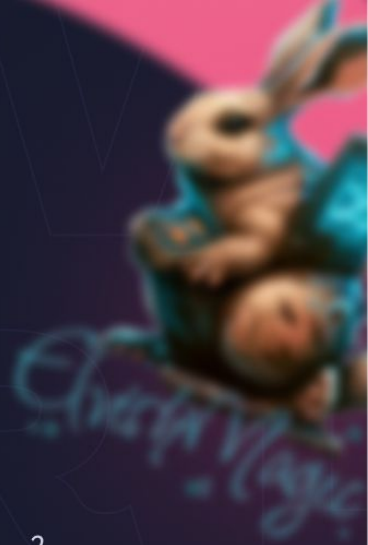


TABLE OF CONTENTS

Table of Contents	2
Disclaimer	3
Procedure	4
Terminology	5
Limitations	5
Basic Security Recommendation	6
Token Contract Details for 29.01.2023	7
Audit Details	7
Social Profiles	8
Contract Analytics	9
EMP Distribution	9
EMP Top Holders	9
Project Website SSL Certification	10
Project Website Optimization for Desktop	11
Project Website Optimization for Mobile	11
Contract Function Details	12
Vulnerabilities checking	14
Security Issues	15
Conclusion for Project Owner	16
Whitepaper of the project	17
INSPECTOR LOVELY Info	18





**INSPECTOR
LOVELY**

DISCLAIMER

This is a comprehensive report based on our automated and manual examination of cybersecurity vulnerabilities and framework flaws of the project's smart contract. Reading the full analysis report is essential to build your understanding of project's security level. It is crucial to take note, though we have done our best to perform this analysis and report, that you should not rely on the our research and cannot claim what it states or how we created it. Before making any judgments, you have to conduct your own independent research. We will discuss this in more depth in the following disclaimer - please read it fully. **DISCLAIMER:** You agree to the terms of this disclaimer by reading this report or any portion thereof. Please stop reading this report and remove and delete any copies of this report that you download and/or print if you do not agree to these conditions. Scan and verify report's presence in the GitHub repository by a qr-code at the title page. This report is for non-reliability information only and does not represent investment advice. No one shall be entitled to depend on the report or its contents, and Inspector Lovely and its affiliates shall not be held responsible to you or anyone else, nor shall Inspector Lovely provide any guarantee or representation to any person with regard to the accuracy or integrity of the report. Without any terms, warranties or other conditions other than as set forth in that exclusion and Inspector Lovely excludes hereby all representations, warrants, conditions and other terms (including, without limitation, guarantees implied by the law of satisfactory quality, fitness for purposes and the use of reasonable care and skills). The report is provided as "as is" and does not contain any terms and conditions. Except as legally banned, Inspector Lovely disclaims all responsibility and responsibilities and no claim against Inspector Lovely is made to any amount or type of loss or damages (without limitation, direct, indirect, special, punitive, consequential or pure economic loses or losses) that may be caused by you or any other person, or any damages or damages, including without limitations (whether innocent or negligent). Security analysis is based only on the smart contracts. No applications or operations were reviewed for security. No product code has been reviewed.



PROCEDURE

Our analysis contains following steps:

1. Project Analysis;

2. Manual analysis of smart contracts:

- Deploying smart contracts on any of the network(Ropsten/Rinkeby) using Remix IDE
- Hashes of all transaction will be recorded
- Behaviour of functions and gas consumption is noted, as well.

3. Unit Testing:

- Smart contract functions will be unit tested on multiple parameters and under multiple conditions to ensure that all paths of functions are functioning as intended.
- In this phase intended behaviour of smart contract is verified.
- In this phase, we would also ensure that smart contract functions are not consuming unnecessary gas.
- Gas limits of functions will be verified in this stage.

4. Automated Testing:

- Mythril
- Oyente
- Manticore
- Solgraph



TERMINOLOGY

We categorize the finding into 4 categories based on their vulnerability:

- Low-severity issue - less important, must be analyzed
- Medium-severity issue - important, needs to be analyzed and fixed
- High-severity issue - important, might cause vulnerabilities, must be analyzed and fixed
- Critical-severity issue - serious bug causes, must be analyzed and fixed.

