

ETH Smart Contract Audit Report

MD5:8c783dc02e4af6047c56e7889c4f00fa

Runtime:0.8s

Scored higher
than42% of similar
code
amongst 50k smart contracts audited by
Anchain.AI .

Score

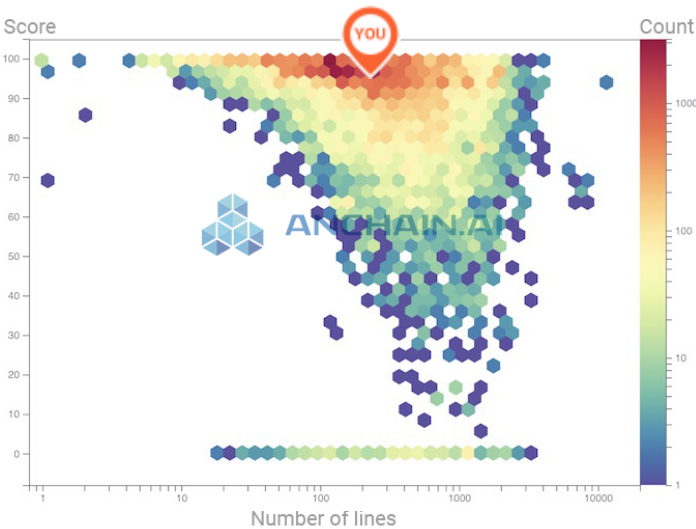
96

Threat Level

Low

Number of lines

232



Overview

Code Class	EVM Coverage
BasicToken	94.6%
ERC20	0%
EZDEX	91.4%
SafeMath	100.0%
StandardToken	92.9%

0 Vulnerabilities Found

High Risk

Medium Risk

Low Risk

Recommendations

- No visibility specified. Defaulting to 'public'.
See line(s) [227](#)
- Consider using exact language version instead.
See line(s) [9](#)
- Missing check on 'msg.data.length' could lead to short-address attack in this ERC20 transfer function.
See line(s) [104](#), [143](#)
- This ERC20 contract does not have a 'decimals' state variable or a 'decimals()' function.
See line(s) [132](#)
- This ERC20 contract does not have a 'name' state variable or a 'name()' function.
See line(s) [132](#)

This ERC20 contract does not have a 'symbol' state variable or a 'symbol()' function.
See line(s) [132](#)

Vulnerability Checklist

BasicToken

- Integer Underflow
- Integer Overflow
- Parity Multisig Bug
- Callstack Depth Attack
- Transaction-Ordering Dependency
- Timestamp Dependency
- Re-Entrancy

ERC20

- Integer Underflow
- Integer Overflow
- Parity Multisig Bug
- Callstack Depth Attack
- Transaction-Ordering Dependency
- Timestamp Dependency
- Re-Entrancy

EZDEX

- Integer Underflow
- Integer Overflow
- Parity Multisig Bug
- Callstack Depth Attack
- Transaction-Ordering Dependency
- Timestamp Dependency
- Re-Entrancy

SafeMath

- Integer Underflow
- Integer Overflow

Parity Multisig Bug
Callstack Depth Attack

Transaction-Ordering Dependency

Timestamp Dependency

Re-Entrancy

StandardToken

Integer Underflow

Integer Overflow

Parity Multisig Bug

Callstack Depth Attack

Transaction-Ordering Dependency

Timestamp Dependency

Re-Entrancy

Similar Smart Contracts

[0xb23b3536ca43e7247ed8660734a083b512fa2e0d](#)



[0xba1acbb3fd5ebf246f32f1392da487d5a882b132](#)



[0xbb6959b0bd7e1fa6578cc72eaf97538a6d446514](#)



[0x1815f95ed77be3e215fc0b93e31a33e0f3c9a78f](#)



[0x23e12ede899959ed3e36331a2e5ec0af4a05f2ca](#)



Refresh



Generated by Anchain.AI CAP Sandbox on 2021/7/22

TERMS OF SERVICE

AnChain.ai's product service is a best-efforts smart contract auditing service. The auditing report may contain false positives and false negatives, and should be only used as a recommendation for code improvement. We accept only legal pieces of code. All entries are logged for research and improvement of service. AnChain.ai does not warrant any rights or service levels, nor does it acquire any rights on the code entered. U.S. law is applicable. The place of jurisdiction is California, United States of America. By sending code to AnChain.ai, you warrant that all your entries are in your sole responsibility and you do not infringe any laws or third-party rights like copyrights and the like. AnChain.ai and its employees shall not be liable for any entries and for any damages resulting thereof. You agree to indemnify, defend and hold them harmless from any legal or financial demands or arising out of the breach of these terms of use, especially from third-party claims regarding infringement of copyrights and the like.