

Smart Contract Code Review and Security Analysis Report

Customer: Oxcert Date: June 29, 18 This document contains confidential information about IT systems and intellectual properties of the customer, as well as information about potential vulnerabilities and methods of their exploitation.

This confidential information is for internal use by the customer only and shall not be disclosed to third parties.

Document:

	Smart Contract Code Review and Security Analysis Report for Oxcert
Date:	29.06.2018



Table of contents

Introduction	3
Scope	3
Executive Summary	
Severity Definitions	
AS-IS overview	
Audit overview Conclusion	
Disclaimers	
Appendix A. Evidences	11
Appendix B. Automated tools reports	12



Introduction

Hacken OÜ (Consultant) was contracted by 0xcert (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of Customer's smart contract and its code review conducted between June 19th, 2018 - June 29th, 2018.

Scope

The scope of the project is 0xcert smart contracts, which can be found on github by links below:

- Zxc Token https://github.com/0xcert/ethereum-zxc/tree/master/contracts
 - o Commit 1fafdebea4ae77ebc5dac574485b8f28c3ea93f7
- Xcert https://github.com/0xcert/ethereum-xcert/tree/master/contracts
 - o Commit bea01c566327fc98fa077ba92ef593548efc7365
- Crowdsale https://github.com/0xcert/ethereum-crowdsale/tree/master/contracts
 - o Commit 3a11ff95121f6a71e0052fd95a9a23a88b622499

The full version of tested and audited contracts is: ERC165, ERC20, ERC721, ERC721Enumerable, ERC721Metadata, ERC721TokenReceiver, NFToken, NFTokenEnumerable, NFTokenMetadata, Ownable, SafeMath, SupportsInterface, Token, Xcert, RevokableXcert, PausableXcert, MutableXcert, BurnableXcert, Selector, Zxc, ZxcCrowdsale.

Migrations.sol and mocks contracts used in repositories are not in scope of the project.

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are considered (the full list includes them but is not limited to them):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Byte array vulnerabilities
- Style guide violation
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed



- Unchecked external call Unchecked math
- Unsafe type inference
- Implicit visibility level

Executive Summary

According to the assessment, Customer's smart contracts are well secured and no major fixes are required.

Our team performed analysis of code functionality, manual audit and automated checks with solc, Mythrill and remix IDE (see Appendix B pic 1-29). All found issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in Audit Overview section. General overview is presented in AS-IS section and all found issues can be found in Audit overview section.

We found no security issues; however, some informational statements and code style guide violations were found. They can't have security or functional impact, but they should be analyzed by Customer and its users.

Graph 1. Vulnerabilities distribution





Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens lose etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution
Lowest / Code Style / Info	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

AS-IS overview

ZXC contract overview

Zxc.sol is contract, which describes standard ERC20 (inherits from Token, Ownable) token for 0xcert project.

Zxc contract constructor sets:

- tokenName to 0xcert Protocol Token
- tokenSymbol to ZXC
- tokenDecimals to 18
- transferEnabled to false
- balances[owner] to tokenTotalSupply

Zxc contract determines next functions:

- with validDestination modifier guarantees that _to address is a valid destination to transfer tokens to and with onlyWhenTransferAllowed modifier – checks if tokens can be transferred
 - o transfer, transferFrom functions from ERC20 standard
- external with onlyOwner modifier—allows function call to owner address:
 - o enableTransfer allows owner account to send tokens
 - o burn allows owner account to burn tokens
 - o setCrowdsaleAddress allows owner account to set crowdsaleAddress address



XCert contracts overview

XCert is ERC721 extension that describes customized non-fungible token.

XCert inherits from NFTokenEnumerable and NFTokenMetadata contracts; NFTokenEnumerable inherits from NFToken contract and ERC721Enumerable interface; NFTokenMetadata inherits from NFToken contract and ERC721Metadata interface; NFToken inherits from Ownable contract, ERC721 interface, SupportsInterface contract (which inherits from ERC165 interface).