LIMITATIONS

The security audit of Smart Contract cannot cover all vulnerabilities. Even if no vulnerabilities are detected in the audit, there is no guarantee that future smart contracts are safe, Smart contracts are in most cases safeguarded against specific sorts of attacks. In order to find as many flaws as possible we carried out a comprehensive smart contract audit. Audit is a document that is not legally binding and guarantees nothing.





**INSPECTOR
LOVELY**

BASIC SECURITY RECOMMENDATION

Unlike hardware and paper wallets, hot wallets are connected to the internet and store private keys online, which exposes them to greater risk. If a company or an individual holds significant amounts of cryptocurrency in a hot wallet, they should consider using MultiSig addresses. Wallet security is enhanced when private keys are stored in different locations and are not controlled by a single entity.



**INSPECTOR
LOVELY**



TOKEN CONTRACT DETAILS FOR 29.01.2023

Contract name: **ElvishMagic**

Deployed adress: **0x21d902132A1029B63030F6405a3e025da5459E18**

Total Supply: **1,000,000,000**

Token tracker: **EMP**

Decimals: **18**

Token holders: **1**

Transactions count: **1**

Top 100 holders dominance: **100%**

AUDIT DETAILS

Project Name: **Elvish Magic**

Language: **Solidity**

Compiler Version: **v0.8.3**

Blockchain: **BSC**

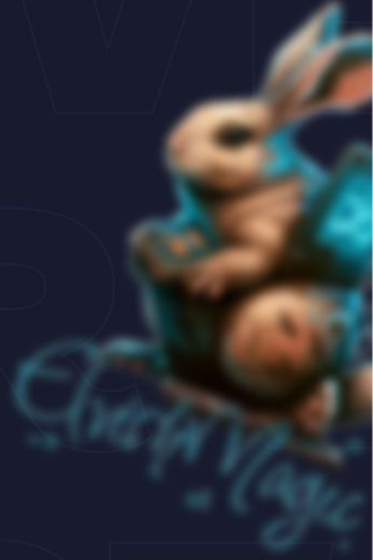


[inspector.lovely.finance](https://twitter.com/inspectorlovely)

Audited by **INSPECTOR LOVELY**



**INSPECTOR
LOVELY**



SOCIAL PROFILES

Project website: <https://elvishmagic.com/>

Project Twitter: <https://twitter.com/elvishmagicpad>

Project Telegram: <https://t.me/elvishmagic>

Project Medium: <https://medium.com/@elvishmagic>

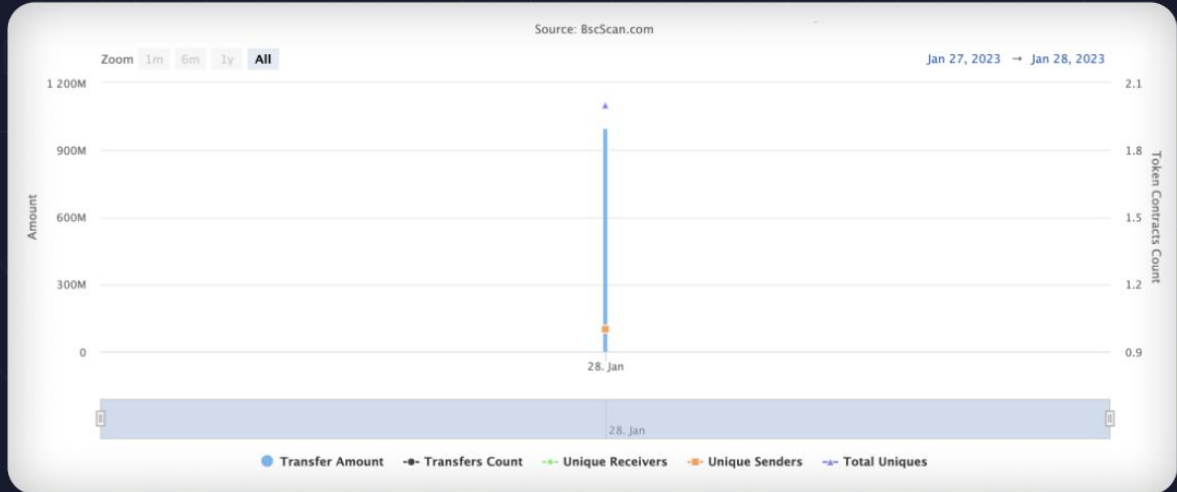


[inspector.lovely.finance](https://www.inspector.lovely.finance)

