

META SWAP MET\$ AUDIT





Meta Swap Launchpad

Security Assessment

Project Name: Meta Swap
Simbol: MET\$

Network: Binance Smart Chain

Submitted for verification at Etherscan.io on 2023-03-23



Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer – please make sure to read it in full. By reading this report or any part of it, you agree to the terms of this disclaimer. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Plutonium and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) owe no duty of care towards you or any other person, nor does Plutonium make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Plutonium hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Plutonium hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Plutonium, for any amount or kind of loss or damage that may result to you or any other person including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort including without limitation negligence, contract, breach of statutory duty, misrepresentation or otherwise under any claim of any nature whatsoever in any jurisdiction in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.



Summary

This report has been prepared for META SWAP smart contracts, to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry



Summary

Project Summary

Audit Summary

Project Name: META SWAP MET\$

Description: Platform DEX AMM / LIQUIDITY PROVIDER / LAUNCHPAD / FARM STAKING / LOTTERY

Platform: Binance Smart Chain

Language: Solidity

Token Contract Address: 0xDd770074aa548B66Ab54CF0f6E67F9354E3b732e

Liquidity Provider Contract Address:

Factory: 0x6E111fbE8995b36735f45fEa44161CfE6Ef9636c Router: 0x47715f01238f52fAD6FD6cA5822db4d68F1F35C0

Launchpad Contract Address:

IDO Factory: 0x38a9d7420891d8177Bfd243b5C8aE3CBcd17E49C

Token Locker Factory: 0xdF13aE34D602b6b289C39b0bF81EA350f4A750F8

Lottery Contract Address: 0xdD8C1193D3B5e572D814035B4Cd9B28c719A3336

Owner Address: 0x2423B8F754AC12f1Fb2Ef5f93ea5F73e3648591b

Farming staking

Farming and staking do not include contracts that have already been deposited as they will be created and verified with each request received.

DEX AMM

The exchange does not provide for the use of contracts, as the liquidity to carry out the transactions is taken directly from the following pools: QUIKSWAP / ANYSWAP / UNISWAP / PANCAKESWAP

