

Smart Contract Security Audit V1

Metaverset Open World NFT

17/12/2021



<https://saferico.com/>

business@saferico.com

https://t.me/SFI_ANN

—

Table of Contents

Table of Contents

Background

Project Information

NFT Information

Executive Summary

File and Function Level Report

File in Scope:

Issues Checking Status

Severity Definitions

Audit Findings

Automatic testing

Testing proves

Inheritance graph

Call graph

Unified Modeling Language (UML)

Functions signature

Automatic general report

Conclusion

Disclaimer

Background

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.

Project Information

- **Website:**
- **Telegram group:**
- **Twitter:**
- **GitHub:**
- **WhitePaper:**
- **Platform:**
- **Contract Address:**

NFT Information

- Name: MOW
- Total Supply:
- Holders:
- Total transactions:

Contracts address deployed to test net (ETH)

MOW NFT contract on testnet

0xaFf5b572793B762B2CAF4358C361756F9D6A9dEf

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well Secured**.

Well Secured	✓
Secured	
Poor Secured	
Insecure	

Automated checks are with remix IDE. All issues were performed by the team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the Project Information section and all issues found are located in the audit overview section.

Team found 0 critical, 0 high, 0 medium, 0 low, 0 very low-level issues and 3 notes in all solidity files of the contract

The files:

MetaverserNFT.sol

IMarketplaceAssets.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
MetaverserNFT.sol	9d51f95c0ab06466b1c3afb9ba691a85f299acce21b57f915eb3bf4ecedc8b04	0xaFf5b572793B762B2CAF4358C361756F9D6A9dEf

- Contract: MetaverserNFT
- Inherit: IMarketplaceAssets,ERC721URIStorage,ERC721Enumerable,Ownable,ERC721Holder
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

Function	Test Result	Type / Return Type	Score
name	✓	Read / public	Passed
symbol	✓	Read / public	Passed
tokenURI	✓	Read / public	Passed
getGameAssetsByTokenId	✓	Read / public	Passed
balanceOf	✓	Read / public	Passed
Owner	✓	Read / public	Passed
TokenIdCounter	✓	Read / public	Passed
totalSupply	✓	Read / public	Passed
getApproved	✓	Read / public	Passed
tokenByIndex	✓	Read / public	Passed
tokenOfOwnerByIndex	✓	Read / public	Passed

supportsInterface	✓	Read / public	Passed
isApprovedForAll	✓	Read / public	Passed
ownerOf	✓	Read / public	Passed
GameAssets	✓	Read / public	Passed
createToken	✓	Write / public	Passed
onERC721Received	✓	Write / public	Passed
setaccessListAddress	✓	Write / public	Passed
approveSpendingToken	✓	Write / public	Passed
approve	✓	Write / public	Passed
safeTransferFrom	✓	Write / public	Passed
safeTransferFrom	✓	Write / public	Passed
setApprovalForAll	✓	Write / public	Passed
setTokenAssetType	✓	Write / public	Passed
setTokenName	✓	Write / public	Passed
setTokenURI	✓	Write / public	Passed
TransferFrom	✓	Write / public	Passed
renounceOwnership	✓	Write / public	Passed

Issues Checking Status

No.	Issue Description	Checking Status
1	Compiler warnings.	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 with note
10	Methods execution permissions.	Passed
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	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.	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss 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
Note	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

Audit Findings

Critical:

No critical severity vulnerabilities were found.

High:

No High severity vulnerabilities were found

Medium:

No Medium severity vulnerabilities were found.

Low:

No Low severity vulnerabilities were found.

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

#Note1

#Gas cost:

In detail

Gas requirement of function ERC721Holder.onERC721Received is 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).

```
function onERC721Received(
    address,
    address,
    uint256,
    bytes memory
) public virtual override returns (bytes4) {
    return this.onERC721Received.selector;
}
```

This on local calls:

Use of "this" for local functions: Never use "this" to call functions in the same contract, it only consumes more gas than normal local calls.

```
function onERC721Received(
    address,
    address,
    uint256,
    bytes memory
) public virtual override returns (bytes4) {
    return this.onERC721Received.selector;
}
```

#Delete dynamic array:

The "delete" operation when applied to a dynamically sized array in Solidity generates code to delete each of the elements contained. If the array is large, this operation can surpass the block gas limit and raise an OOG exception. Also nested dynamically sized objects can produce the same results.

```
function _burn(uint256 tokenId) internal virtual override {
    super._burn(tokenId);

    if (bytes(_tokenURIs[tokenId]).length != 0) {
        delete _tokenURIs[tokenId];
    }
}
```

Automatic Testing

1- Check for security

9d51f95c0ab06466b1c3afb9ba691a85f299acce21b57f915eb3bf4ecedc8b04

File: Metaverse... | Language: solidity | Size: 5074 bytes | Date: 2021-12-17T11:32:36.788Z