Xcert contract determines next parameters:

- nftConventionId ID which determines each Xcert smart contract type by its JSON convention
- idToProof mapping maps token id to token proof
- config mapping maps token id to token config protocol
- data mapping maps token id to token convention data
- addressToAuthorized mapping maps address to bool (true if address can mint new tokens)

Xcert contract determines modifier is Authorized that requires msg.sender to be owner or address To Authorized [msg.sender] is true.

Xcert contract constructor sets supportedInterfaces[0x6be14f75] to true.

Xcert contract determines next functions:

- external with is Authorized modifier:
 - o mint allows address to min new token
- external with onlyOwner modifier:
 - o setAuthorizedAddress allows owner to set addressToAuthorized[address] to true
- external view:
 - o conventionId allows to get a bytes4 of keccak256 of json schema representing 0xcert protocol convention
 - o tokenProof allows to get proof for token with given token id
 - o tokenExpirationTime allows to get expiration time for token with given token id
 - o isAuthorizedAddress allows to check whether address is authorized to mint tokens
- public view:
 - o tokenDataValue allows to get convention data for token with given token id

There are variations of Xcert in repository:

- BurnableXcert Xcert token implementation where tokens can be burnt by contract operator
- MutableXcert Xcert token implementation where data can be changed
- PausableXcert XCert token implementation where tokens can be paused
- RevokableXcert Xcert token implementation where tokens can be burnt by issuer



ZxcCrowdsale contract overview

ZxcCrowdsale contract is used for distributing ZXC tokens.

ZxcCrowdsale uses parameters Zxc token and Xcert xcertKyc to call functions in token and xcertKyc contracts. Other ZxcCrowdsale parameters are:

- startTimePresale presale start time
- startTimeSaleWithBonus token sale with bonus start time
- startTimeSaleNoBonus token sale without bonus start time
- bonusPresale percent amount for bonus on presale
- bonusSale percent amount for bonus on crowdsale
- endTime crowdsale end time
- minimumPresaleWeiDeposit minimum deposit in wei for presale
- preSaleZxcCap number of Zxc tokens available on presale
- crowdSaleZxcSupply number of Zxc tokens available on crowdsale
- zxcSold number of Zxc sold during the crowdsale
- wallet address for received funds
- rate number of tokens received per 1 wei

Xcert contract constructor sets wallet, Zxc token address, Xcert xcertKyc address, startTimePresale, startTimeSaleWithBonus, startTimeSaleNoBonus, endTime, rate, preSaleZxcCap, crowdSaleZxcSupply, bonusPresale, bonusSale, minimumPresaleWeiDeposit to values specified by ZxcCrowdsale contract owner.

ZxcCrowdsale contract determines next functions:

- external payable:
 - o fallback function calls buyTokens function
- public payable:
 - o buyTokens allows to purchase Zxc tokens
- external view:
 - o hasEnded checks whether crowdsale is ended (true if ended)
- internal view:
 - o isInTimeRange checks whether current period is in crowdsale period
 - getTokenAmount calculates number of tokens will be received by investor for given amount of wei and given bonus percent



Audit overview

Critical

No critical vulnerabilities were found.

High

No critical vulnerabilities were found.

Medium

No medium severity vulnerabilities were found.

Low

No low severity vulnerabilities were found.

Lowest / Code style / Info

Informational statements

Informational statements are audit team findings that doesn't have any security issues. However, they are presented in report to clarify and outline functionality and business requirements.

- 1. XCert config is not changeable by contract owner or XCert owner. For instance, digital asset ExpirationTime can't be prolongated or config items can't be removed or added.
- 2. tokenExpirationTime is stored as first item of config array and it takes 32 bytes to be stored what is much more than needed for timestamp. It can be optmized in different ways to have some gas economy.
- 3. Indexes are changed after token burned/revoked. XCert contract operate with both Ids and indexes. If DApps that operate with burnable/revokable XCert and token indexes are changed after each burn and revoke, that DApps can potentially have some errors or inconveniencies in development.



