Callisto (https://callisto.network)

# Aurora Token (AOA) Security Audit

Are Your Funds Safe?
During July 2019, Callisto Network conducted Aurora (AOA) smart contract's audit.
Callisto Network
Aurora
#AuditedbyCLO

## Aurora Token (AOA) security audit, conducted by the Callisto Network Security Department in July 2019.

## Aurora (AOA) Specificities

### Audit Request

Audit Top 200 CoinMarketCap tokens.

Aurora (AOA).

https://www.aurorachain.io/ (https://www.aurorachain.io/)

**Deployed at:**

- https://etherscan.io/address/0x9ab165d795019b6d8b3e971dda91071421305e5a#contracts (https://etherscan.io/address/0x9ab165d795019b6d8b3e971dda91071421305e5a#contracts)

**Source Code:**

- https://etherscan.io/address/0x9ab165d795019b6d8b3e971dda91071421305e5a#contracts (https://etherscan.io/address/0x9ab165d795019b6d8b3e971dda91071421305e5a#contracts)

**Disclosure policy:**

Public.

**Platform:**

ETH.

**Number of lines:**

66.

# Aurora (AOA) Smart Contract Security Audit Report

*Are Your Funds Safe?*

## 1. In scope

- ArthurStandardToken (https://etherscan.io/address/0x9ab165d795019b6d8b3e971dda91071421305e5a#contracts).

## 2. Findings

In total, **5 issues** were reported including:

- 2 medium severity issues.
- 3 low severity issues.

No critical security issues were found.

### 2.1. Transfer and TransferFrom (ERC20 Compliance)

**Severity: medium.**

**Description:**

As per ERC20 standard (https://eips.ethereum.org/EIPS/eip-20) “The function SHOULD throw if the message caller’s account balance does not have enough tokens to spend” however the implemented `transfer` and `transferFrom` functions returns false instead of throwing the transaction if the user doesn’t have enough of fund.

In the other hand dapp developers should also be aware that as per ERC20 standard “Callers MUST handle false from returns (bool success). Callers MUST NOT assume that false is never returned!”, meaning that `transfer` and `transferFrom` functions is returning false which it shouldn’t but the dapp developers must not assume that false is never returned.

**Code Snippet:**

```
    function transfer(address _to, uint256 _value) public returns (bool success) {
        if (balances[msg.sender] >= _value && _value > 0) {
            balances[msg.sender] -= _value;
            balances[_to] += _value;
            Transfer(msg.sender, _to, _value);
            return true;
        } else { return false; }
    }
```

```
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
        //same as above. Replace this line with the following if you want to protect against wrapping uints.
        //if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && balances[_to] + _value > balances[_t
o]) {
        if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && _value > 0) {
            balances[_to] += _value;
            balances[_from] -= _value;
            allowed[_from][msg.sender] -= _value;
            Transfer(_from, _to, _value);
            return true;
        } else { return false; }
    }
```

## 2.2. Transfer of Zero Value (ERC20 Compliance)

**Severity: medium.**

**Description:**
As per ERC20 standard (https://eips.ethereum.org/EIPS/eip-20) “Transfers of 0 values MUST be treated as normal transfers and fire the Transfer event” both `transfer` and `transferFrom` ignore transfers of zero value and return false, instead of threating it as normal transfers.

**Code snippet:**

```
function transfer(address _to, uint256 _value) public returns (bool success) {
        if (balances[msg.sender] >= _value && _value > 0) {
            balances[msg.sender] -= _value;
            balances[_to] += _value;
            Transfer(msg.sender, _to, _value);
            return true;
        } else { return false; }
    }
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
        //same as above. Replace this line with the following if you want to protect against wrapping uints.
        //if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && balances[_to] + _value > balances[_t
o]) {
        if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && _value > 0) {
            balances[_to] += _value;
            balances[_from] -= _value;
            allowed[_from][msg.sender] -= _value;
            Transfer(_from, _to, _value);
            return true;
        } else { return false; }
    }
```

## 2.3. Transfer to Address(0)

### Severity: low.

### Description:

`transfer` and `transferFrom` allow the destination address to be equal to zero, meaning that users fund can be lost if sent to it by mistake.

