Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

September, 2021

Audit Details

Audited project

ASSX Token

Deployer address

0x1723aD8207049ee668111D36d22235eF273529D0

Client contacts:

ASSX Token team

Blockchain

Ethereum

Project website:

www.assx.me

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 below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. 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 TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate 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 TechRate 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, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, 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 (whether innocent or negligent) 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.

Background

TechRate was commissioned by ASSX Token to perform an audit of smart contracts:

https://etherscan.io/address/0x55250813d5c4bc373fc17022e6ff8a3551990176#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

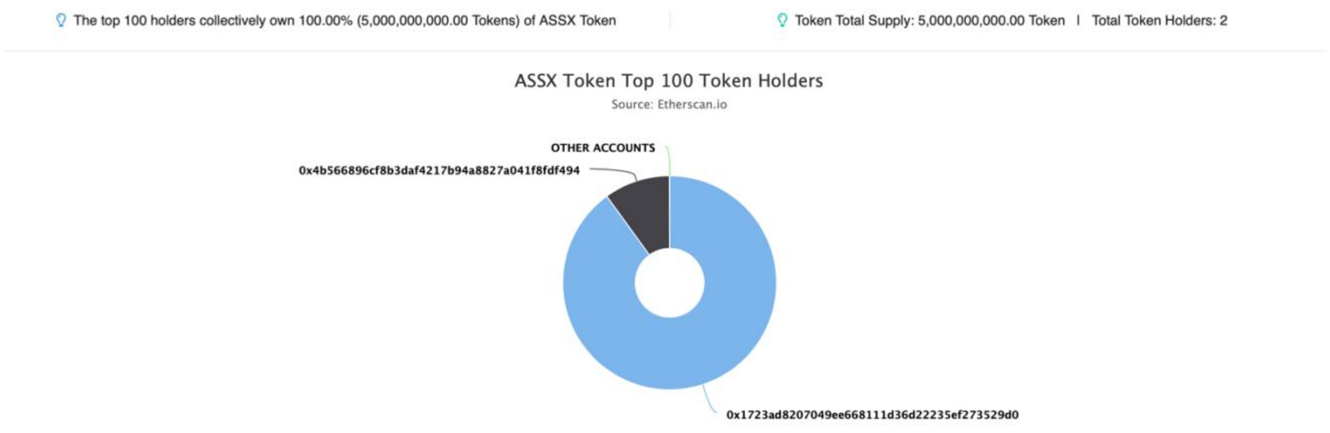
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 16.09.2021

Contract name	ASSX Token
Contract address	0x55250813D5C4bc373FC17022E6FF8a3551990176
Total supply	5,000,000,000
Token ticker	ASSX
Decimals	18
Token holders	2
Transactions count	2
Top 100 holders dominance	100%
Contract deployer address	0x1723aD8207049ee668111D36d22235eF273529D0
Contract’s current owner address	0x1723aD8207049ee668111D36d22235eF273529D0

FIA Protocol Token Distribution



The top 100 holders collectively own 100.00% (5,000,000,000.00 Tokens) of ASSX Token