Critical	High	Medium	Low	Note
0	0	0	0	3

✓

2- SOLIDITY STATIC ANALYSIS

SOLIDITY STATIC ANALYSIS

☒ Select all ☒ Autorun Run

Security

☒ Select Security

- ☒ **Transaction origin:**
'tx.origin' used
- ☒ **Check-effects-interaction:**
Potential reentrancy bugs
- ☒ **Inline assembly:**
Inline assembly used
- ☒ **Block timestamp:**
Can be influenced by miners
- ☒ **Low level calls:**
Should only be used by experienced devs
- ☒ **Block hash:**
Can be influenced by miners
- ☒ **Selfdestruct:**
Contracts using destructed contract can be broken

Gas & Economy

☒ Select Gas & Economy

- ☒ **Gas costs:**
Too high gas requirement of functions
- ☒ **This on local calls:**
Invocation of local functions via 'this'
- ☒ **Delete dynamic array:**
Use require/assert to ensure complete deletion
- ☒ **For loop over dynamic array:**
Iterations depend on dynamic array's size
- ☒ **Ether transfer in loop:**
Transferring Ether in a for/while/do-while loop

SOLIDITY STATIC ANALYSIS

ERC

☒ Select ERC

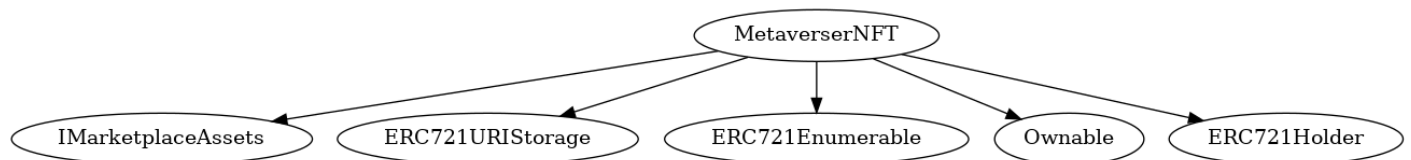
- ☒ **ERC20:**
'decimals' should be 'uint8'

Miscellaneous

☒ Select Miscellaneous

- ☒ **Constant/View/Pure functions:**
Potentially constant/view/pure functions
- ☒ **Similar variable names:**
Variable names are too similar
- ☒ **No return:**
Function with 'returns' not returning
- ☒ **Guard conditions:**
Ensure appropriate use of require/assert
- ☒ **Result not used:**
The result of an operation not used
- ☒ **String length:**
Bytes length != String length
- ☒ **Delete from dynamic array:**
'delete' leaves a gap in array
- ☒ **Data truncated:**
Division on int/uint values truncates the result

3- Inheritance graph



4- SOLIDITY UNIT TESTING

```

// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.4.22 <0.9.0;

// This import is automatically injected by Remix
import "remix_tests.sol";

// This import is required to use custom transaction context
// Although it may fail compilation in 'Solidity Compiler' plugin
// But it will work fine in 'Solidity Unit Testing' plugin
import "remix_accounts.sol";
import "../MetaverserNFT.sol";

// File name has to end with '_test.sol', this file can contain more than one
testSuite contracts
contract testSuite {

    /// 'beforeAll' runs before all other tests
    /// More special functions are: 'beforeEach', 'beforeAll', 'afterEach' &
    'afterAll'
    function beforeAll() public {
        // <instantiate contract>
        Assert.equal(uint(1), uint(1), "1 should be equal to 1");
    }

    function checkSuccess() public {
        // Use 'Assert' methods: https://remix-ide.readthedocs.io/en/latest/assert\_library.html
        Assert.ok(2 == 2, 'should be true');
        Assert.greaterThan(uint(2), uint(1), "2 should be greater than to 1");
        Assert.lessThan(uint(2), uint(3), "2 should be lesser than to 3");
    }

    function checkSuccess2() public pure returns (bool) {
        // Use the return value (true or false) to test the contract
        return true;
    }

    function checkFailure() public {
        Assert.notEqual(uint(1), uint(2), "1 should not be equal to 1");
    }

    /// Custom Transaction Context: https://remix-ide.readthedocs.io/en/latest/unittesting.html#customization
    /// #sender: account-1
    /// #value: 100
    function checkSenderAndValue() public payable {
        // account index varies 0-9, value is in wei
        Assert.equal(msg.sender, TestsAccounts.getAccount(1), "Invalid sender");
        Assert.equal(msg.value, 100, "Invalid value");
    }
}

```

SOLIDITY UNIT TESTING



Test your smart contract in Solidity.

Select directory to load and generate test files.