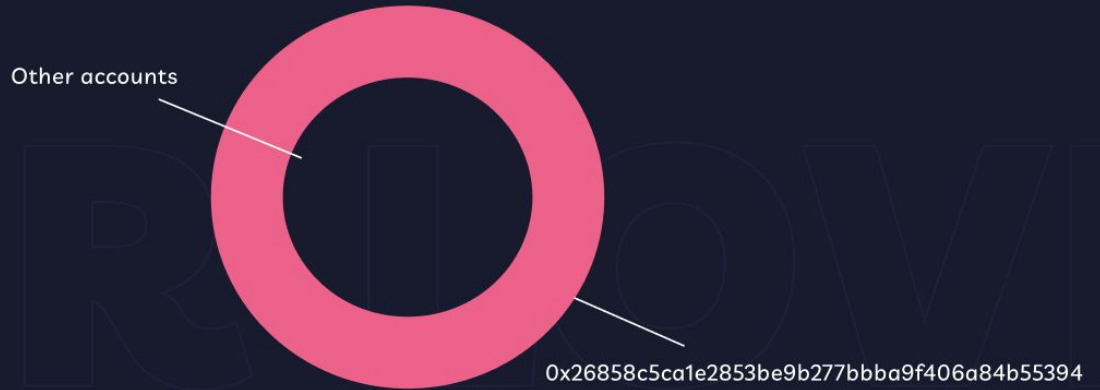
Audited by INSPECTOR LOVELY



CONTRACT ANALYTICS



EMP DISTRIBUTION



LST TOP HOLDERS

Rank	Adress	Quantity (Token)	Percentage
1	0x26858c5ca1e2853be9b277bbba9f406a84b55394	1,000,000,000	100.00%





**INSPECTOR
LOVELY**

Welcome to ElvishMagic

ElvishMagicPad - The fully decentralized protocol for launching new ideas. An all-in-one Incubation Hub with a full-stack DeFi platform across all main blockchain networks.

[Whitepaper](#)

- ✓ JavaScript errors hasn't been found.
- ✓ Malware pop-up windows hasn't been detected.
- ✓ No issues with loading elements, code, or stylesheets.

Project Website SSL Certification

Issued To

Common Name (CN)	*.elvishmagic.com
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	GTS CA 1P5
Organization (O)	Google Trust Services LLC
Organizational Unit (OU)	<Not Part Of Certificate>

