Smart Contract Security Audit V1

NFT TIMES Smart Contract

28/2/2022

https://saferico.com/

business@saferico.com
https://t.me/SFI_ANN

Table of Contents

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

- **Platform**: Binance Smart Chain
- **Contract Address**: 0x65B5C5C70b306A4868d423a686Fdb4cfC03F7C2B
- **Code:**

https://testnet.bscscan.com/address/0x064be8c541e417c818DFe4195218B5A512d151F6#code

NFT Information

- Name: NFT_TIMES
- Total Supply:
- Holders:
- Total transactions:

Contracts address deployed to test net (BSC)

NFT TIMES Smart contract on BSC test net to test write functions by the auditor.

https://testnet.bscscan.com/address/0x65b5c5c70b306a4868d423a686fdb4cfc03f7c2b

Executive Summary

According to our assessment, the customer`s solidity smart contract is **Well-Secured**. Because the team fixed the high and low issues.

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, 1 high, 0 medium, 3 low, 0 very low-level issues and 2 note in all solidity files of the contract

The files:

NFTTIMES.sol

File and Function Level Report

File in Scope:

Contract Name	SHA 256 hash	Contract Address
NFTTIMES.sol	**dc3874ab96981038ca26e215ba3c8485ca782118e99ee9497cc862a2d36d70a4**	0x65B5C5C70b306A4868d423a686Fdb4cfC03F7C2B

- Contract: NFTTIMES
- Inherit: ERC721
- 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**
adminAddress	✓	Read / public	**Passed**
supportsInterface	✓	Read / public	**Passed**
ownerOf	✓	Read / public	**Passed**
balanceOf	✓	Read / public	**Passed**
Owner	✓	Read / public	**Passed**
creationFees	✓	Read / public	**Passed**
referenceFees	✓	Read / public	**Passed**
getApprovedForAll	✓	Read / public	**Passed**
sellignFees	✓	Read / public	**Passed**
getApproved	✓	Read / public	**Passed**

setSellignFees	✓	Write / public	**Passed**
approve	✓	Write / public	**Passed**
safeTransferFrom	✓	Write / public	**Passed**
safeTransferFrom	✓	Write / public	**Passed**
setReferenceFees	✓	Write / public	**Passed**
setCreationFees	✓	Write / public	**Passed**
mintNFT	✓	Write / payable	**Passed**
setAdminAddress	✓	Write / public	**Passed**
claimAmount	✓	Write / payable	**Passed**
setApprovalForAll	✓	Write / public	**Passed**
transferFrom	✓	Write / public	**Passed**
buyWithBNB	✓	Write / payable	**Passed**
buyWithTokens	X	Write / public	**Not 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	Design Logic.	**Passed**
6	Timestamp dependence.	**Passed with Notes**
7	Integer Overflow and Underflow.	**Passed**
8	DoS with Revert.	**Passed**
9	DoS with block gas limit.	**Passed**
10	Methods execution permissions.	**Passed**
11	Economy model. 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**

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:

#Possibility of losing all funds

Description

The NFT smart contract doesn't have any withdraw function which allow to the team to withdraw the funds to complete their project without this function all funds will be locked in the smart contract forever, the project will fail without these funds.

Remediation

You have to add withdraw function to make sure all funds go to the team to able to complete the project.

Status: Closed. Fixed in version2.

Medium:

No Medium severity vulnerabilities were found

Low:

#Pragam version not fixed

Description

It is a good practice to lock the solidity version for a live deployment (use 0.8.2 instead of ^0.8.0). contracts should be deployed with the same compiler version and flags that they have been tested the most with. Locking the pragma helps ensure that contracts do not accidentally get deployed using, for example, the latest compiler which may have higher risks of undiscovered bugs. Contracts may also be deployed by others and the pragma indicates the compiler version intended by the original authors.

Remediation

Remove the ^ sign to lock the pragma version.

Status: Closed. Fixed in version2.

# Deploy Constructor should be public not public payable

Description

The developer makes the main constructor of NFT Smart contract public payable and it is normally the main constructor is only public, this can lead to a vulnerability or make unauthorized or unintended state changes.

```
    constructor() payable ERC721('Nft_Times', 'NFT_TIMES') {
        adminAddress = payable(msg.sender);
        sellingFess = 5;
    }
```

Remediation

The team should payable and make only public not public payable.

Status: Closed. Fixed in version2.

# BuyWithTokens function doesn't work correctly

Description

The developer adds buyWithToken as a public write function and it doesn't work correctly, this can lead to a vulnerability or make unauthorized or unintended state changes.

```
    function buyWithTokens(uint256 _from, uint256 _to, uint256 _royalties, uint256 
buyAmount, address paymentCurrency, address payable _artist) public  {
            require(buyAmount > 0, "Amount should be grater then 0 ");
            calclulateAmount(_from, paymentCurrency, buyAmount, _royalties, 
_artist);
            for(uint256 i = _from; i <= _to; i++) {
                address _owner = (ownerOf(i));
                _transfer(_owner, msg.sender, i);
            }
    }
```

Remediation

The team should make it public and payable to be able to pay to buy with it.

Status: Closed. Fixed in version2.

Very Low:

No Very Low severity vulnerabilities were found.

