MetroGalaxy

Security Analysis

Moonpool|MetroGalaxy

1 Overview Abstract

In this report, we consider the security of the MetroGalaxy project. Our main task is to find and describe security issues in the smart contracts of the platform to the team.

1.1 Limitations and use of the report

Broadly speaking, the assessments can not uncover all vulnerabilities of the given smart contracts, thus, it is not guaranteed that the system is secured even if no vulnerabilities are found. The focus of the assessments was limited to given smart contracts, other contracts were excluded (including external libraries or third party codes).

1.2 Summary

We have found **2** high severity issues, **2** medium severity issues and **14** low severity issues.
All of the issues have been adjusted and not presented in the latest source codes.

1.3 Recommendations

We recommend the team to fix all issues, as well as test coverage to ensure the security of the contracts.

2 Assessment Overview

Scope of the audit

In-scope contracts for audit:

- Following file in ***contracts*** folder

```
contracts
├── MetroGalaxyMarketplace.sol
├── MetroToken.sol
├── MetronionAccessories.sol
├── MetronionNFT.sol
├── MetronionSale.sol
├── interface
│   ├── IMetroGalaxyMarketplace.sol
│   ├── IMetronionAccessories.sol
│   ├── IMetronionNFT.sol
│   ├── IMetronionSale.sol
│   └── IWhitelist.sol
├── lib
│   └── ERC1155.sol
└── utils
        ├── AcceptedAssets.sol
        ├── AcceptedToken.sol
        ├── PermissionGroup.sol
        ├── TokenWithdrawable.sol
        └── Whitelist.sol
```

Other source codes are out of scope for this report.

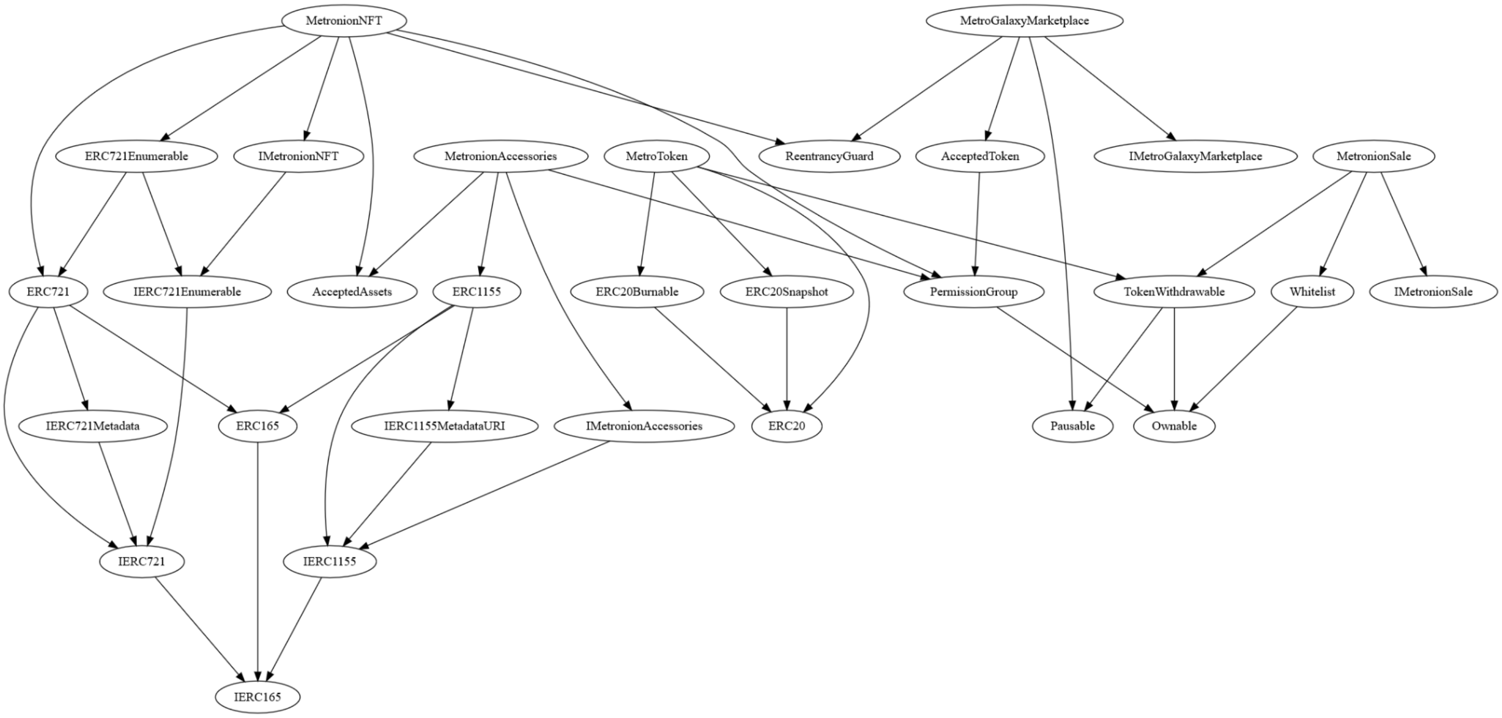
The table below indicates the code versions relevant to this report and when they were received.

	Date	**Commit Hash**	**Note**
1	Dec 14th 2021	1f641995f90adda6c6d6fce711d32e4ca20b5eb4	Initial Version
2	Dec 31th 2021	7c2dab40c8815df9601976e355fcc75058e31f35	Version with fixes

3 System Overview

MetroGalaxy is a metaverse project that uniquely blends a social platform together with an online virtual game that lets its players role-play as anyone they want, do anything they want in an ever-expanding decentralized world.