Test directory:

tests

Create

Generate

How to use...

Run

Stop

☒ Select all

☒ tests/MetaverserNFT_test.sol

Progress: 1 finished (of 1)

PASS testSuite

(tests/MetaverserNFT_test.sol)

✓ Before all



✓ Check success



✓ Check success2



✓ Check failure



✓ Check sender and value



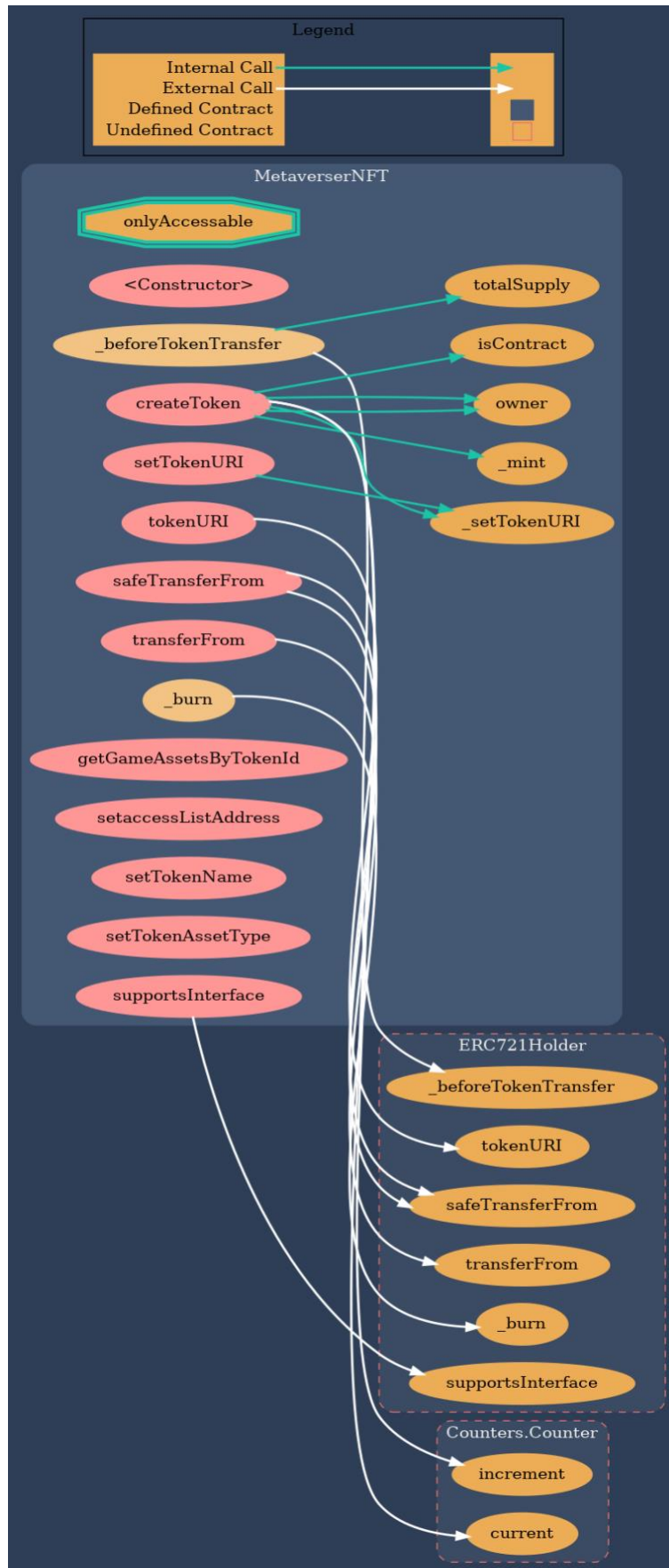
Result for

tests/MetaverserNFT_test.sol

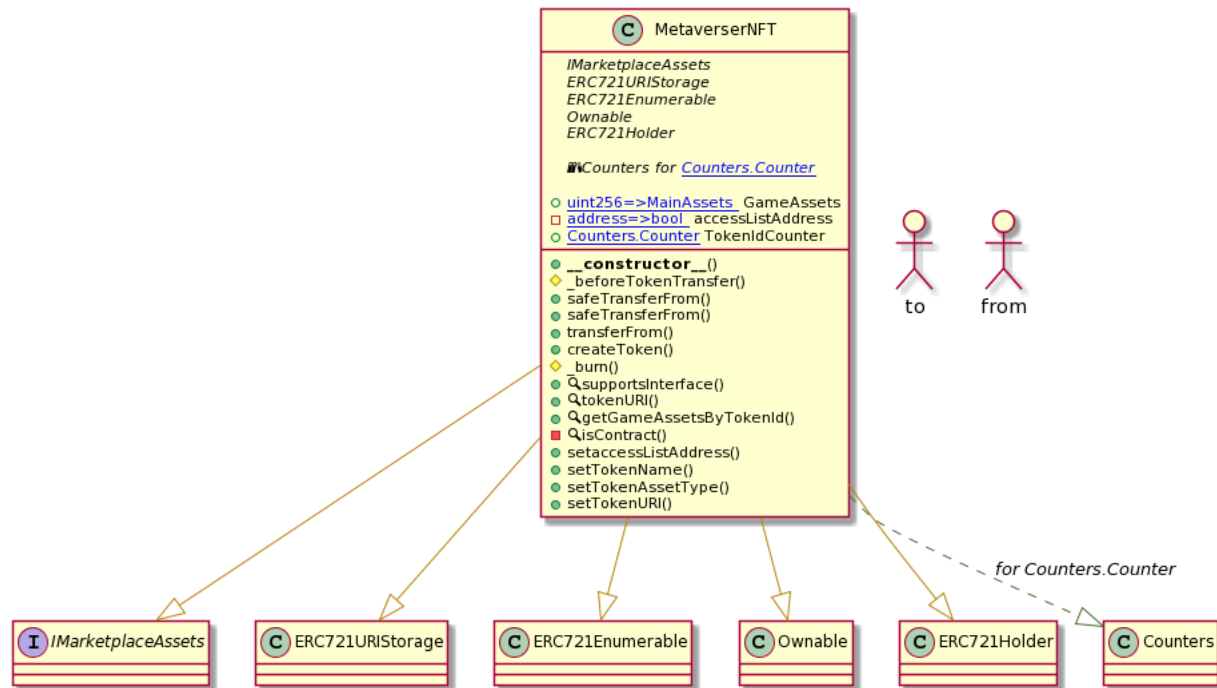
Passing: 5

Total time: 0.39s

5- Call graph



Unified Modeling Language (UML)



Functions signature

```
16279055 => isContract(address)
9693b8f3 => getFullData()
cad3be83 => _beforeTokenTransfer(address,address,uint256)
42842e0e => safeTransferFrom(address,address,uint256)
b88d4fde => safeTransferFrom(address,address,uint256,bytes)
23b872dd => transferFrom(address,address,uint256)
322caed1 => createToken(string,string,uint256,string)
9b1f9e74 => _burn(uint256)
01ffc9a7 => supportsInterface(bytes4)
c87b56dd => tokenURI(uint256)
afd1b116 => getGameAssetsByTokenId(uint256)
e0ac1709 => setaccessListAddress(address,bool)
cdb0e89e => setTokenName(uint256,string)
dacc6e3e => setTokenAssetType(uint256,uint256)
162094c4 => setTokenURI(uint256,string)
```


Automatic general report



Files Description Table

File Name	SHA-1 Hash
/Users/macbook/Desktop/smart contracts/MetaverserNFT.sol	80df2982f170ac08a12306ae62f7c3b22d0b32bd

Contracts Description Table

Contract	Type	Bases	
:-----: :-----: :-----: :-----:			
-----:			
L	**Function Name**	**Visibility**	**Mutability**
Modifiers			
IMarketplaceAssets	Interface		
MetaverserNFT	Implementation	IMarketplaceAssets, ERC721URIStorage, ERC721Enumerable, Ownable, ERC721Holder	
L	<Constructor>	Public !	ERC721
L	_beforeTokenTransfer	Internal	
L	safeTransferFrom	Public !	onlyAccessable
L	safeTransferFrom	Public !	onlyAccessable
L	transferFrom	Public !	onlyAccessable
L	createToken	Public !	NO!
L	_burn	Internal	
L	supportsInterface	Public !	NO!
L	tokenURI	Public !	NO!
L	getGameAssetsByTokenId	Public !	NO!
L	isContract	Private	
L	setaccessListAddress	Public !	onlyOwner
L	setTokenName	Public !	onlyAccessable
L	setTokenAssetType	Public !	onlyAccessable
L	setTokenURI	Public !	onlyAccessable

Legend

Symbol	Meaning
:-----:	
	Function can modify state
	Function is payable

Conclusion

The contracts are written systematically. Team found no critical issues. So, it is good to go for production.

Since possible test cases can be unlimited and developer level documentation (code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan Everything.

Security state of the reviewed contract is “well secured”.

- ✓ No volatile code.
- ✓ Not many high severity issues were found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of 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 the team on the basis of what it says or doesn't say, or how team produced it, and it is important for you to conduct your own independent investigations before making any decisions. team go into more detail on this in the below disclaimer below – please make sure to read it in full.

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 Saferico and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Saferico s) owe no duty of care towards you or any other person, nor does Saferico 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 Saferico 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, Saferico hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Saferico, 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.