Notes:

#Unnecessary use of SafeMath

Description
Solidity version 0.8 was released with SafeMath checks inbuilt, we can avoid using an explicit safe math library.

Remediation
Remove SafeMath Library to save gas fees.

Status: Closed. Fixed in version2.

#There isn't any transferOwnership function

Description

The developer adds setAdmainAdress to transfer the ownership to a new address and make it as the controller, which it isn't the best way to transfer the ownership.

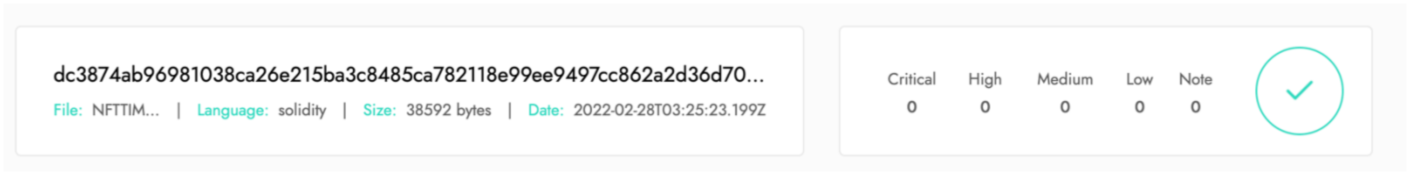
```
function setAdminAddress(address payable _adminAdd) public {
        require(msg.sender == adminAddress, "Only owner can set the address");
        adminAddress = _adminAdd;
    }
```

Remediation
The team have to add ownable library to the contract and make NFTTIMES contract inhert ERC721 and ownable.

Status: Acknowledged.

Automatic Testing

1- Check for security

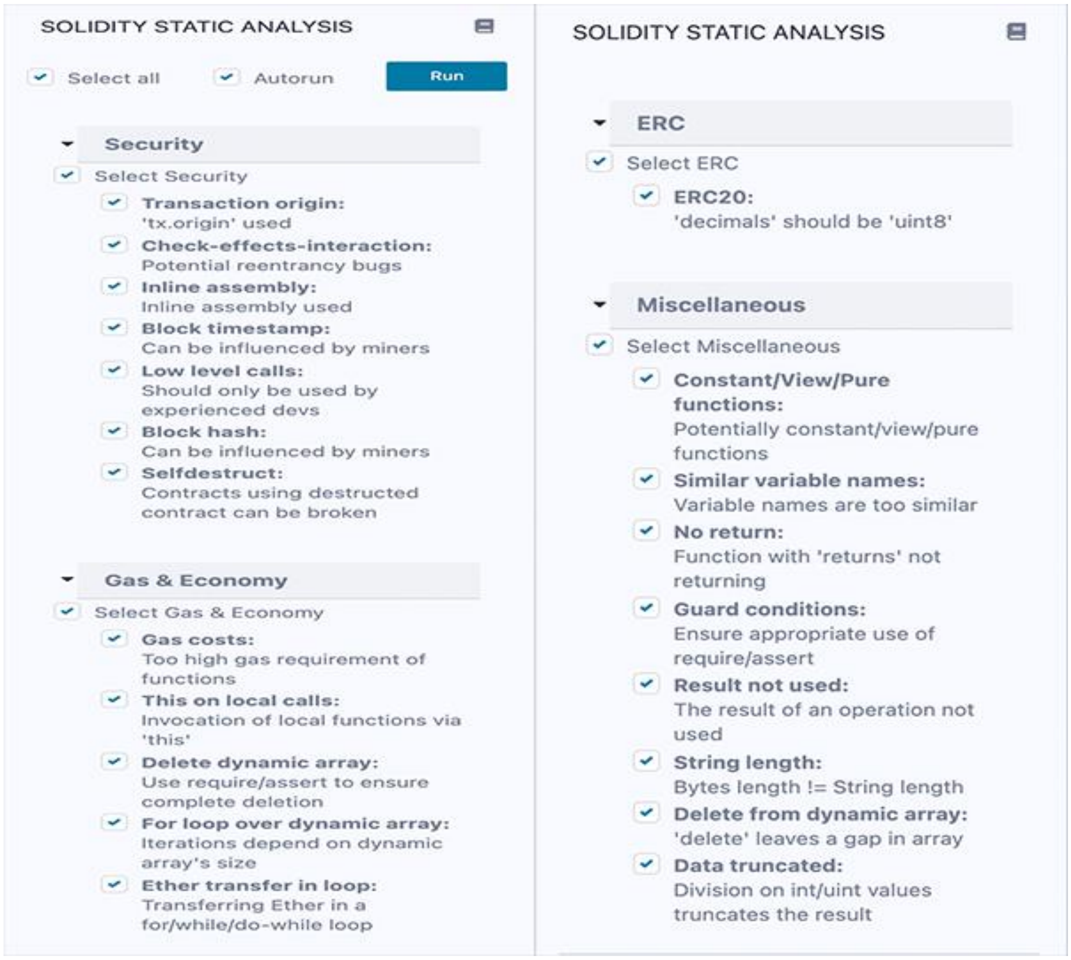


dc3874ab96981038ca26e215ba3c8485ca782118e99ee9497cc862a2d36d70...