Code Style Issues

Code style issues are related to Solidity Code Style Best Practices, however, they and can't have any security or functional impact.

4. Redundant code in ZxcCrowdsale.sol (lines 259-264, see Appendix A pic 1 for evidence).

```
if (now >= _startTime && now < _endTime) {
    return true;
}
else {
    return false;
}
// It can be rewritten as presented below

return(now >= _startTime && now < _endTime);</pre>
```

Consider changing code as presented above.



Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. For these contracts high level description of functionality was presented in As-is overview section of the report.

Audit team haven't found any security issues during both manual and automated audits and the report contains informational statements and code style notice related to the reviewed code.

Overall quality of reviewed contracts is high and no fixes are required.

Disclaimers

Disclaimer

The audit does not give any warranties on the security of the code. One audit cannot be considered enough. We always recommend proceeding to several independent audits and a public bug bounty program to ensure the security of the smart contracts.

Technical Disclaimer

Smart contract build on the top of Ethereum blockchain means that a lot of features could be covered by tests, but Turing completeness of Solidity programming language realization leaves some space for unexpected runtime exceptions.



Appendix A. Evidences

Pic 1. Redundant code in ZxcCrowdsale.sol:

```
function isInTimeRange(
252
         uint256 _startTime,
253
         uint256 _endTime
254
255
         internal
256
         view
257
         returns(bool)
258 +
259 +
         if (now >= _startTime && now < _endTime) {</pre>
260
          return true;
261
262 ₹
         else {
263
         return false;
264
265
266
```



Appendix B. Automated tools reports

Pic. 1 Solc ZxcCrowdsale automated report

max@max-VirtualBox:~/solidity/projects/0xcert\$ solc -o . --bin --abi --overwrite ZxcCrowdsale.sol
max@max-VirtualBox:~/solidity/projects/0xcert\$

Pic. 2 Mythrill ZxcCrowdsale automated report

Pic. 3 Mythrill ZxcCrowdsale automated report

Pic. 4 Solc Zxc automated report

max@max-VirtualBox:~/solidity/projects/0xcert\$ solc -o . --bin --abi --overwrite Zxc.sol
max@max-VirtualBox:~/solidity/projects/0xcert\$



Pic. 5 Mythrill Zxc automated report

Pic. 6 Solc Selector automated report

max@max-VirtualBox:~/solidity/projects/0xcert\$ solc -o . --bin --abi --overwrite Selector.sol
max@max-VirtualBox:~/solidity/projects/0xcert\$

Pic. 7 Mythrill Selector automated report

max@max-VirtualBox:~/solidity/projects/0xcert\$ myth -x Selector.sol
The analysis was completed successfully. No issues were detected.

max@max-VirtualBox:~/solidity/projects/0xcert\$

Pic. 8 Solc BurnableXcert automated report

max@max-VirtualBox:~/solidity/projects/0xcert\$ solc -o . --bin --abi --overwrite BurnableXcert.sol
max@max-VirtualBox:~/solidity/projects/0xcert\$



Pic. 9 Mythrill BurnableXcert automated report

Pic. 10 Mythrill BurnableXcert automated report

Pic. 11 Solc MutableXcert automated report

```
max@max-VirtualBox:~/solidity/projects/0xcert$ solc -o . --bin --abi --overwrite MutableXcert.sol
max@max-VirtualBox:~/solidity/projects/0xcert$
```