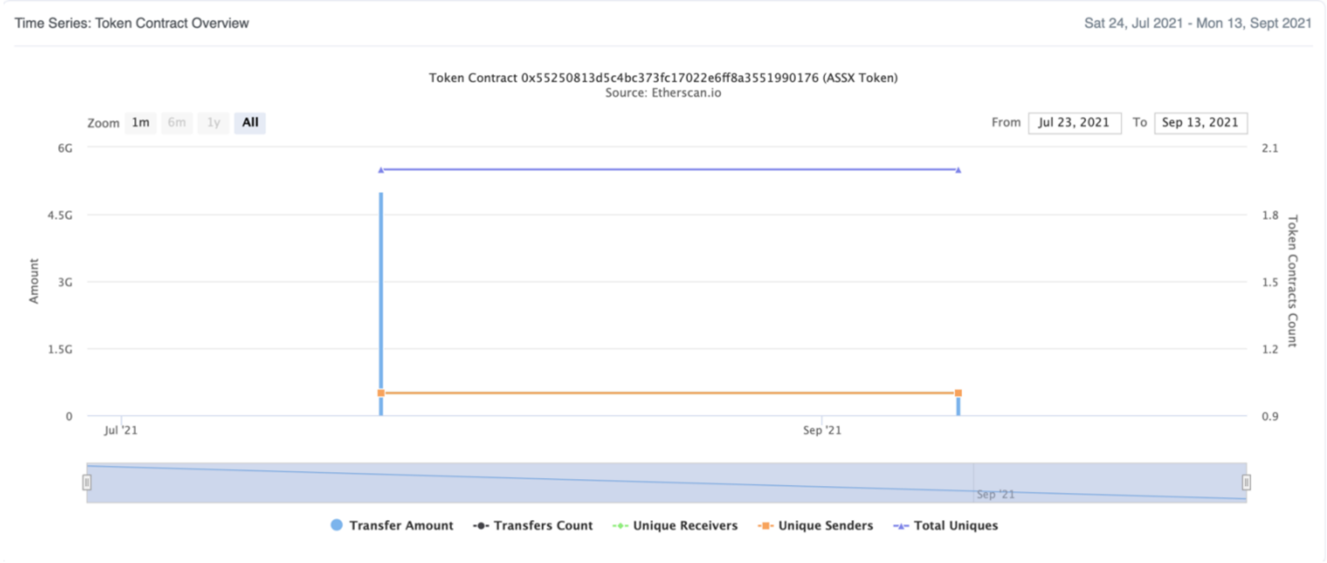
Token Total Supply: 5,000,000,000.00 Token | Total Token Holders: 2

ASSX Token Top 100 Token Holders

Source: Etherscan.io

OTHER ACCOUNTS
0x4b566896cf8b3daf4217b94a8827a041f8fdf494
0x1723ad8207049ee668111d36d22235ef273529d0

FIA Protocol Contract Interaction Details



Time Series: Token Contract Overview

Sat 24, Jul 2021 - Mon 13, Sept 2021

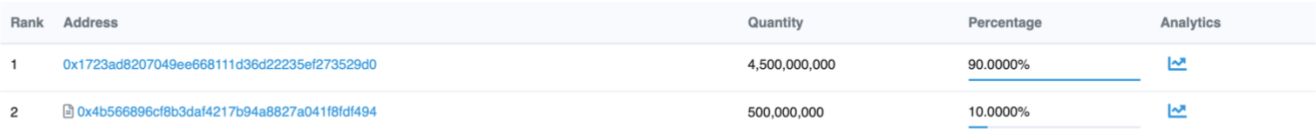
Token Contract 0x55250813d5c4bc373fc17022e6ff8a3551990176 (ASSX Token)

Source: Etherscan.io

Zoom 1m 6m 1y All

From Jul 23, 2021 To Sep 13, 2021

FIA Protocol Top 10 Token Holders



Rank	Address	Quantity	Percentage	Analytics
1	0x1723ad8207049ee668111d36d22235ef273529d0	4,500,000,000	90.0000%	📈
2	0x4b566896cf8b3daf4217b94a8827a041f8fdf494	500,000,000	10.0000%	📈

Contract functions details

```
+  Context
  - [Int] _msgSender
  - [Int] _msgData

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] div
  - [Int] mod
  - [Int] mod

+  ERC20 (Context, IERC20)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Int] _transfer #
  - [Int] _mint #
  - [Int] _burn #
  - [Int] _approve #
  - [Int] _setupDecimals #
  - [Int] _beforeTokenTransfer #

+  Ownable (Context)
  - [Int] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner

+  Authorizable (Ownable)
```

  - [Pub] addAuthorized #
    - modifiers: onlyOwner
  - [Pub] removeAuthorized #
    - modifiers: onlyOwner

