WorldesIO Smart Contract
Audit Report

Thu Sep 05 2024

contact@bitslab.xyz https://twitter.com/scalebit_

ScaleBit

WorldesIO Smart Contract Audit Report

1 Executive Summary

1.1 Project Information

Description	Achieves the tokenization of real estate assets.
Type	Token
Auditors	ScaleBit
Timeline	Mon May 20 2024 - Wed Jun 12 2024
Languages	Solidity
Platform	ETH ARB POL BSC
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/WorldesIO/smart-contract
Commits	ca7d926e9387f304ccea4ab72860e882242109af 917497ff5a64e2e3fc8b527d0d28d6f8ef87428c 221e2022d3c9a592b7b13b479805f35381258370 d4199a7d833053f78de6ae87d8aa416d9393ff58 c23ec6b0314777fc9b386557951ef3103616353f db9259425d1c127b925dc86a182a6f75ca99b75d

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash
WPR	contracts/asset/WorldesPropertyRights.sol	e24275aa73ab9b43edf522e39f0146e22b3b233a
WRAT	contracts/asset/WRAToken.sol	b1b5f650afe2346c9d810b4c0d9e5d5df9a5f4dd
WRWAT	contracts/asset/WorldesRWAToken.sol	69898c22d80808cf1ca4de2d8d6a9c84fd5758fb
WRWATF	contracts/asset/WorldesRWATokenFactory.sol	47753853421f546a9aecc6b7544fc25be552a5e0

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	6	5	1
Informational	2	2	0
Minor	3	3	0
Medium	0	0	0
Major	1	0	1
Critical	0	0	0

1.4 ScaleBit Audit Breakdown

ScaleBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow
- Number of rounding errors
- Unchecked External Call
- Unchecked CALL Return Values
- Functionality Checks
- Reentrancy
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic issues
- Gas usage
- Fallback function usage
- tx.origin authentication
- Replay attacks
- Coding style issues

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** and **"Formal Verification"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by WorldEs to identify any potential issues and vulnerabilities in the source code of the WorldesIO smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified 6 issues of varying severity. After communicating with the WorldEs and reviewing the updated code, the current status is as follows. More information can be found in the detailed findings in Section 4.

ID	Title	Severity	Status
WPR-1	Inconsistency between Addition and Deletion	Minor	Fixed
WPR-2	Lack of Events Emit	Minor	Fixed
WPR-3	Gas Optimization	Minor	Fixed
WPR-4	`_RWA_ADDRESS_TO_TOKEN_ID_` Status Not Updated	Informational	Fixed
WPR-5	Incorrect Spelling	Informational	Fixed
WRA-1	Centralization Risk	Major	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the WorldesIO Smart Contract :

- The `WorldesPropertyRights` contract is deployed at `0xEb8172A596fDA7b8c9992dc97ba21db8dFC9ce2d` .
- The `WorldesRWATokenFactory` contract is deployed at `0x4Ef31B45919aE1874840B9563D46FCD57E2Ae0b7` .
- The `WorldesRWAToken` contract is deployed at `0xcEab5Af10D5376016c8C352ea77F8Bc6a88bDa11` .

Owner

- The owner can add and remove a minter for the `WPR` token through `addMinter()` \ `removeMinter()` .
- The owner can set the `_WHITE_LISTED_ENABLE_` status of the `WorldesRWAToken` contract through `enableWhiteList()` \ `disableWhiteList()` .
- The owner can set the `_LIST_ADMIN_` through `setListAdmin()` .
- The owner can pause and unpause the `WorldesRWAToken` contract through `pause()` \ `unpause()` .
- The owner can mint any amount of `WorldesRWAToken` to any address through `mint()` .

Minter

- The minter can set the `URI` of the `WPR NFT` through `setTokenURI()` .
- The minter can mint `WPR NFT` through `safeMint()` .

Notray

- The notray can set the status of the `WPR` asset through `setRwaStatus()` .

ListAdmin

- The list admin can add\remove the whitelist through `addWhiteList()` \ `removeWhiteList()` .
- The list admin can add\remove the blacklist through `addBlackList()` \ `removeBlackList()` .

4 Findings

WPR-1 Inconsistency between Addition and Deletion

Severity: Minor

Status: Fixed

Code Location:

contracts/asset/WorldesPropertyRights.sol#102

Descriptions:

When deleting the token ID, the corresponding information stored in the contract should be deleted to improve code security and maintainability:

```
function _burn(uint256 tokenId)
    internal
    override(ERC721, ERC721URIStorage)
{
    super._burn(tokenId);
}
```

Suggestion:

It is recommended to delete relative infomation in `_TOKEN_ID_TO_RWA_ADDRESS_` `_RWA_ADDRESS_TO_TOKEN_ID_` `_ASSET_STATUS_BY_TOKEN_ID_` .

Resolution:

This issue has been fixed. The client has adopted our suggestions.

WPR-2 Lack of Events Emit

Severity: Minor

Status: Fixed

Code Location:

contracts/asset/WorldesPropertyRights.sol#62,66,70,87;

contracts/asset/WorldesRWAToken.sol#45,49,53,57,61,65,69,74,78

Descriptions:

The contract lacks appropriate events for monitoring sensitive operations, which could make it difficult to track sensitive actions or detect potential issues.

Suggestion:

It is recommended to emit events for those important functions.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

WPR-3 Gas Optimization

Severity: Minor

Status: Fixed

Code Location:

contracts/asset/WorldesPropertyRights.sol#134

Descriptions:

In the `_beforeTokenTransfer` function, assertions can be placed before the function call to reduce gas consumption.

```
require(_ASSET_STATUS_BY_TOKEN_ID_[firstTokenId] == AssetStatus.Tradable, "WPR: this token is untradable.");
```

Suggestion:

It is recommended to place the assertion before the function call.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

WPR-4 `_RWA_ADDRESS_TO_TOKEN_ID_` Status Not Updated

Severity: Informational

Status: Fixed

Code Location:

contracts/asset/WorldesPropertyRights.sol#90

Descriptions:

After calling the `clearRwaRelation` function, the `_RWA_ADDRESS_TO_TOKEN_ID_[address(rwaToken)]` status should be updated to 0.

Suggestion:

It is recommended to update the `_RWA_ADDRESS_TO_TOKEN_ID_[address(rwaToken)]` status.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

WPR-5 Incorrect Spelling

Severity: Informational

Status: Fixed

Code Location:

contracts/asset/WorldesPropertyRights.sol#42

Descriptions:

There is a spelling error in the WorldesPropertyRights contract which makes the code confusing, `Notray` should be `Notary`.

Suggestion:

It is recommended to use the correct spelling.

Resolution:

This issue has been fixed. The client has adopted our suggestions.

WRA-1 Centralization Risk

Severity: Major

Status: Acknowledged

Code Location:

contracts/asset/WRAToken.sol;

contracts/asset/WorldesRWAToken.sol

Descriptions:

Centralization risk was identified in the smart contract.

- The Owner of the contract can get one hundred million `WRA` tokens and mint any amount of `WESToken` tokens to any address.

Suggestion:

It is recommended that measures be taken to reduce the centralization issue like using multi-sig.

Resolution:

The client replied: The minting authority is controlled by a multi-signature wallet, with participants including representatives from the exchange, the project team, lawyers, accountants, and market makers. The multi-signature participants act as notaries and authorize whether RWA can be issued and the issuance quantity based on the latest land deeds, audit reports, and property valuation legal documents. The RWA project team is required to announce the asset issuance in advance and use a time-lock mechanism in the smart contract.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.