Pic. 12 Mythrill MutableXcert automated report

```
nax@max-VirtualBox:~/solidity/projects/0xcert$ myth -x MutableXcert.sol
==== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: setTokenData(uint256,bytes32[])
PC address: 1584
A possible integer overflow exists in the function `setTokenData(uint256,bytes32[])`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: MutableXcert.sol:36
function setTokenData(
uint256 _tokenId,
bytes32[] _data
      validNFToken(_tokenId)
isAuthorized()
      external
     data[_tokenId] = _data;
emit TokenDataChange(_tokenId, _data);
==== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: setTokenData(uint256,bytes32[])
PC address: 1590
A possible integer overflow exists in the function `setTokenData(uint256,bytes32[])`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: MutableXcert.sol:36
function setTokenData(
      uint256 _tokenId,
bytes32[] _data
      validNFToken(_tokenId)
isAuthorized()
     data[_tokenId] = _data;
emit TokenDataChange(_tokenId, _data);
```



Pic. 13 Mythrill MutableXcert automated report

```
Type: Warning

Type:
```

Pic. 14 Mythrill MutableXcert automated report

```
Type: Informational
Contract: Unknown
Function name: fallback
PC addres: 12294
A reachable exception (opcode 0xfe) has been detected. This can be caused by type errors, division by zero, out-of-bounds array access, or assert violations. This is acceptable in most situations. Note however that 'sasert()' should only be used to check invariants. Use 'require()' for regular input checking.

In file: SafeMath.sol:80

assert(c >= _a)
```



Pic. 15 Mythrill MutableXcert automated report

```
=== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: setTokenData(uint256,bytes32[])
PC address: 12462
A possible integer overflow exists in the function `setTokenData(uint256,bytes32[])`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: MutableXcert.sol:8
contract MutableXcert is Xcert {
   * @dev Emits when an Token data is changed.
* @param _id NFT that data got changed.
* @param _data New data.
  event TokenDataChange(
     uint256 indexed _id,
     bytes32[] _data
   * @dev Contract constructor.
   * @notice When implementing this contract don't forget to set nftConventionId, nftName and
    * nftSymbol.
   */
  constructor()
     public
     supportedInterfaces[0x59118221] = true; // MutableXcert
   st @dev Modifies convention data by setting a new value for a given index field.
   \star @param _tokenId Id of the NFT we want to set key value data.
      @param _data New token data.
  function setTokenData(
     uint256 _tokenId,
bytes32[] _data
     validNFToken(_tokenId)
     isAuthorized()
     external
     data[_tokenId] = _data;
     emit TokenDataChange(_tokenId, _data);
```



Pic. 16 Mythrill MutableXcert automated report

```
==== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: setTokenData(uint256,bytes32[])
PC address: 12464
A possible integer overflow exists in the function `setTokenData(uint256,bytes32[])`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: MutableXcert.sol:8
contract MutableXcert is Xcert {
   * @dev Emits when an Token data is changed.
   * @param _id NFT that data got changed.
   * @param _data New data.
  event TokenDataChange(
   uint256 indexed _id,
    bytes32[] _data
  /**
   * @dev Contract constructor.
   * @notice When implementing this contract don't forget to set nftConventionId, nftName and
   * nftSymbol.
  constructor()
   public
    supportedInterfaces[0x59118221] = true; // MutableXcert
   * @dev Modifies convention data by setting a new value for a given index field.
   * @param _tokenId Id of the NFT we want to set key value data.
    @param _data New token data.
  function setTokenData(
   uint256 _tokenId,
bytes32[] _data
    validNFToken(_tokenId)
    isAuthorized()
    external
    data[_tokenId] = _data;
    emit TokenDataChange(_tokenId, _data);
max@max-VirtualBox:~/solidity/projects/0xcert$
```

Pic. 17 Solc PausableXcert automated report

```
max@max-VirtualBox:~/solidity/projects/0xcert$ solc -o . --bin --abi --overwrite PausableXcert.sol
max@max-VirtualBox:~/solidity/projects/0xcert$
```