Metronions are randomly generated with different traits and will be revealed after the NFT sale period ends. All Metronions come with basic accessories that can be changed by minting accessories (ERC1155). These accessories are reflected directly in the Metronion's design in the NFT and in-game.



4 Findings

We have found **2** high severity issues, **2** medium severity issues and **14** low severity issues.

[Fixed] [High] 4.1 MetroGalaxyMarketplace: The owner can block users from delisting assets or canceling offers

In the **delist** and **cancelOffer** functions of the **MetroGalaxyMarketplace** contract, it is required that the **assetAddr** must be accepted. With the ability to update the **acceptedAssets** list, the owner can block users from delisting assets or canceling offers by removing some assets from the accepted list.

Example:

- User 1 lists an asset X to sell, which is an accepted asset.
- The owner removes X from the accepted list.
- User 1 can not delist the asset X, and no one will be able to buy the asset X.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][High] 4.2 MetroGalaxyMarketplace: The owner can withdraw all tokens from offers.

In the **MetroGalaxyMarketplace** contract, anyone can place an offer to buy an accepted asset. At the time of placing the offer, the **acceptedToken** will be transferred from the user's wallet to the marketplace contract. With the ability of changing the **acceptedToken**, the owner can withdraw all tokens from the offers.

Another issue happens when changing the **acceptedToken**:

- All old offers are associated with the old **acceptedToken**, thus, when users cancel their offers, they will receive a new **acceptedToken** instead of the old one.

Example for the owner to withdraw all tokens from offers:

- User offers 100 tokens A to buy an asset X, at that moment, A is the **acceptedToken**.
- The owner deploys an arbitrary token B and changes the **acceptedToken** to token B. The owner makes another offer with the same token amount, i.e 100 tokens B to buy any asset.
- The owner changes the **acceptedToken** back to token A.
- The owner cancels the offer. As now the **acceptedToken** is token A, the contract transfers 100 token A back to the owner's wallet.
- Thus, the owner has successfully withdrawn all tokens A.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Medium] 4.3 MetronionNFT: Broken ERC165 Support

In the **MetronionNFT** contract, the ERC165 interface must be implemented which defines a method to detect what interfaces a smart contract implements. The **MetronionNFT** only overrides this function and returns the logic from the super contracts without any additional logic, thus, querying all new functions will return **false**.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Medium] 4.4 ERC1155: Use contract from OpenZeppelin

Should use the ERC1155 contract from **OpenZeppelin** instead of copying the source code to change the visibility of the **_balances**.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.5 AcceptedAssets: Pure abstract contract should be defined as an interface

The **AcceptedAssets** is a pure abstract contract without any implementations. Can consider defining it as an Interface instead.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.6 MetroGalaxyMarketplace: Redundant SafeMath, redundant check for allowance and balance

From solidity >= 0.8.0, all math operations will be checked for underflow and overflow by default, thus, there is no need to use **SafeMath**.

The balance and allowance will be checked when calling **safeTransferFrom**, thus, it is redundant to check for balance and allowance in the **buy** and **offer** functions.

Comment: This issue has been adjusted and not presented in the latest code.

Function _requireAssetAllowance does not use anymore after the fix, recommend remove it.

[Fixed][Low] 4.6 MetroGalaxyMarketplace: Local variables to save gas

In the **takeOffer** and **buy** functions, the **asset.amount** is accessed multiple times, thus, it should be defined as a local variable to save gas.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.8 MetroGalaxyMarketplace: Use ERC1155Holder and ERC721Holder from OpenZeppelin

Should use the ERC1155Holder & ERC721Holder from OpenZeppelin instead of implementation **onERC721Received** and **onERC1155Received**.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.9 MetronionAccessories: Redundant SafeMath

From solidity >= 0.8.0, all math operations will be checked for underflow and overflow by default, thus, there is no need to use **SafeMath**.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.10 MetronionAccessories: Redundant local variable for getAllAccessoryType function

The **getAllAccessoryType** function is fetching all accessory type names from the **accessoriesTypeName** which is also an array, thus, can return **accessoriesTypeName** directly instead of using a local array.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.11 MetronionAccessories: Use local variables to save gas

- **accessory.minted** is accessed twice in the **mint** function.
- **accessory.burnt** is accessed twice in the **burn** function.
- **_balances[accessoryId][account]** is accessed twice in the **storeAccessories** function.

- **_equipmentStorage[accessoryId][account]** is accessed twice in the **returnAccessories** function.
- **_equipmentStorage[accessoryId][from]** is accessed twice in the **transferEquippedAccessories** function.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.12 MetronionNFT: Inconsistent naming convention

Most private variables have the _ prefix while the **versions** variable doesn't have.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.13 MetronionNFT: Use local variables to save gas

- **version.currentSupply** is accessed multiple times in the **mintMetronion** function, thus, should define a local variable for it.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.14 TokenWithdrawable: Use SafeERC20 and support native token withdrawal

- No support for native tokens withdrawal.
- Should use **safeTransfer** for token transfer.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.15 TokenWithdrawable: Redundant whitelist and pause logics

- Only the owner can withdraw tokens.

- Only the owner can change the whitelist and pause status.

Thus it is quite redundant to have whitelisted tokens and pause mechanisms in the contract.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.16 PermissionGroup: No function to get the list of operators

Can only check if an address is an operator, there is no function to fetch the list of current operators from mapping.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.17 Whitelist: EnumerableSet has _values() to fetch all values in the set

Should use the _values() function to fetch all values in the set.

Comment: This issue has been adjusted and not presented in the latest code.

[Fixed][Low] 4.18 MetronionSale: Use sendValue from Address lib to send the native token

The Address lib provides a sendValue function to send the native token, thus, should use this function instead of re-implementing.

Comment: This issue has been adjusted and not presented in the latest code.