File: NFTTIM... | Language: solidity | Size: 38592 bytes | Date: 2022-02-28T03:25:23.199Z

Critical	High	Medium	Low	Note
0	0	0	0	0

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

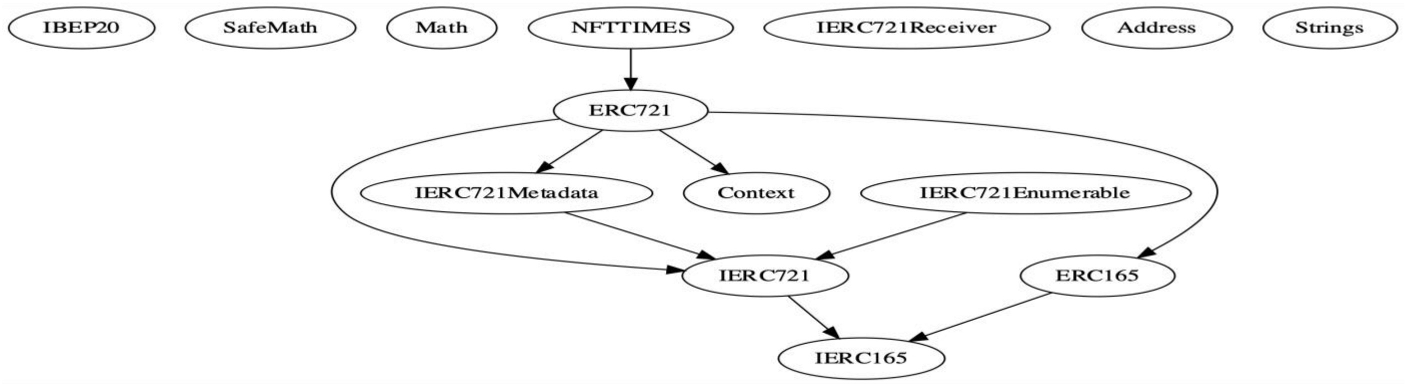
- ✓ **ERC20:** 'decimals' should be 'uint8'

Miscellaneous

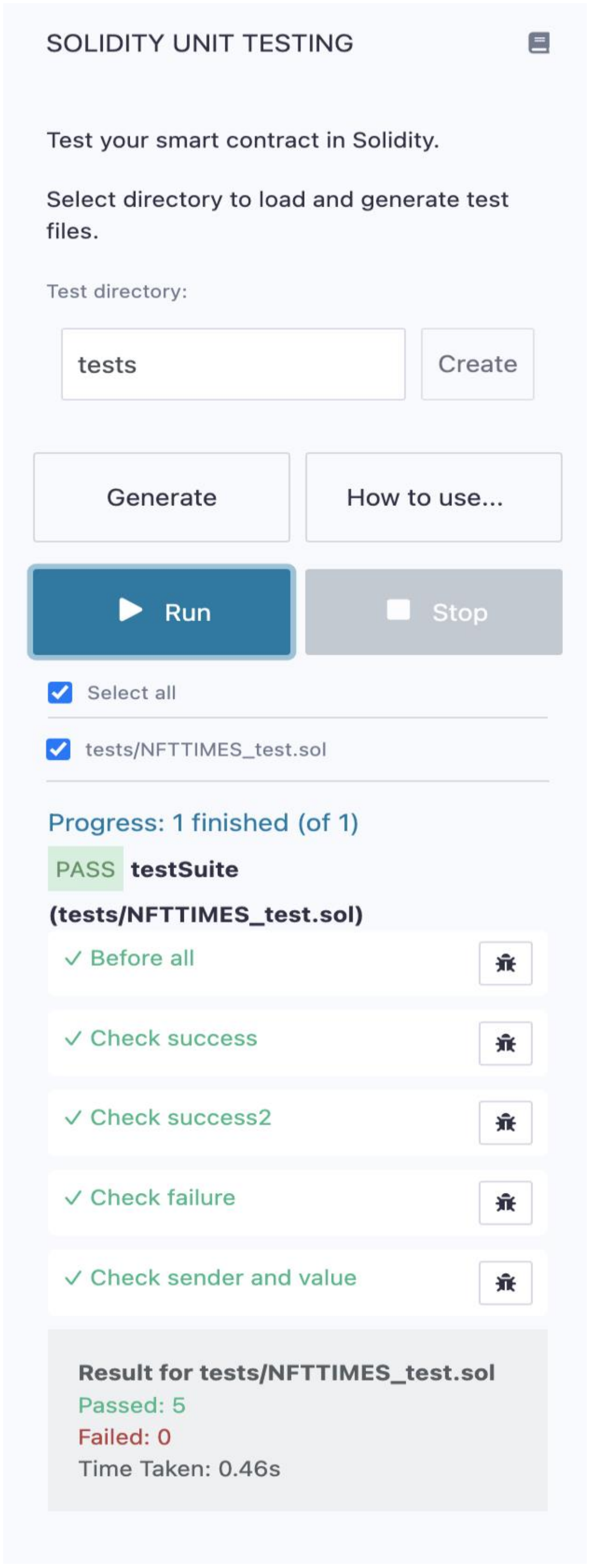
✓ Select Miscellaneous

- ✓ **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


SOLIDITY UNIT TESTING

Test your smart contract in Solidity.