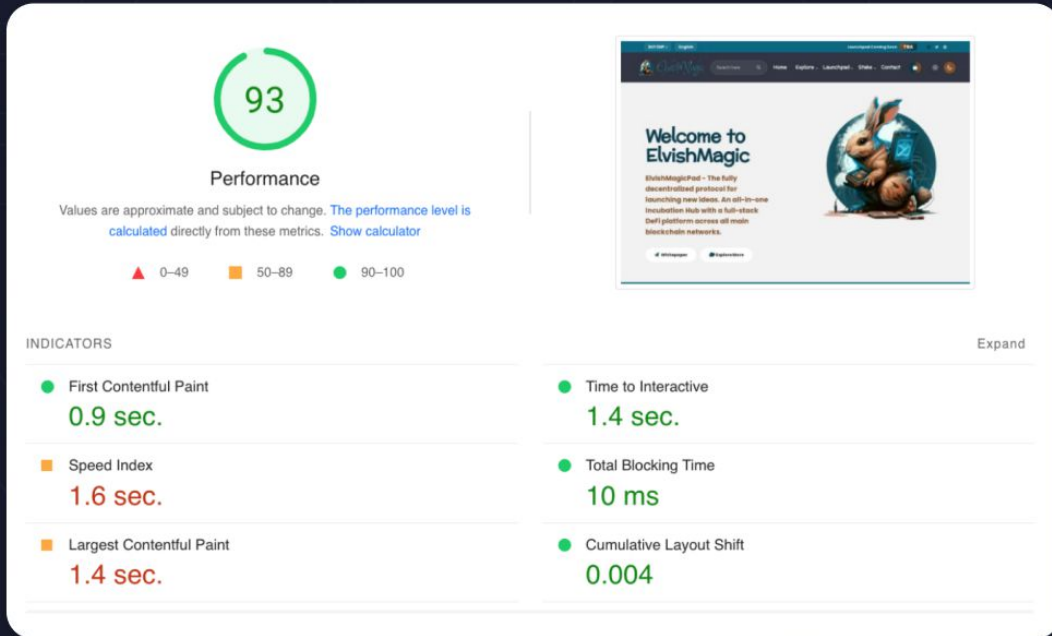


inspector.lovely.finance

Audited by INSPECTOR LOVELY

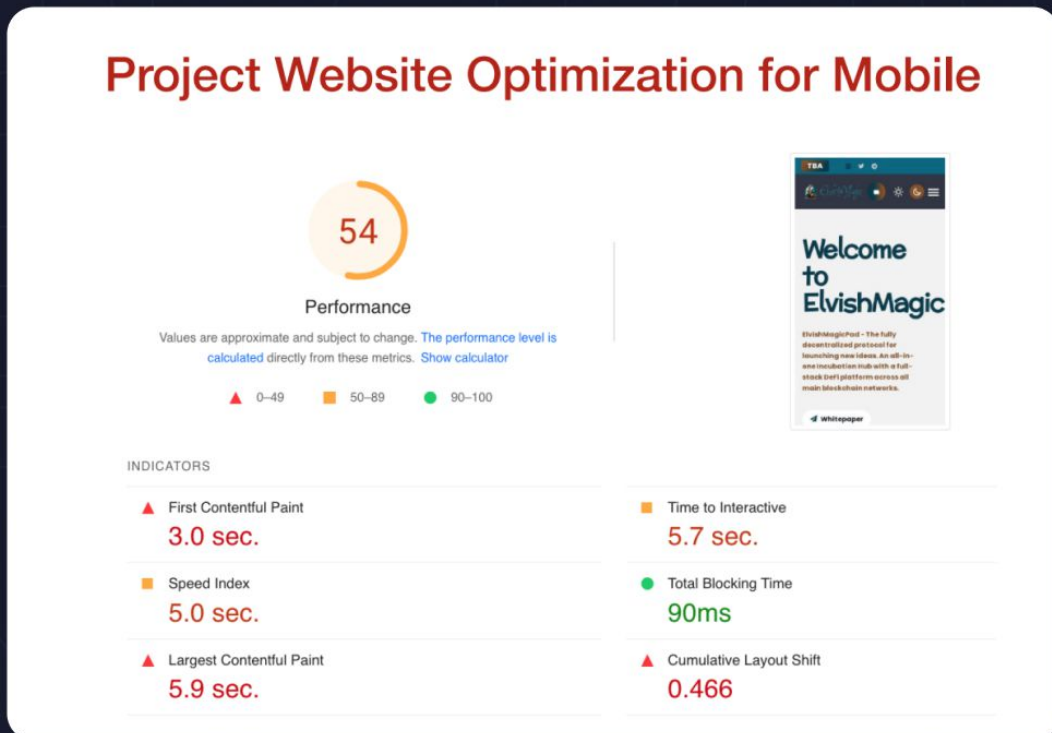


PROJECT WEBSITE PERFORMANCE AUDIT



PROJECT WEBSITE PERFORMANCE AUDIT

Project Website Optimization for Mobile





CONTRACT FUNCTION DETAILS

- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] transfer
- [Ext] allowance
- [Ext] approve
- [Ext] transferFrom
- [Int] _msgSender
- [Pub] owner
- [Pub] renounceOwnership
- [Pub] transferOwnership
- [Pub] name
- [Pub] symbol
- [Pub] decimals
- [Ext] feeTo
- [Ext] feeToSetter
- [Ext] getPair
- [Ext] allPairs
- [Ext] allPairsLength
- [Ext] createPair
- [Ext] setFeeTo
- [Ext] setFeeToSetter
- [Ext] name
- [Ext] symbol
- [Ext] decimals
- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] allowance
- [Ext] approve
- [Ext] transfer
- [Ext] transferFrom
- [Ext] DOMAIN_SEPARATOR
- [Ext] PERMIT_TYPEHASH
- [Ext] nonces
- [Ext] permit
- [Ext] MINIMUM_LIQUIDITY
- [Ext] factory
- [Ext] token0
- [Ext] token1
- [Ext] getReserves
- [Ext] price0CumulativeLast
- [Ext] price1CumulativeLast
- [Ext] kLast
- [Ext] mint
- [Ext] burn
- [Ext] swap
- [Ext] skim
- [Ext] sync
- [Ext] initialize
- [Ext] factory
- [Ext] WETH
- [Ext] addLiquidity
- [Ext] addLiquidityETH
- [Ext] removeLiquidity
- [Ext] removeLiquidityETH
- [Ext] removeLiquidityWithPermit
- [Ext] removeLiquidityETHWithPermit
- [Ext] swapExactTokensForTokens
- [Ext] swapTokensForExactTokens
- [Ext] swapExactETHForTokens
- [Ext] swapTokensForExactETH
- [Ext] swapExactTokensForETH
- [Ext] swapETHForExactTokens
- [Ext] quote
- [Ext] getAmountOut
- [Ext] getAmountIn
- [Ext] getAmountsOut
- [Ext] getAmountsIn





**INSPECTOR
LOVELY**

CONTRACT FUNCTION DETAILS

- [Ext] removeLiquidityETHSupportingFeeOnTransferTokens
- [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens
- [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens
- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens
- [Ext] setUniswapV2Pair
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer
- [Pub] allowance
- [Pub] approve
- [Pub] transferFrom
- [Pub] increaseAllowance
- [Pub] decreaseAllowance
- [Ext] setBuyFeePercent
- [Ext] setSellFeePercent
- [Ext] setTreasuryAddress
- [Pub] setSwapAndLiquifyEnabled
- [Ext] changeNumTokensSellToAddToLiquidity
- [Ext] excludeFromFee
- [Ext] includeInFee
- [Int] _transfer