+ ASSXToken (ERC20, Ownable, Authorizable)
  - [Pub] <Constructor> #
  - [Pub] cap
  - [Pub] capUpdate #
    - modifiers: onlyAuthorized
  - [Pub] lockFromUpdate #
    - modifiers: onlyAuthorized
  - [Pub] lockToUpdate #
    - modifiers: onlyAuthorized
  - [Pub] unlockedSupply
  - [Pub] lockedSupply
  - [Pub] circulatingSupply
  - [Pub] totalLock
  - [Int] _beforeTokenTransfer #
  - [Int] _transfer #
  - [Pub] mint #
    - modifiers: onlyOwner
  - [Pub] manualMint #
    - modifiers: onlyAuthorized
  - [Pub] totalBalanceOf
  - [Pub] lockOf
  - [Pub] lastUnlockBlock
  - [Pub] lock #
    - modifiers: onlyOwner
  - [Pub] canUnlockAmount
  - [Pub] unlock #
  - [Pub] transferAll #
  - [Ext] delegates
  - [Ext] delegate #
  - [Ext] delegateBySig #
  - [Ext] getCurrentVotes
  - [Ext] getPriorVotes
  - [Int] _delegate #
  - [Int] _moveDelegates #
  - [Int] _writeCheckpoint #
  - [Int] safe32
  - [Int] getChainId

($) = payable function
# = non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Low issue
19.	Cross-function race conditions.	Passed

20.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

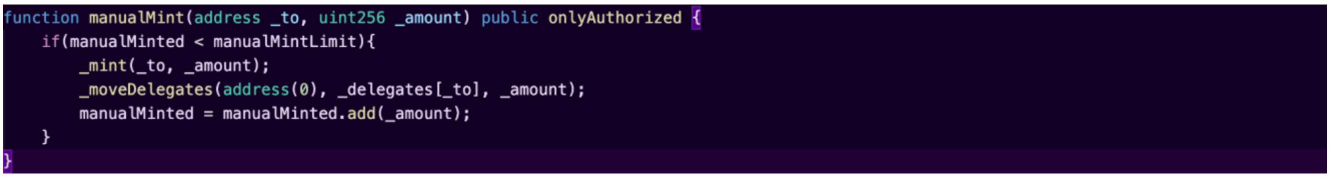
No medium severity issues found.

Low Severity Issues

1. Manual mint limit check

Issue:

- Manual mint limit check should be done before adding tokens.



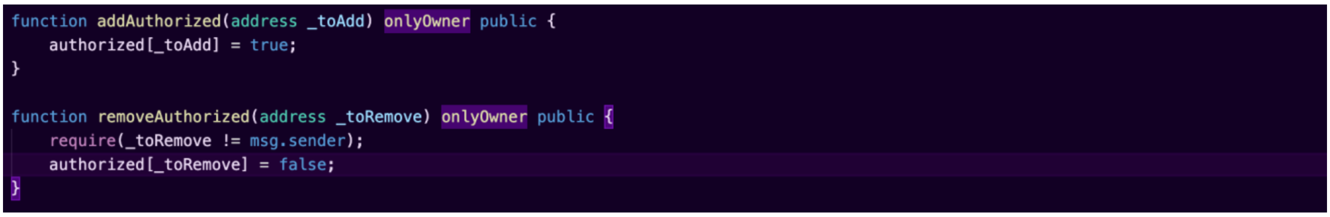
```
function manualMint(address _to, uint256 _amount) public onlyAuthorized {
    if(manualMinted < manualMintLimit){
        _mint(_to, _amount);
        _moveDelegates(address(0), _delegates[_to], _amount);
        manualMinted = manualMinted.add(_amount);
    }
}
```

Recommendation:

Check manual mint before add tokens.

Owner privileges (In the period when the owner is not renounced)