Audit Summary

Delivery March 03th, 2023. Methodology Static and Manual Analysis

Audit Summary

Total Issues 0

Critical: 0

· Medium: 0

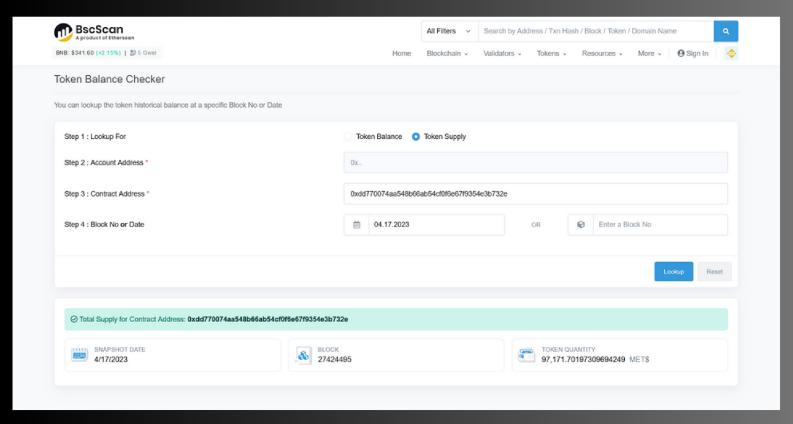
• Low: 0

Informative: 0Optimization: 0



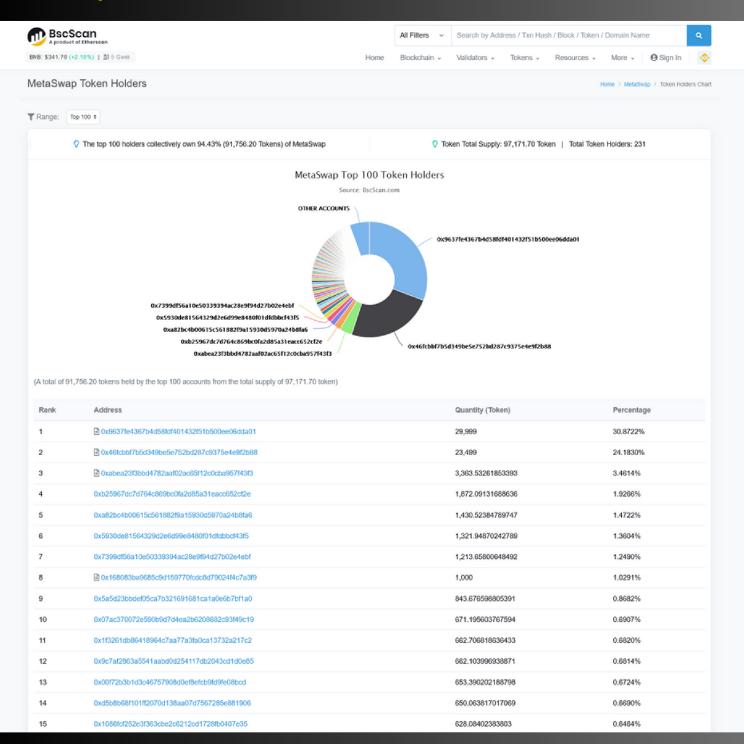


Token Balance Checker





Meta Swap Token Holders





Meta Swap Solidity UML Diagram

Solidity UML Diagram Internal: Public: mul(a: uint256, b: uint256): (c: uint256) <<abstract>> totalSupply(): uint256 div(a: uint256, b: uint256): uint256 <<abstract>> balanceOf(who: address): uint256 sub(a: uint256, b: uint256): uint256 <<abstract>> transfer(to: address, value: uint256); bool add(a: uint256, b: uint256): (c: uint256) <<event>>> Transfer(from: address, to: address, value: uint256) Public: balances: mapping(address=>uint256) totalSupply_: uint256 <abstract>> allowance(owner: address, spender: address): uint256</abstract>> transferFrom(from: address, to: address, value: uint256): bool totalSupply(): uint256 transfer(_to: address, _value: uint256): bool balanceOf(_owner: address): uint256 <<abstract>> approve(spender: address, value: uint256): bool <<event>> Approval(owner: address, spender: address, value: uint256) Public: StandardToken owner: address allowed: mapping(address=>mapping(address=>uint256)) transferOwnership(_newOwner: address) Public: ublic:
transferFrom(_from: address,__to: address,__value: uint256); bool
approve(_spender: address,__value: uint256); bool
allowance(_owner: address,__spender: address); uint256
increaseApproval(_spender: address,__dedValue: uint); bool
decreaseApproval(_spender: address,__subtractedValue: uint); bool volue.
volue. constructor() renounceOwnership() transferOwnership(_newOwner: address) Internal: chains: mapping(bytes32=>uint64) freezings: mapping(bytes32=>uint) freezingBalance: mapping(address=>uint) Pausable MintableToken Public: Public: mintingFinished: bool toKey(_addr: address, _release: uint): (result: bytes32) freeze(_to: address, _until: uint64) Public: <<event>> Pause() <<event>> Mint(to: address, amount: uint256) <event>> Freezed(to: address, release: uint64, amount: uint) <<event>> Unpause()
<<modifier>> whenNotPaused()
<<modifier>> whenPaused() <<event>> MintFinished()
<modifier>> canMint()
<modifier>> hasMintPermination <-event>> Freezed(to: address, release: uint64, amount: uint)

<-event>> Released(owner: address): (balance: uint256)
actualBalanceOf(_owner: address): (balance: uint256)
freezingBalanceOf(_owner: address): (balance: uint256)
freezingCount(_addr: address): (count: uint)
getFreezing(_addr: address, _index: uint): (_release: uint64, _balance: uint)
freezeTo(_to: address, _amount: uint, _until: uint64)
releaseCnee(_selease(_selease): _address): (_selease(_selease(_selease): _address): (_selease(_selease): _address): (_selease(_selease(_selease): _address): (_selease(_selease): _address): (_selease(_selease): (_selease(_selease): _address): (_ mint(_to: address, _amount: uint256): bool finishMinting(): bool pause() unpause() releaseOnce() releaseAll(): (tokens: uint) BurnableToken CappedToken Public: cap: uint256 burn(_who: address, _value: uint256) constructor(_cap: uint256) <event>> Burn(burner: address, value: uint256) mint(to: address, amount: uint256): bool burn(_value: uint256) TOKEN DECIMALS: uint TOKEN_DECIMALS_UINT8: uint8
TOKEN_DECIMAL_MULTIPLIER: uint
TOKEN_NAME: string
TOKEN_SYMBOL: string FreezahleMintableToken Public PAUSED: bool
TARGET_USER: address
CONTINUE_MINTING: bool mintAndFreeze(_to: address, _amount: uint, _until: uint64): bool MainToken initialized: bool Private <<event>> Initialized() constructor()
name(): (_name: string)
symbol(): (_symbol: string) decimals(): (_decimals: uint8)

transferfrom(_from: address,_to; address,_value: uint256): (_success: bool) transfer(_to: address,_value: uint256): (_success: bool)



Meta Swap Security Review

The smart contract consists of 765 lines of code.

In them, a total of 0 problems have been encountered.

The optimization rate is 0.5%

We therefore declare

- 0% low severity problems
- 0% medium problems
- 0% severity problems
- 0% high severity problems

Which produces a total ranking of 99 points.

Conclusion

The verified smart contract does not contain critical errors that can be exploited by third parties.

The contract contains 0 high severity issues and 0 medium severity problems.

To date, no problems or defects are known, in which it could be exploited without the complicity of the developers.

Efficiency has been optimized Gas consumption and reading the code.

The security of the liquidity pair contract is not checked as it is out of scope.

Liquidity will be given on Pancakeswap and Meta Swap.

In greater quantities on the first LP and in smaller quantities on the internal LP maintaining the right initial ratio.

Notes

Please check the disclaimer above and note that the audit makes no representations or guarantees on the business model, on the attractiveness of investments or on the sustainability of the code. The report is expected for the only contract mentioned in the report and does not include any other potential contracts distributed by the Owner.







https://www.metaswapdex.app



★ https://t.me/metaswapdex





★ https://t.me/metaswapinternational





https://twitter.com/MetaSwapDexLp



https://www.youtube.com/@MetaSwap



https://www.tiktok.com/@metaswapdex



https://www.twitch.tv/metaswapdex