Select directory to load and generate test files.

Test directory: tests

✓ Select all

✓ tests/NFTTIMES_test.sol

Progress: 1 finished (of 1)

PASS **testSuite (tests/NFTTIMES_test.sol)**

✓ Before all

✓ Check success

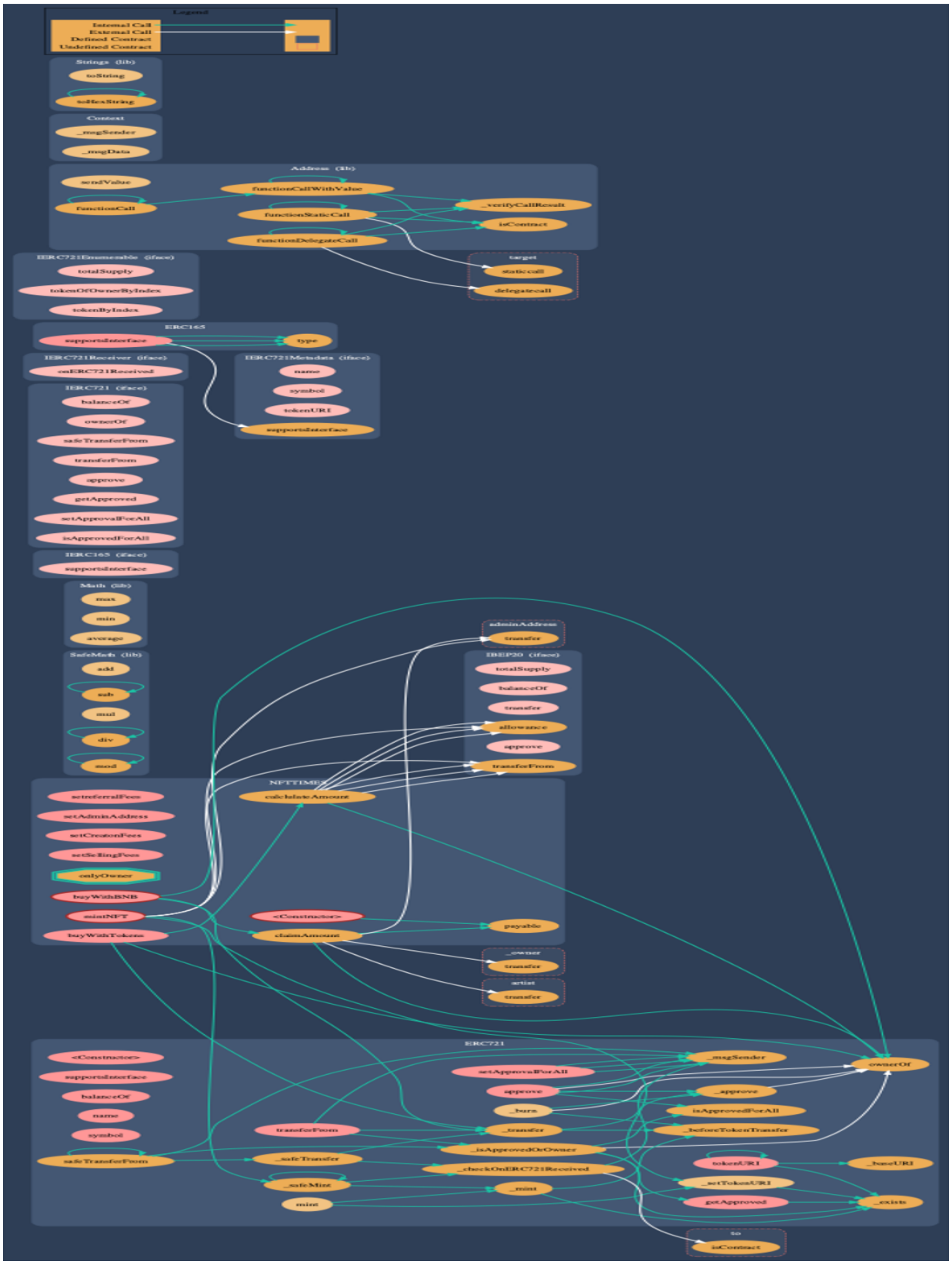
✓ Check success2

✓ Check failure

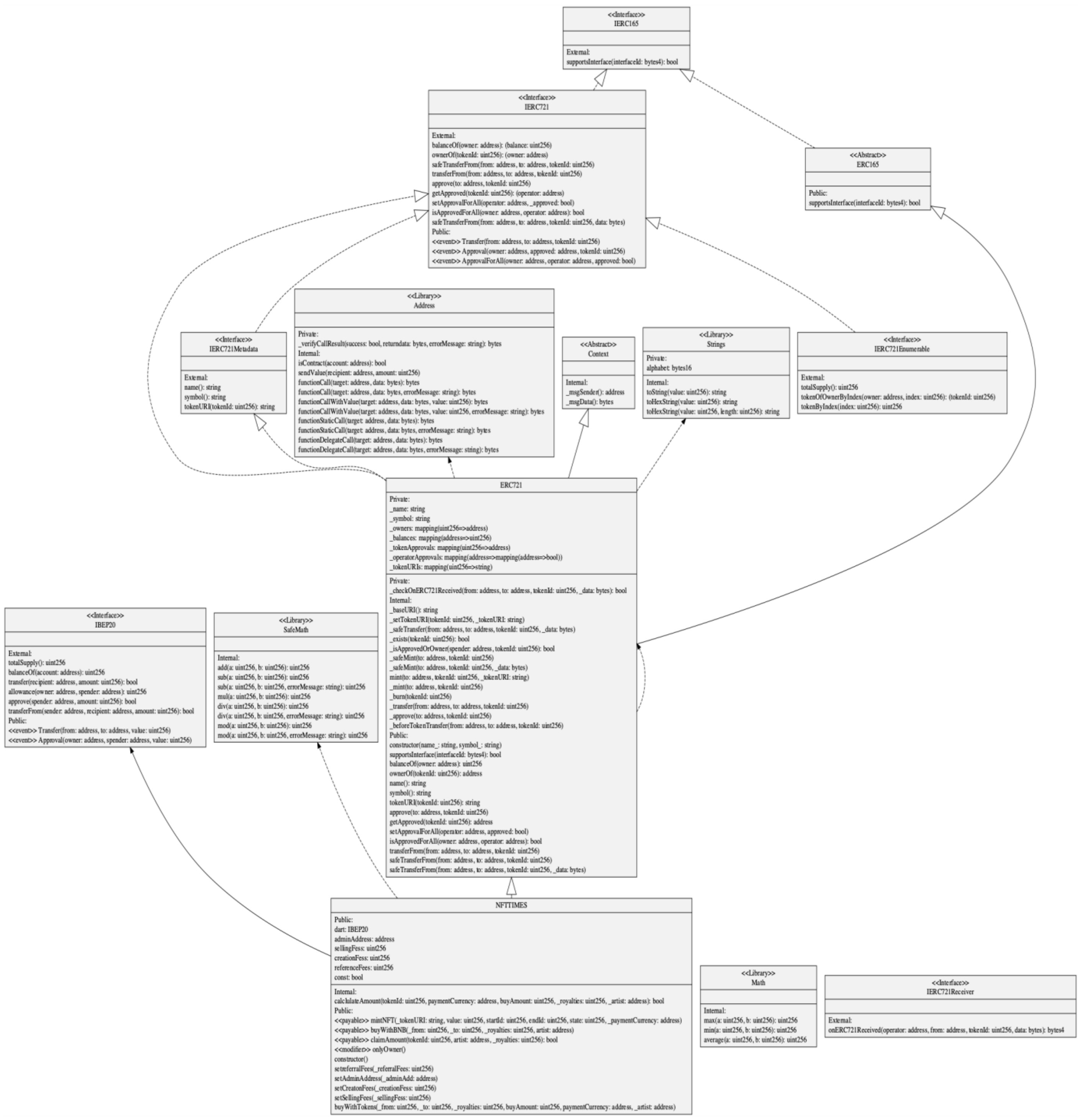
✓ Check sender and value

Result for tests/NFTTIMES_test.sol
Passed: 5
Failed: 0
Time Taken: 0.46s

5- Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Strings (lib)
toString
toHexString
Context
_msgSender
_msgData
Address (lib)
sendValue
functionCall
functionCallWithValue
functionStaticCall
functionDelegateCall
_verifyCallResult
isContract
target
staticcall
delegatecall
IERC721Enumerable (iface)
totalSupply
tokenOfOwnerByIndex
tokenByIndex
ERC165
supportsInterface
type
IERC721Receiver (iface)
onERC721Received
IERC721Metadata (iface)
name
symbol
tokenURI
supportsInterface
IERC721 (iface)
balanceOf
ownerOf
safeTransferFrom
transferFrom
approve
getApproved
setApprovalForAll
isApprovedForAll
IERC165 (iface)
supportsInterface
Math (lib)
max
min
average
SafeMath (lib)
add
sub
mul
div
mod
adminAddress
transfer
IBEP20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
NFTTIMES
setreferralFees
setAdminAddress
setCreationFees
setSellingFees
onlyOwner
buyWithBNB
mintNFT
buyWithTokens
calculateAmount
<Constructor>
claimAmount
payable
_owner
transfer
artist
transfer
ERC721
<Constructor>
supportsInterface
balanceOf
name
symbol
safeTransferFrom
transferFrom
_safeTransfer
_safeMint
mint
setApprovalForAll
approve
_burn
_transfer
_isApprovedOrOwner
_checkOnERC721Received
_mint
_msgSender
ownerOf
_approve
isApprovedForAll
_beforeTokenTransfer
tokenURI
_baseURI
_setTokenURI
getApproved
_exists
to
isContract

Unified Modeling Language (UML)