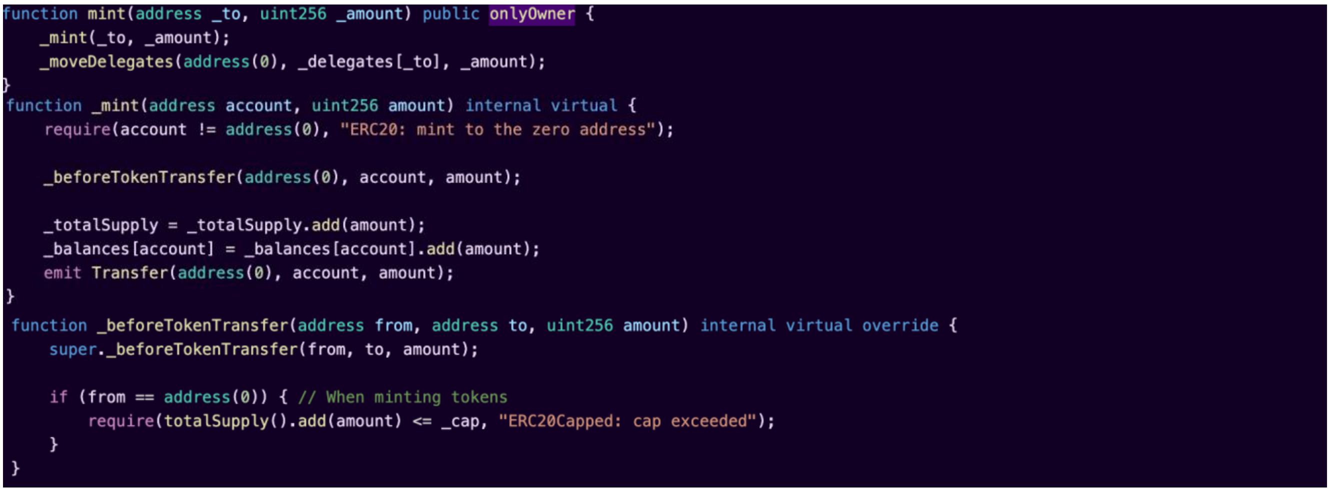
- Owner can add / remove authorized user.



```
function addAuthorized(address _toAdd) onlyOwner public {
    authorized[_toAdd] = true;
}

function removeAuthorized(address _toRemove) onlyOwner public {
    require(_toRemove != msg.sender);
    authorized[_toRemove] = false;
}
```

- Owner can mint tokens (up to capitalization amount).



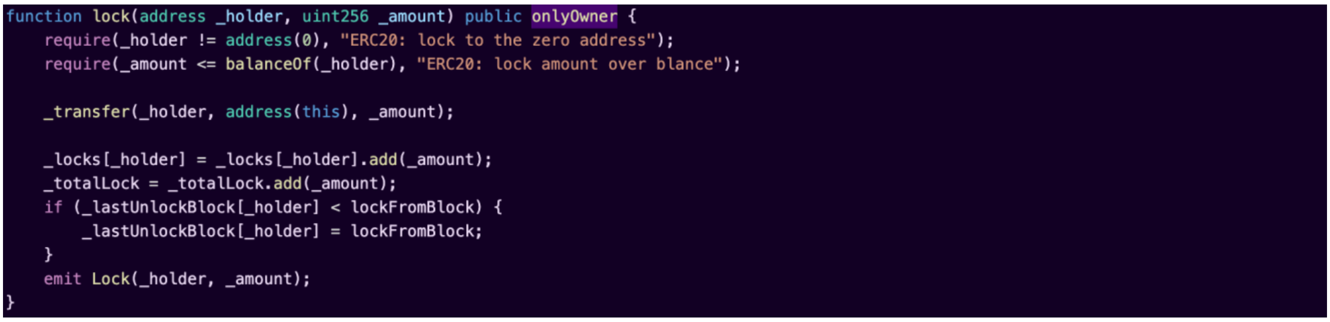
```
function mint(address _to, uint256 _amount) public onlyOwner {
    _mint(_to, _amount);
    _moveDelegates(address(0), _delegates[_to], _amount);
}
function _mint(address account, uint256 amount) internal virtual {
    require(account != address(0), "ERC20: mint to the zero address");

    _beforeTokenTransfer(address(0), account, amount);

    _totalSupply = _totalSupply.add(amount);
    _balances[account] = _balances[account].add(amount);
    emit Transfer(address(0), account, amount);
}
function _beforeTokenTransfer(address from, address to, uint256 amount) internal virtual override {
    super._beforeTokenTransfer(from, to, amount);

    if (from == address(0)) { // When minting tokens
        require(totalSupply().add(amount) <= _cap, "ERC20Capped: cap exceeded");
    }
}
```

- Owner can lock tokens of any user.



