# Smart Contract
# Security Audit
# V1

## For Loot And Glory Royalty Smart Contract

23/4/2022

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**: Polygon
- **Contract Address**: 0xE42517349ebf890F8899d89edA47b391CD6F545f
- **Code:**

https://mumbai.polygonscan.com/address/0x11eee4a744149419ddffc1b8bfb5d3f1c7cc7b37#code

## Smart Contract Information

- Name: For Loot And Glory Royalty
- Type: It is a Stake contract for FLAG token.
- Token: https://mumbai.polygonscan.com/address/0x8208b50a7cb9ab44ee773f5f48721edc8a71cce7

## Contracts address deployed to test net (Polygon)

For Loot And Glory Royalty contract on Polygon test net to test every function by the auditor.

https://mumbai.polygonscan.com/address/0x4e64810a3c56a098027dfe292836d437be3a8f7b

# Executive Summary

According to our assessment, the customer`s solidity smart contract is **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, 1 medium, 2 low, 0 very low-level issues and 0 note in all solidity files of the contract

The files:

ForLootAndGloryRoyalty.sol

# File and Function Level Report

## File in Scope:

| Contract Name | SHA 256 hash | Contract Address |
|---|---|---|
| ForLootAndGloryRoyalty.sol | **d4df14764007d98e87ab3540c38b1c842f1b9c99783903393103d6899e13444f** | 0xE42517349ebf890F8899d89edA47b391CD6F545f |

- Contract: ForLootAndGloryRoyalty
- Inherit: AccessControl, ReentrancyGuard, Pausable
- Observation: All passed including security check
- Test Report: passed
- Score: passed
- Conclusion: passed

| Function | Test Result | Type / Return Type | Score |
|---|---|---|---|
| addressStakedBalance | ✓ | Read / public | **Passed** |
| bigMultiplier | ✓ | Read / public | **Passed** |
| dailyEmissionsRate | ✓ | Read / public | **Passed** |
| DEFAULT_ADMIN_ROLE | ✓ | Read / public | **Passed** |
| DEPOSIT_ROLE | ✓ | Read / public | **Passed** |
| getRoleAdmin | ✓ | Read / public | **Passed** |
| hasRole | ✓ | Read / public | **Passed** |
| isStaker | ✓ | Read / public | **Passed** |
| lastRewardTime | ✓ | Read / public | **Passed** |
| paused | ✓ | Read / public | **Passed** |
| PAUSER_ROLE | ✓ | Read / public | **Passed** |
| rewardsBalance | ✓ | Read / public | **Passed** |

| | | | |
|---|---|---|---|
| showBigRewardsPerToken | ✓ | Read / public | **Passed** |
| showRewardToken | ✓ | Read / public | **Passed** |
| showStakingToken | ✓ | Read / public | **Passed** |
| supportsInterface | ✓ | Read / public | **Passed** |
| timeSinceLastReward | ✓ | Read / public | **Passed** |
| totalStakedSupply | ✓ | Read / public | **Passed** |
| totalWeightedStakedSupply | ✓ | Read / public | **Passed** |
| userPendingRewards | ✓ | Read / public | **Passed** |
| depositRewards | ✓ | Write / public | **Passed** |
| createStake | ✓ | Write / public | **Passed** |
| emergencyUnlockToggle | ✓ | Write / public | **Passed** |
| emergencyUnStake | ✓ | Write / public | **Passed** |
| grantRole | ✓ | Write / public | **Passed** |
| pause | ✓ | Write / public | **Passed** |
| getRewards | ✓ | Write / public | **Passed** |
| removeStake | ✓ | Write / public | **Passed** |
| renounceRole | ✓ | Write / public | **Passed** |
| revokeRole | ✓ | Write / public | **Passed** |
| setDailyEmissions | ✓ | Write / public | **Passed** |
| unpause | ✓ | Write / public | **Passed** |
| updateBigRewardsPerToken | ✓ | Write / public | **Passed** |
| withdrawRewards | ✓ | 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 | 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 with Notes** |
| 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,<br>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:**

#Centralization Risks

Description

The role owner has the authority to :
- Can Pause /un paused the contract.

It has to be noted that the owner can paused / un paused the contract at any time.

```
function pause() external onlyRole(PAUSER_ROLE) {
        _pause();
    }

    function unpause() external onlyRole(PAUSER_ROLE) {
        _unpause();
    }
```

Remediation

Make these functions internal in next version or the team should announce the investors before pause and un pause to give them time if they want to do anything.

P.S: This issue is common to the majority of Stake smart contracts.

Status: Acknowledged by the Auditee

**Low:**

#Missing zero address validation

Description

When setting the FLAG token address, and the reward token address too, it should be checked for zero address. Otherwise, tokens sent to the zero address may be burnt forever.

```
constructor(address _stakingToken, address _rewardToken) {
        _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);
        _grantRole(PAUSER_ROLE, msg.sender);
        _grantRole(DEPOSIT_ROLE, msg.sender);
        stakingToken = IERC20(_stakingToken);
        rewardToken = IERC20(_rewardToken);
        lastRewardTime = block.timestamp; // set this to when the contract is
```

```
initiated so the first harvest isn't huge!
        dailyEmissionsRate = 1000000000000000000; // daily emissions for the
initial
        updateBigRewardsPerToken();
    }
```

Remediation

Use a require statement to check for zero address when setting the FLAG token address, and reward address too.