<<Interface>>
IERC165
External:
supportsInterface(interfaceId: bytes4): bool
<<Interface>>
IERC721
External:
balanceOf(owner: address): (balance: uint256)
ownerOf(tokenId: uint256): (owner: address)
safeTransferFrom(from: address, to: address, tokenId: uint256)
transferFrom(from: address, to: address, tokenId: uint256)
approve(to: address, tokenId: uint256)
getApproved(tokenId: uint256): (operator: address)
setApprovalForAll(operator: address, _approved: bool)
isApprovedForAll(owner: address, operator: address): bool
safeTransferFrom(from: address, to: address, tokenId: uint256, data: bytes)
Public:
<<event>> Transfer(from: address, to: address, tokenId: uint256)
<<event>> Approval(owner: address, approved: address, tokenId: uint256)
<<event>> ApprovalForAll(owner: address, operator: address, approved: bool)
<<Abstract>>
ERC165
Public:
supportsInterface(interfaceId: bytes4): bool
<<Interface>>
IERC721Metadata
External:
name(): string
symbol(): string
tokenURI(tokenId: uint256): string
<<Library>>
Address
Private:
_verifyCallResult(success: bool, returndata: bytes, errorMessage: string): bytes
Internal:
isContract(account: address): bool
sendValue(recipient: address, amount: uint256)
functionCall(target: address, data: bytes): bytes
functionCall(target: address, data: bytes, errorMessage: string): bytes
functionCallWithValue(target: address, data: bytes, value: uint256): bytes
functionCallWithValue(target: address, data: bytes, value: uint256, errorMessage: string): bytes
functionStaticCall(target: address, data: bytes): bytes
functionStaticCall(target: address, data: bytes, errorMessage: string): bytes
functionDelegateCall(target: address, data: bytes): bytes
functionDelegateCall(target: address, data: bytes, errorMessage: string): bytes
<<Abstract>>
Context
Internal:
_msgSender(): address
_msgData(): bytes
<<Library>>
Strings
Private:
alphabet: bytes16
Internal:
toString(value: uint256): string
toHexString(value: uint256): string
toHexString(value: uint256, length: uint256): string
<<Interface>>
IERC721Enumerable
External:
totalSupply(): uint256
tokenOfOwnerByIndex(owner: address, index: uint256): (tokenId: uint256)
tokenByIndex(index: uint256): uint256
ERC721
Private:
_name: string
_symbol: string
_owners: mapping(uint256=>address)
_balances: mapping(address=>uint256)
_tokenApprovals: mapping(uint256=>address)
_operatorApprovals: mapping(address=>mapping(address=>bool))
_tokenURIs: mapping(uint256=>string)
Private:
_checkOnERC721Received(from: address, to: address, tokenId: uint256, _data: bytes): bool
Internal:
_baseURI(): string
_setTokenURI(tokenId: uint256, _tokenURI: string)
_safeTransfer(from: address, to: address, tokenId: uint256, _data: bytes)
_exists(tokenId: uint256): bool
_isApprovedOrOwner(spender: address, tokenId: uint256): bool
_safeMint(to: address, tokenId: uint256)
_safeMint(to: address, tokenId: uint256, _data: bytes)
mint(to: address, tokenId: uint256, _tokenURI: string)
_mint(to: address, tokenId: uint256)
_burn(tokenId: uint256)
_transfer(from: address, to: address, tokenId: uint256)
_approve(to: address, tokenId: uint256)
_beforeTokenTransfer(from: address, to: address, tokenId: uint256)
Public:
constructor(name_: string, symbol_: string)
supportsInterface(interfaceId: bytes4): bool
balanceOf(owner: address): uint256
ownerOf(tokenId: uint256): address
name(): string
symbol(): string
tokenURI(tokenId: uint256): string
approve(to: address, tokenId: uint256)
getApproved(tokenId: uint256): address
setApprovalForAll(operator: address, approved: bool)
isApprovedForAll(owner: address, operator: address): bool
transferFrom(from: address, to: address, tokenId: uint256)
safeTransferFrom(from: address, to: address, tokenId: uint256)
safeTransferFrom(from: address, to: address, tokenId: uint256, _data: bytes)
<<Interface>>
IBEP20
External:
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(recipient: address, amount: uint256): bool
allowance(owner: address, spender: address): uint256
approve(spender: address, amount: uint256): bool
transferFrom(sender: address, recipient: address, amount: uint256): bool
Public:
<<event>> Transfer(from: address, to: address, value: uint256)
<<event>> Approval(owner: address, spender: address, value: uint256)
<<Library>>
SafeMath
Internal:
add(a: uint256, b: uint256): uint256
sub(a: uint256, b: uint256): uint256
sub(a: uint256, b: uint256, errorMessage: string): uint256
mul(a: uint256, b: uint256): uint256
div(a: uint256, b: uint256): uint256
div(a: uint256, b: uint256, errorMessage: string): uint256
mod(a: uint256, b: uint256): uint256
mod(a: uint256, b: uint256, errorMessage: string): uint256
NFTTIMES
Public:
dart: IBEP20
adminAddress: address
sellingFess: uint256
creationFess: uint256
referenceFees: uint256
const: bool
Internal:
calclulateAmount(tokenId: uint256, paymentCurrency: address, buyAmount: uint256, _royalties: uint256, _artist: address): bool
Public:
<<payable>> mintNFT(_tokenURI: string, value: uint256, startId: uint256, endId: uint256, state: uint256, _paymentCurrency: address)
<<payable>> buyWithBNB(_from: uint256, _to: uint256, _royalties: uint256, artist: address)
<<payable>> claimAmount(tokenId: uint256, artist: address, _royalties: uint256): bool
<<modifier>> onlyOwner()
constructor()
setreferralFees(_referralFees: uint256)
setAdminAddress(_adminAdd: address)
setCreatonFees(_creationFess: uint256)
setSellingFees(_sellingFess: uint256)
buyWithTokens(_from: uint256, _to: uint256, _royalties: uint256, buyAmount: uint256, paymentCurrency: address, _artist: address)
<<Library>>
Math
Internal:
max(a: uint256, b: uint256): uint256
min(a: uint256, b: uint256): uint256
average(a: uint256, b: uint256): uint256
<<Interface>>
IERC721Receiver
External:
onERC721Received(operator: address, from: address, tokenId: uint256, data: bytes): bytes4