### Code snippet:

```
 function transfer(address _to, uint256 _value) public returns (bool success) {
        if (balances[msg.sender] >= _value && _value > 0) {
            balances[msg.sender] -= _value;
            balances[_to] += _value;
            Transfer(msg.sender, _to, _value);
            return true;
        } else { return false; }
    }
```

```
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
        //same as above. Replace this line with the following if you want to protect against wrapping uints.
        //if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && balances[_to] + _value > balances[_t
o]) {
        if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && _value > 0) {
            balances[_to] += _value;
            balances[_from] -= _value;
            allowed[_from][msg.sender] -= _value;
            Transfer(_from, _to, _value);
            return true;
        } else { return false; }
    }
```

## 2.4. Constructor Transfer Event (ERC20 Compliance)

### Severity: low.

### Description:

`Transfer` event is not emitted when the initial fund are assigned to the msg.sender inside the constructor.

ERC20 (https://eips.ethereum.org/EIPS/eip-20): “A token contract which creates new tokens SHOULD trigger a Transfer event with the _from address set to 0x0 when tokens are created.”

**Code snippet:**

```
    function ArthurStandardToken(
         uint256 _initialAmount,
         string _tokenName,
         uint8 _decimalUnits,
         string _tokenSymbol
         ) public {

         balances[msg.sender] = _initialAmount;               // Give the creator all initial tokens
         totalSupply = _initialAmount;                        // Update total supply
         name = _tokenName;                                   // Set the name for display purposes
         decimals = _decimalUnits;                            // Amount of decimals for display purposes
         symbol = _tokenSymbol;                               // Set the symbol for display purposes
     }
```

## 2.5. Known vulnerabilities of ERC-20 token

**Severity: low.**

**Description:**

1. It is possible to double withdrawal attack. More details here (https://docs.google.com/document/d/1YLPtQxZu1UAvO9cZ1O2RPXBbT0mooh4DYKjA_jp-RLM/edit)
2. Lack of transaction handling mechanism issue. WARNING! (https://gist.github.com/Dexaran/ddb3e89fe64bf2e06ed15fbd5679bd20) This is a very common issue and it already caused millions of dollars losses for lots of token users! More details here (https://docs.google.com/document/d/1Feh5sP6oQL1-1NHi-X1dbgT3ch2WdhbXRevDN681Jv4/edit)

## 3. Conclusion

ERC20 Compliance issues should be solved.

## 4. Revealing audit reports

- https://gist.github.com/yuriy77k/4b8bf6b07d896a18c75237bbefd5022b (https://gist.github.com/yuriy77k/4b8bf6b07d896a18c75237bbefd5022b)
- https://gist.github.com/yuriy77k/1139b269b38e984a8189a9713e8ea051 (https://gist.github.com/yuriy77k/1139b269b38e984a8189a9713e8ea051)
- https://gist.github.com/yuriy77k/ddd0f9b86ad2856c9f53dac3a220975e (https://gist.github.com/yuriy77k/ddd0f9b86ad2856c9f53dac3a220975e)

# Appendix

Smart Contract Audits by Callisto Network. (https://callisto.network/smart-contract-audit/)

## Miscellaneous

Why Audit Smart Contracts? (https://callisto.network/why-audit-smart-contracts/)

Our Most Popular Audit Reports. (https://callisto.network/security-audits/)

## Trust the Blockchain, Audit the Smart Contracts.

*Follow Callisto's Security Department on Twitter (https://twitter.com/Callisto_Audits) to get our latest news and updates!*

Published on **November 4, 2020**

(https://(https://(https://(https://(https://(https://callisto.network/aurora-

token- token- token- token- token- token-

aoa- aoa- aoa- aoa- aoa- aoa-

security-security-security-security-security-security-

audit/) audit/) audit/) audit/) audit/) audit/)

Security Audits (https://callisto.network/tag/security-audits/)

‹ Previous post (https://callisto.network/earn-crypto-in-3-quick-steps/) Next post ›

Callisto Network LTD

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

Join Our Community

(https://t.me/CallistoNet) (https://twitter.com/CallistoSupport) (http://reddit.com/r/CallistoCrypto) (https://www.youtube.com/channel/UC1WMae32v_eJ8qOtLQ

(https://www.instagram.com/callisto.network/) (https://www.facebook.com/callistonetwork) (https://www.linkedin.com/company/callisto-network/) (https://t.co/DAWunSR1tm)

## Resources

FAQ (https://callisto.network/faq/)

Timeline (https://callisto.network/timeline/)

Airdrop (https://callisto.network/callisto-airdrop/)

Community Guidelines (https://callisto.network/community-guidelines/)

## Callisto

Partners (https://callisto.network/partners/)

Our GitHub repositories (https://github.com/EthereumCommonwealth)

Media Kit (https://github.com/EthereumCommonwealth/Callisto-Media-Kit)

Contact us (https://callisto.network/contact-us/)

Want to sell your CLO coins OTC? (mailto:vladimir.vencalek@invictussolutions.cz)