- [Prv] swapAndLiquify
- [Prv] addLiquidity
- [Int] _approve
- [Ext] withdrawStuckBNB
- [Ext] removeStuckToken



[inspector.lovely.finance](https://www.inspector.lovely.finance)

Audited by INSPECTOR LOVELY



**INSPECTOR
LOVELY**

ISSUE DESCRIPTION

CHECKING STATUS

Compiler Errors	Completed
Delays in Data Delivery	Completed
Re-entrancy	Completed
Transaction-Ordering Dependence	Completed
Timestamp Dependence	Completed
Shadowing State Variables	Completed
DoS with Failed Call	Completed
DoS with Block Gas Limit	Completed
Outdated Compiler Version	Completed
Assert Violation	Completed
Use of Deprecated Solity Functions	Completed
Integer Overflow and Underflow	Completed
Function default Visibility	Completed
Malicious Event log	Completed
Math Accuracy	Completed
Design Logic	Completed
Fallback Function Security	Completed
Cross-Function race conditions	Completed
Safe Zeppelin Module	Completed



inspector.lovely.finance

Audited by INSPECTOR LOVELY



SECURITY ISSUES

- 1) Incorrect Modifier: **High-severity.**
L313-L317

A modifier plays an important role in implementing access control and other validations for functions. These modifiers should be properly implemented and must use either `_`; or a `revert()` to return back to the function. A misconfigured modifier `lockTheSwap` was detected which missed one of these. This will cause the modifier to return the default value causing unexpected and erroneous behaviour.

Recommendation: Make sure that the modifiers are properly implemented and use at least `_`; or a `revert()`.

- **SOLVED**





**INSPECTOR
LOVELY**

CONCLUSION

High-severity issues exist within smart contracts.

NOTE: Please check the disclaimer above and note, that audit makes no statements or warranties on business model, investment attractiveness or code sustainability. Contract security report for community.



inspector.lovely.finance

Audited by INSPECTOR LOVELY



**INSPECTOR
LOVELY**

WHITEPAPER OF THE PROJECT

The whitepaper of Elvish Magic project has been verified on behalf of INSPECTOR LOVELY team.



[inspector.lovely.finance](https://www.inspectorlovely.finance)

Audited by INSPECTOR LOVELY



**INSPECTOR
LOVELY**

INSPECTOR LOVELY

INFO

Website: Inspector.lovely.finance

Telegram community: t.me/inspectorlovely



[inspector.lovely.finance](https://Inspector.lovely.finance)