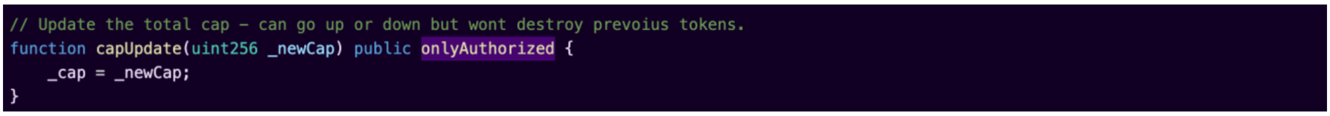
```
function lock(address _holder, uint256 _amount) public onlyOwner {
    require(_holder != address(0), "ERC20: lock to the zero address");
    require(_amount <= balanceOf(_holder), "ERC20: lock amount over blance");

    _transfer(_holder, address(this), _amount);

    _locks[_holder] = _locks[_holder].add(_amount);
    _totalLock = _totalLock.add(_amount);
    if (_lastUnlockBlock[_holder] < lockFromBlock) {
        _lastUnlockBlock[_holder] = lockFromBlock;
    }
    emit Lock(_holder, _amount);
}
```

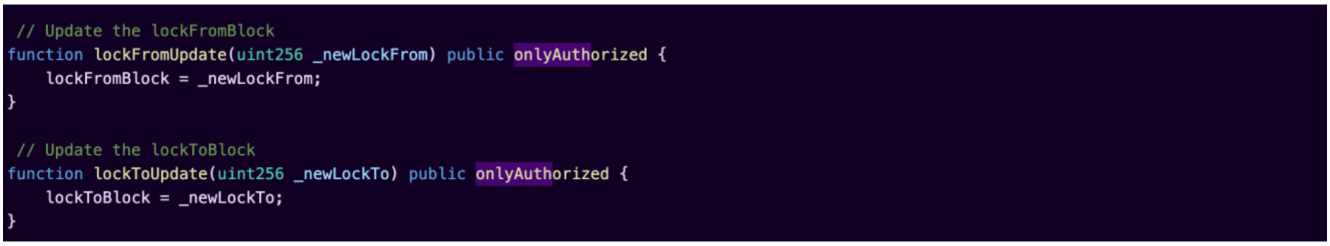
Authorized privileges (owner is authorized user)

- Authorized can update capitalization amount.



```
// Update the total cap - can go up or down but wont destroy prevouis tokens.
function capUpdate(uint256 _newCap) public onlyAuthorized {
    _cap = _newCap;
}
```

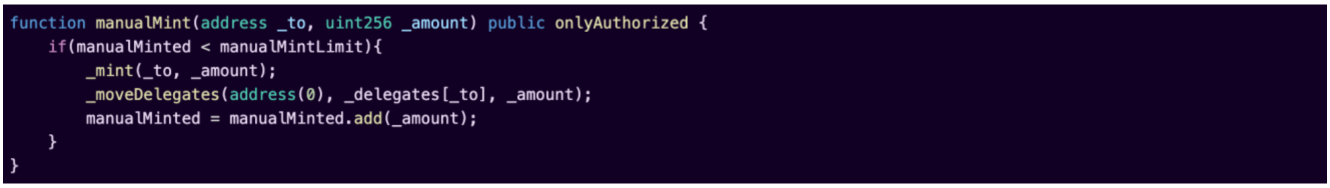
- Authorized can update lock from block / lock to block values.



```
// Update the lockFromBlock
function lockFromUpdate(uint256 _newLockFrom) public onlyAuthorized {
    lockFromBlock = _newLockFrom;
}

// Update the lockToBlock
function lockToUpdate(uint256 _newLockTo) public onlyAuthorized {
    lockToBlock = _newLockTo;
}
```

- Authorized can mint tokens (up to 5,000,000,000).



```
function manualMint(address _to, uint256 _amount) public onlyAuthorized {
    if(manualMinted < manualMintLimit){
        _mint(_to, _amount);
        _moveDelegates(address(0), _delegates[_to], _amount);
        manualMinted = manualMinted.add(_amount);
    }
}
```

Conclusion

Smart contracts contain low severity issue and owner / authorized user privileges!

TechRate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.