IDEX Security Audit

IDEX security audit, conducted by the Callisto Network Security Department in May 2019.

IDEX Smart Contract Specificities

Audit Request

It is an Ethereum-driven decentralized exchange that supports Ethereum and ERC20 token trading pairs.

https://idex.market/

Source Code:

- https://github.com/AuroraDAO/idex

Disclosure policy:

https://discordapp.com/invite/UHAGGBz

support@auroradao.com

Platform:

ETH.

Number of lines:

324.

IDEX Smart Contract Security Audit Report

Are Your Funds Safe?

1. In scope

Commit hash 4a05eb28e570e9820066474ff2adc924ce7a27bd.

- DVIP.sol.
- Exchange.sol.
- ExchangeWhitelist.sol.
- MyToken.sol.

2. Findings

In total, **4 issues** were reported including:

- 1 medium severity issues.
- 3 low severity issues.

2.1. ERC20 Compliance: `transfer` function returns nothing

Severity: medium.

Description:

Following EIP-20 specifications:

ERC-20 Token Standard specifies for function transfer.
The current convention is for ERC20 tokens to revert if there's an error and return true if there isn't, because this is safest should work for everyone.
But in this implementation transfer function returns 0 bytes is violating the ERC20 interface.

The biggest risk is that a smart contract that is compiled with solc $\geqslant$ 0.4.22 , which is expecting an ERC20 interface, will not be able to interact with MyTokens. This could mean that tokens which are send to such a contract, will be stuck there forever even if the contract has a function to transfer ERC20 token. There are many different scenarios where contracts, handling ERC20 tokens would run into this bug. One example is, that you would not be able to use decentralized exchanges that compiled its contract with solc $\geqslant$ 0.4.22 with this implementation of MyToken.
More details here.

Code snippet:

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/MyToken.sol#L34

2.2. Known vulnerabilities of ERC-20 token

Severity: low.

Description:

1. It is possible to double withdrawal attack. More details here.
2. Lack of transaction handling mechanism issue. WARNING! This is a very common issue and it already caused millions of dollars losses for lots of token users! More details here.

Recommendation:

Add into a function `transfer(address _to, ... )` following code:

```
require( _to != address(this) );
```

2.3. Allowance Approval

Severity: low.

Description:

According to ERC20 standard, when initializing a token contract if any token value is set to any given address a transfer event should be emitted.

Code snippet:

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/MyToken.sol#L26

2.4. Check for an empty input value.

Severity: low.

Description:

There is no check for an empty input value.

Code snippet:

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/Exchange.sol#L42

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/Exchange.sol#L49

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/Exchange.sol#L53

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/ExchangeWhitelist.sol#L33

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/ExchangeWhitelist.sol#L40

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/ExchangeWhitelist.sol#L44

2.5. Deprecated method.

Severity: minor observation.

Description:

The function () { throw; } was a pattern used to prevent implicit acceptance of ether in Solidity versions older than 0.4.0, but today this is unneeded.

Code snippet:

https://github.com/AuroraDAO/idex/blob/4a05eb28e570e9820066474ff2adc924ce7a27bd/MyToken.sol#L72-L74

3. Conclusion

The audited smart contract must not be deployed. Reported issues must be fixed prior to the usage of this contract.

4.Revealing audit reports

https://gist.github.com/yuriy77k/cf776b11b3297a0b0a46e50349ecc78a

https://gist.github.com/yuriy77k/dc45fb5987564479eb29e58b0d485157

https://gist.github.com/yuriy77k/cf2e2fb5c5177f1e3e42c92eee54bc7d

Appendix

Smart Contract Audits by Callisto Network.

Miscellaneous

Why Audit Smart Contracts?

Our Most Popular Audit Reports.

Trust the Blockchain, Audit the Smart Contracts.

Follow Callisto's Security Department on Twitter to get our latest news and updates!

Published on **December 22, 2020**

Security Audits

‹ Previous post

Next post ›

Callisto Network LTD

71-75 Shelton Street
London, Greater London
United Kingdom, WC2H 9JQ

Join Our Community

Resources

FAQ
Timeline
Airdrop
Community Guidelines

Callisto

Partners
Our GitHub repositories
Media Kit

Contact us

Want to sell your CLO coins OTC?