Functions signature

```
Sighash    |   Function Signature
========================
16279055  =>  isContract(address)
18160ddd  =>  totalSupply()
70a08231  =>  balanceOf(address)
a9059cbb  =>  transfer(address,uint256)
dd62ed3e  =>  allowance(address,address)
095ea7b3  =>  approve(address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
b745d336  =>  div(uint256,uint256,string)
f43f523a  =>  mod(uint256,uint256)
71af23e8  =>  mod(uint256,uint256,string)
6d5433e6  =>  max(uint256,uint256)
7ae2b5c7  =>  min(uint256,uint256)
2b7423ab  =>  average(uint256,uint256)
01ffc9a7  =>  supportsInterface(bytes4)
6352211e  =>  ownerOf(uint256)
42842e0e  =>  safeTransferFrom(address,address,uint256)
081812fc  =>  getApproved(uint256)
a22cb465  =>  setApprovalForAll(address,bool)
e985e9c5  =>  isApprovedForAll(address,address)
b88d4fde  =>  safeTransferFrom(address,address,uint256,bytes)
150b7a02  =>  onERC721Received(address,address,uint256,bytes)
06fdde03  =>  name()
95d89b41  =>  symbol()
c87b56dd  =>  tokenURI(uint256)
2f745c59  =>  tokenOfOwnerByIndex(address,uint256)
4f6ccce7  =>  tokenByIndex(uint256)
24a084df  =>  sendValue(address,uint256)
a0b5ffb0  =>  functionCall(address,bytes)
241b5886  =>  functionCall(address,bytes,string)
2a011594  =>  functionCallWithValue(address,bytes,uint256)
d525ab8a  =>  functionCallWithValue(address,bytes,uint256,string)
c21d36f3  =>  functionStaticCall(address,bytes)
dbc40fb9  =>  functionStaticCall(address,bytes,string)
ee33b7e2  =>  functionDelegateCall(address,bytes)
57387df0  =>  functionDelegateCall(address,bytes,string)
18c2c6a2  =>  _verifyCallResult(bool,bytes,string)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
6900a3ae  =>  toString(uint256)
8fba8d5c  =>  toHexString(uint256)
63e1cbea  =>  toHexString(uint256,uint256)
743976a0  =>  _baseURI()
01538868  =>  _setTokenURI(uint256,string)
24b6b8c0  =>  _safeTransfer(address,address,uint256,bytes)
f8e76cc0  =>  _exists(uint256)
4cdc9549  =>  _isApprovedOrOwner(address,uint256)
```

```
b3e1c718  =>  _safeMint(address,uint256)
6a4f832b  =>  _safeMint(address,uint256,bytes)
d3fc9864  =>  mint(address,uint256,string)
4e6ec247  =>  _mint(address,uint256)
9b1f9e74  =>  _burn(uint256)
30e0789e  =>  _transfer(address,address,uint256)
7b7d7225  =>  _approve(address,uint256)
1fd01de1  =>  _checkOnERC721Received(address,address,uint256,bytes)
cad3be83  =>  _beforeTokenTransfer(address,address,uint256)
5ebb5a65  =>  setreferralFees(uint256)
2c1e816d  =>  setAdminAddress(address)
c067ccca  =>  setCreatonFees(uint256)
916642f7  =>  setSellingFees(uint256)
863d4f5c  =>  mintNFT(string,uint256,uint256,uint256,uint256,address)
c18bc5b8  =>  buyWithBNB(uint256,uint256,uint256,address)
2b8ed3de  =>  buyWithTokens(uint256,uint256,uint256,uint256,address,address)
459e7e93  =>  calclulateAmount(uint256,address,uint256,uint256,address)
c873f4c6  =>  claimAmount(uint256,address,uint256)
```