Pic. 18 Mythrill PausableXcert automated report

```
max@max-VirtualBox:~/solidity/projects/0xcert$ myth -x PausableXcert.sol
==== Integer Overflow =====
Type: Warning
Contract: Unknown
Function name: fallback
PC address: 8676
A possible integer overflow exists in the function `fallback`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: Xcert.sol:1
import "./SafeM
==== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: fallback
PC address: 11335
A possible integer overflow exists in the function `fallback`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: Xcert.sol:1
import "./SafeM
 ==== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: fallback
PC address: 12186
A possible integer overflow exists in the function 'fallback'.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: SafeMath.sol:79
_a + _b
```

Pic. 19 Mythrill PausableXcert automated report

Pic. 20 Solc RevokableXcert automated report

```
max@max-VirtualBox:~/solidity/projects/0xcert$ solc -o . --bin --abi --overwrite RevokableXcert.sol
max@max-VirtualBox:~/solidity/projects/0xcert$
```



Pic. 21 Mythrill RevokableXcert automated report

```
Assigner-Virtualisor-/solidity/projects/0xcerts myth -x Revokablexcert.sol
Type: Marcing
Type: Marci
```

Pic. 22 Mythrill RevokableXcert automated report

Pic. 23 Solc Xcert automated report

```
max@max-VirtualBox:~/solidity/projects/0xcert$ solc -o . --bin --abi --overwrite Xcert.sol
max@max-VirtualBox:~/solidity/projects/0xcert$
```



Pic. 24 Mythrill Xcert automated report

```
max@max-VirtualBox:~/solidity/projects/0xcert$ myth -x Xcert.sol
==== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: fallback
PC address: 8303
A possible integer overflow exists in the function `fallback`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: Xcert.sol:1
import "./SafeM
==== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: fallback
PC address: 10937
A possible integer overflow exists in the function `fallback`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: Xcert.sol:1
import "./SafeM
==== Integer Overflow ====
Type: Warning
Contract: Unknown
Function name: fallback
PC address: 11788
A possible integer overflow exists in the function `fallback`.
The addition or multiplication may result in a value higher than the maximum representable integer.
In file: SafeMath.sol:79
_a + _b
```

Pic. 25 Mythrill Xcert automated report

```
mass Exception state asset Type: Informational Contract: Unknown Facility Contract C
```



Pic. 26 List of audited contracts with Remix IDE static analysis:





Pic. 26 Remix IDE static analysis report part 1:

7 1 1	
Potential Violation of Checks-Effects-Interaction pattern in <>>zvcCrowdsale (address address	×
Potential Violation of Checks-Effects-Interaction pattern in <i->ZxcCrowdsale buyTokens() Could potentially lead to re-entrancy vulnerability. <i->br/>->Note Modifiers are currently not considered by this static analysis.</i-></i->	×
browser/AddressUtils.sol 28.5 CAUTION: The Contract uses inline assembly, this is only advised in rare cases. Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.	×
browser/ZvCcrowdsale sof 158.34 use of "now"; "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.	×
browser/ZivCcrondsale.sol 241:27 use of "now". "now" does not mean current time. Now is an alias for block timestamp. Block timestamp can be influenced by miners to a certain degree, be careful.	×
browser/ZxcCrowdsale.sol 259.9 use of "now". "now" does not mean current time. Now is an alias for block timestamp. Block timestamp can be influenced by miners to a certain degree, be careful.	×
browser/ZvcCrowdsale.sol 259.30 use of "now", "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.	×
Gas requirement of function NFToken safe TransferFrom/address address unit/256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage).	×
Gas requirement of function NFToken safe TransferFrom(address_address_unit256_b/yes) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage).	n)*
Gas requirement of function NFToken transferFrom(address, address, unit256) high: infinite. If the cas requirement of a function is higher than the block as limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage).	,,,
	Î
Gas requirement of function NFTokenEnumerable safeTransferFrom(address, address, uint256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	×
Gas requirement of function NFTokenEnumerable safeTransferFrom(address, address, unit256, bytes) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	×
Gas requirement of function NFTokenEnumerable transferFrom(address, address, unit256) high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	×
Gas requirement of function NFTokenNetadata name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	×
Gas requirement of function NFTokenMetadata safeTransferFrom(address_address_uint256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please evoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	×
Pic. 27 Remix IDE static analysis report part 2:	
ic. 27 Remix HDL state analysis report part 2.	
Gas requirement of function NETokenMatastats safeTranssferrom/address address; unit756 bytes high infinite. If the pas requirement of a function is higher than the block pas limit it cannot be executed. Please avoid loose in your functions or actions that modify large areas of storage (this includes clearing or conving arrays	×
Gas requirement of function NFTokenMetadata asfeTransferFrom(address_address_uint256_b/des) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	×
	×
in storage)	× ×
In storage) Gas requirement of function NFTokenMetadata symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	×
In storage) Gas requirement of function NFTokenMetadata symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata tokenUR(uirt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	×
In storage) Gas requirement of function NFTokenMetadata symboli) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata tokenURi(urit256) high: infinite. If the gas requirement or a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transferFrom(address address unit256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	× ×
In storage) Gas requirement of function NFTokenMetadata symboli) hight: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata tokenURi(unt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transferFrom(address, address, unt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	× ×
In storage) Gas requirement of function NFTokenMetadata symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata tokenUR(ivint256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transferFrom(address address unit256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token name) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)	× ×
In storage) Gas requirement of function NFTokenMetadata symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata tokenURI(uirt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transferFrom(address address uirt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer[coddress.uu/256] high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your function	× × × × ×
In storage) Gas requirement of function NFTokenMetadata symbol() high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transfer/rom(address, address, unit256) high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transfer/rom(address, address, unit256) high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token name() high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer(address, unit256) high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer(address, unit256) high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/configodress, address, unit256) high: Infinite. If the gas requirement of a function is higher than the bl	× × × × ×
In storage) Gas requirement of function NFTokenMetadata symbol) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata tokenUR(ivirt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token name) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token name) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token symbol) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer@address.um1256 high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer@address.um1256 high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large are	× × × × ×
In storage) Gas requirement of function NFTokenMetadata symbol) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata tokenUR(uirt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token samely high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token symbol) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/configodress, unit 256 high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/configodress, unit256 high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/configodress, unit256 high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your fu	× × × × × × × ×
In storage) Gas requirement of function NFTokenMetadata symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transferFrom(address, address, unit256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transferFrom(address, address, unit256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/(address, unit256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/(address, unit256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/confideres, unit256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Xcert name() high: infinite. If the gas requirement of a function is higher than the block gas l	× × × × × × × ×
In storage) Gas requirement of function NFTokenMetadata symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function NFTokenMetadata transferFrom(address, address, unt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/coddress, until the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/coddress, until 56) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Token transfer/coddress, until 56) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage) Gas requirement of function Xcert name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions	× × × × × × × × ×
Gas requirement of function NFTokenMetadata symbol() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function NFTokenMetadata toxensterFrom(address address unt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Token name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Token name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Token transfer/From(address unt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Zoen transfer/From(address unt256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Xoen transfer/From(address unt256, bring, litting to gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your	× × × × × × × ×
Case requirement of function NFTokenMetadata seymbo() high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function NFTokenMetadata toxinetPrompaddress_address_unitS95) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Token name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Token name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Token name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Token name() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Xiost numbral discless unit256 high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large area	× × × × × × × ×
Cas requirement of function NFToernMetadata symbot) topic infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function NFToernMetadata storative/consideres, address until 250 high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Xion Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Xion Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modif	x x x x x x x x x x x x x x x x x x x
In programment of function NPTownMetadada symbol) high infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modely large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function NPTownMetadada Insential Please avoid loops in your functions or actions that modely large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function NPTownMetadada Insential Please avoid loops in your functions or actions that modely large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Total naturally large, infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modely large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Total naturally large, infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modely large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Total naturally large areas and storage (this includes clearing or copying arrays in storage). Gas requirement of function Total naturally large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Total naturally large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Total naturally large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Total naturally large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Total naturally large areas of storage (this includes clearing or copying arrays in storage). Gas requirement of function Tot	x x x x x x x x x x x x x x x x x x x
Cas requirement of function NFToernMetadata symbot) topic infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function NFToernMetadata storative/consideres, address until 250 high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Xion Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage this includes clearing or copying arrays in storage). Gas requirement of function Xion Tolern Arraysha() high; infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modif	× × × × × × × × × × × × ×



as requirement of function Zxc transfe	
	r(address, uint256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
as requirement of function Zxc.transfe	From(address, address, uint256) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
illback function of contract ZxcCrowd	alle requires too much gas (infinite). If the failback function requires more than 2300 gas, the contract cannot receive Ether
s requirement of function ZxcCrowds	ale buyTokens() high: Infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
ddressUtils.isContract(address) : Is co ore	nstant but potentially should not be <-br/>
FTokenEnumerablemint(address,uir	1256): Potentially should be constant but is not. https://www.note-win-win-win-win-win-win-win-win-win-win
TokenMetadataburn(address,uint2	56): Potentially should be constant but is not. <pre><pre><pre></pre> <pre>Modifiers are currently not considered by this static analysis.</pre></pre></pre>
cc.transfer(address,uint256) : Potentia	lly should be constant but is not. Note- <in analysis.<="" are="" by="" considered="" currently="" modifiers="" not="" p="" static="" this=""></in>
c.transferFrom(address,address,uint	156): Potentially should be constant but is not. https://www.not.englines.org/bright-analysis.
rfeMath.mul(uint256,uint256) : Variab	es have very similar names _a and _b. Note: Modifiers are currently not considered by this static analysis.
rfeMath.div(uint256,uint256) : Variabl	s have very similar names _a and _b. Note: Modifiers are currently not considered by this static analysis.
afeMath.sub(uint256,uint256) : Variab	es have very similar names _a and _b. Note: Modifiers are currently not considered by this static analysis.
rfeMath.add(uint256,uint256): Variab	es have very similar names _a and _b. Note: Modifiers are currently not considered by this static analysis.
ken.balanceOf(address) : Variables h	ave very similar names balances and _balance. Note: Modifiers are currently not considered by this static analysis.
cert.mint(address,uint256,string,string	byles32[] byles32[]): Variables have very similar names _to and _id. Note: Modifiers are currently not considered by this static analysis.
cCrowdsale.(address,address,addre	s, uint256,
	s unit256 unit
c. 29 Remix	IDE static analysis report part 4:
ccrowdsale buyTokens(): Variables c. 29 Remix	IDE static analysis report part 4: ave very similar names token and tokens. Note: Modifiers are currently not considered by this static analysis.
cc. 29 Remix ccrowdsale buyTokens(): Variables t	IDE static analysis report part 4: ave very similar names token and tokens. Note: Modifiers are currently not considered by this static analysis.
cc. 29 Remix c. 0. 29 Remix c. 29 Condisale buy Tokens () : Variables 1 ccrowdsale buy Tokens () : Variables 1 ccrowdsale buy Tokens () : Variables 1	IDE static analysis report part 4: ave very similar names token and tokens. Note: Modifiers are currently not considered by this static analysis. ave very similar names token and tokens. Note: Modifiers are currently not considered by this static analysis. ave very similar names token and tokens. Note: Modifiers are currently not considered by this static analysis. ave very similar names tokens and tokens. Note: Modifiers are currently not considered by this static analysis.
cc. 29 Remix cc. voidsale buyTokens(): Variables to the condition of the	IDE static analysis report part 4: ave very similar names token and tokens. Note: Modifiers are currently not considered by this static analysis.