Automatic general report

```
Files Description Table

|  File Name  |  SHA-1 Hash  |
|-------------|--------------|
| /Users/macbook/Desktop/smart contracts/NFTTIMES.sol |
3a6f7a89a144e2cd58c814c1f3be6949f2437723 |

 Contracts Description Table

|  Contract  |         Type        |       Bases      |                  |
|
|:----------:|:-------------------:|:----------------:|:----------------:|:--------
-------:|
|     └      |  **Function Name**  |  **Visibility**  |  **Mutability**  |
**Modifiers**  |
||||||
| **IBEP20** | Interface |  |||
| └ | totalSupply | External ❗️ |   |NO❗️ |
| └ | balanceOf | External ❗️ |   |NO❗️ |
| └ | transfer | External ❗️ | 🛑  |NO❗️ |
| └ | allowance | External ❗️ |   |NO❗️ |
| └ | approve | External ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | External ❗️ | 🛑  |NO❗️ |
||||||
| **SafeMath** | Library |  |||
| └ | add | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | sub | Internal 🔒 |   | |
| └ | mul | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | div | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
| └ | mod | Internal 🔒 |   | |
||||||
| **Math** | Library |  |||
| └ | max | Internal 🔒 |   | |
| └ | min | Internal 🔒 |   | |
| └ | average | Internal 🔒 |   | |
||||||
| **IERC165** | Interface |  |||
| └ | supportsInterface | External ❗️ |   |NO❗️ |
||||||
| **IERC721** | Interface | IERC165 |||
| └ | balanceOf | External ❗️ |   |NO❗️ |
| └ | ownerOf | External ❗️ |   |NO❗️ |
| └ | safeTransferFrom | External ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | External ❗️ | 🛑  |NO❗️ |
| └ | approve | External ❗️ | 🛑  |NO❗️ |
| └ | getApproved | External ❗️ |   |NO❗️ |
| └ | setApprovalForAll | External ❗️ | 🛑  |NO❗️ |
| └ | isApprovedForAll | External ❗️ |   |NO❗️ |
| └ | safeTransferFrom | External ❗️ | 🛑  |NO❗️ |
||||||
| **IERC721Receiver** | Interface |  |||
| └ | onERC721Received | External ❗️ | 🛑  |NO❗️ |
||||||
| **IERC721Metadata** | Interface | IERC721 |||
| └ | name | External ❗️ |   |NO❗️ |
| └ | symbol | External ❗️ |   |NO❗️ |
| └ | tokenURI | External ❗️ |   |NO❗️ |
||||||
| **IERC721Enumerable** | Interface | IERC721 |||
| └ | totalSupply | External ❗️ |   |NO❗️ |
| └ | tokenOfOwnerByIndex | External ❗️ |   |NO❗️ |
| └ | tokenByIndex | External ❗️ |   |NO❗️ |
||||||
| **Address** | Library |  |||
| └ | isContract | Internal 🔒 |   | |
| └ | sendValue | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCall | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionCallWithValue | Internal 🔒 | 🛑  | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionStaticCall | Internal 🔒 |   | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | functionDelegateCall | Internal 🔒 | 🛑  | |
| └ | _verifyCallResult | Private 🔐 |   | |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **Strings** | Library |  |||
| └ | toString | Internal 🔒 |   | |
| └ | toHexString | Internal 🔒 |   | |
| └ | toHexString | Internal 🔒 |   | |
||||||
| **ERC165** | Implementation | IERC165 |||
| └ | supportsInterface | Public ❗️ |   |NO❗️ |
||||||
| **ERC721** | Implementation | Context, ERC165, IERC721, IERC721Metadata |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | supportsInterface | Public ❗️ |   |NO❗️ |
| └ | balanceOf | Public ❗️ |   |NO❗️ |
| └ | ownerOf | Public ❗️ |   |NO❗️ |
| └ | name | Public ❗️ |   |NO❗️ |
| └ | symbol | Public ❗️ |   |NO❗️ |
| └ | _baseURI | Internal 🔒 |   | |
| └ | tokenURI | Public ❗️ |   |NO❗️ |
| └ | _setTokenURI | Internal 🔒 | 🛑  | |
| └ | approve | Public ❗️ | 🛑  |NO❗️ |
| └ | getApproved | Public ❗️ |   |NO❗️ |
| └ | setApprovalForAll | Public ❗️ | 🛑  |NO❗️ |
| └ | isApprovedForAll | Public ❗️ |   |NO❗️ |
| └ | transferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | safeTransferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | safeTransferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | _safeTransfer | Internal 🔒 | 🛑  | |
| └ | _exists | Internal 🔒 |   | |
| └ | _isApprovedOrOwner | Internal 🔒 |   | |
| └ | _safeMint | Internal 🔒 | 🛑  | |
| └ | _safeMint | Internal 🔒 | 🛑  | |
| └ | mint | Internal 🔒 | 🛑  | |
| └ | _mint | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _checkOnERC721Received | Private 🔐 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |
||||||
| **NFTTIMES** | Implementation | ERC721 |||
| └ | <Constructor> | Public ❗️ | 💵 | ERC721 |
| └ | setreferralFees | Public ❗️ | 🛑  |NO❗️ |
| └ | setAdminAddress | Public ❗️ | 🛑  |NO❗️ |
| └ | setCreatonFees | Public ❗️ | 🛑  |NO❗️ |
| └ | setSellingFees | Public ❗️ | 🛑  |NO❗️ |
| └ | mintNFT | Public ❗️ | 💵 |NO❗️ |
| └ | buyWithBNB | Public ❗️ | 💵 |NO❗️ |
| └ | buyWithTokens | Public ❗️ | 🛑  |NO❗️ |
| └ | calclulateAmount | Internal 🔒 | 🛑  | |
| └ | claimAmount | Public ❗️ | 💵 |NO❗️ |

 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 and no need for redeploy the contract.

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.