Status: Closed. Fixed in version 2.

#Use of block.timestamp for comparisons

Description

The value of block.timestamp can be manipulated by the miner.
And conditions with strict equality is difficult to achieve - block.timestamp

Remediation

Avoid use of block.timestamp

Status: Acknowledged

**Very Low:**

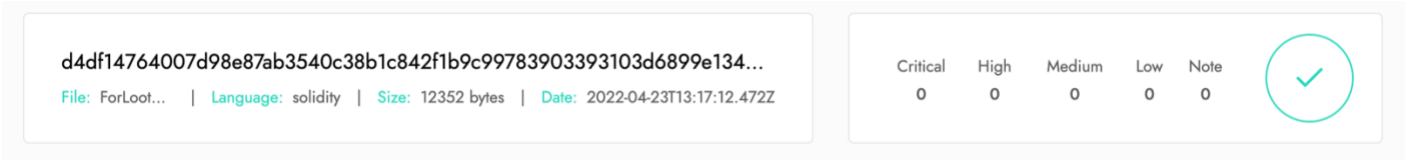
No Very Low severity vulnerabilities were found.

**Notes:**

No Notes vulnerabilities were found.

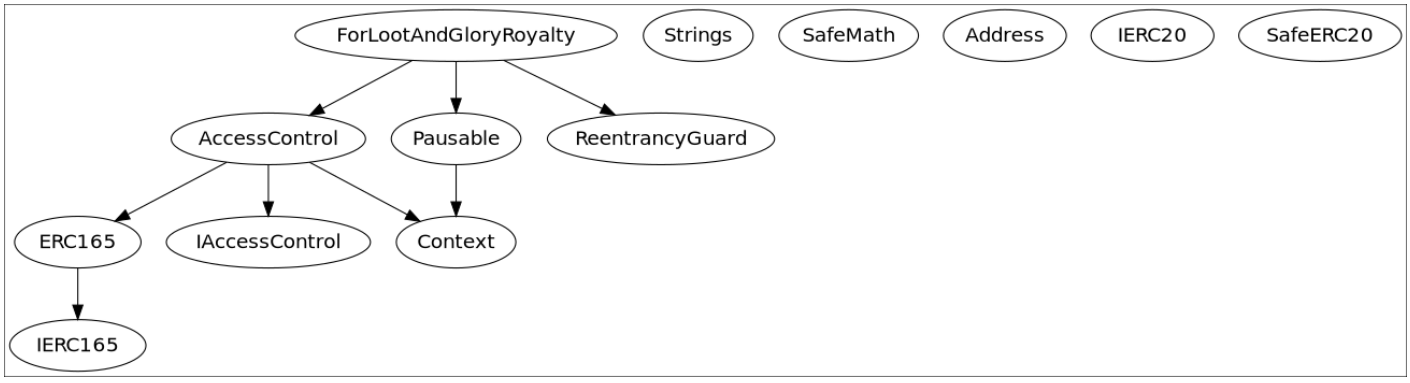
# Automatic Testing

1- Check for security

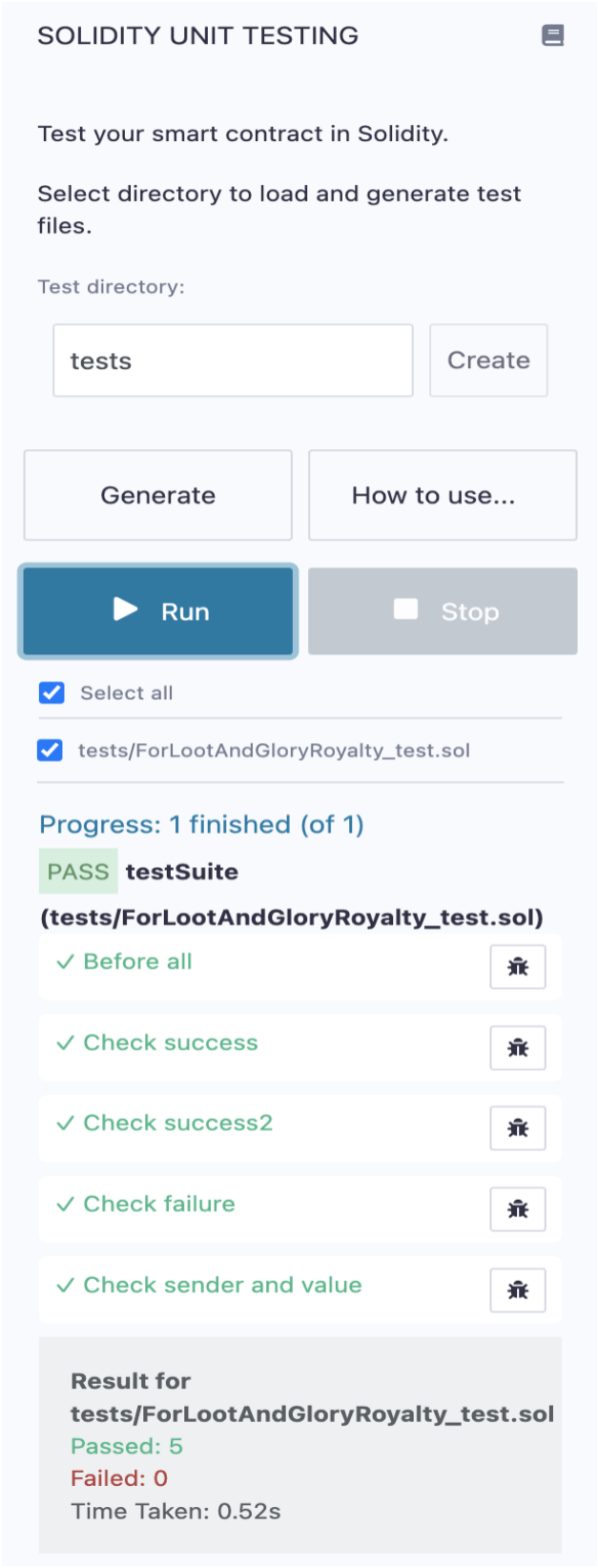



2- SOLIDITY STATIC ANALYSIS

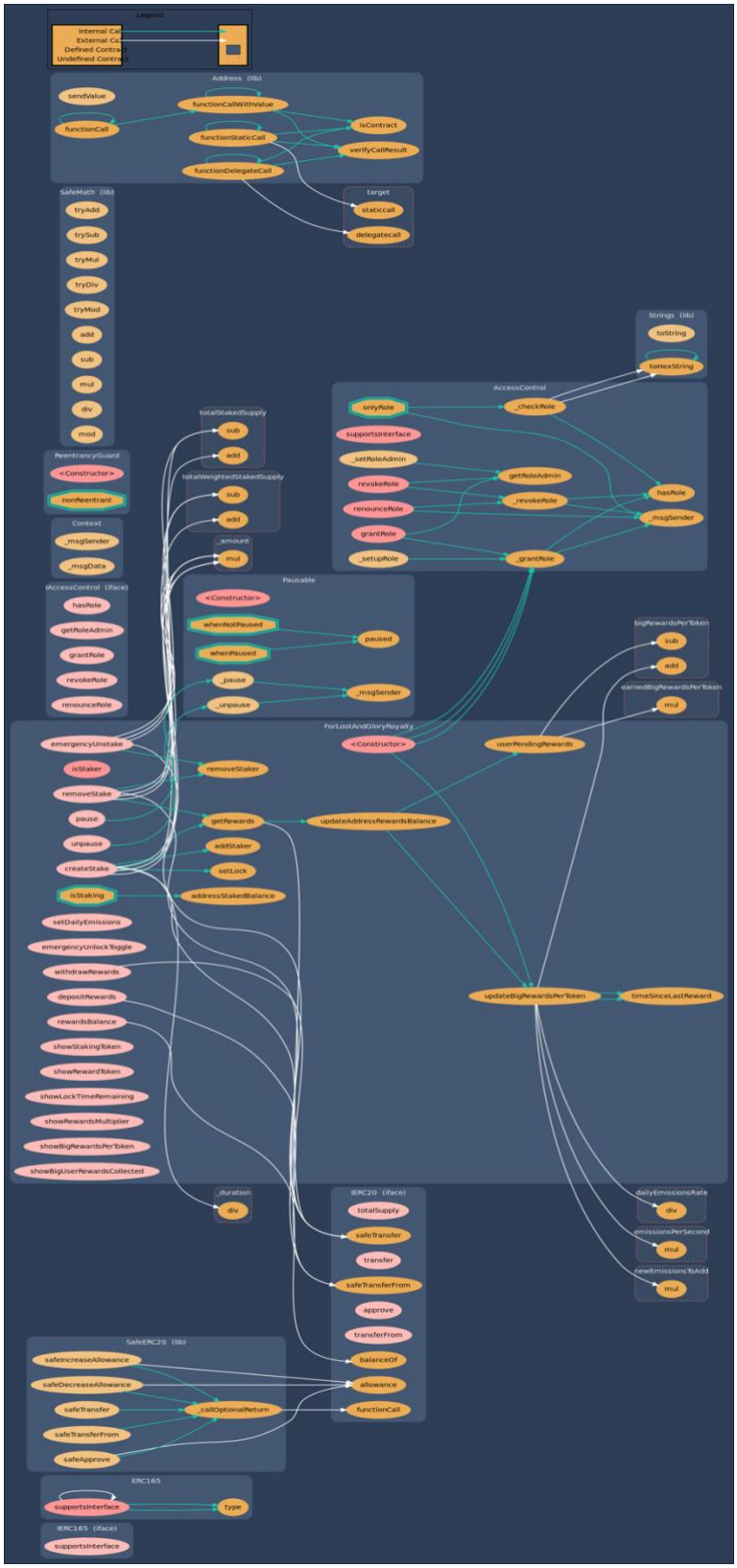

3- Inheritance graph



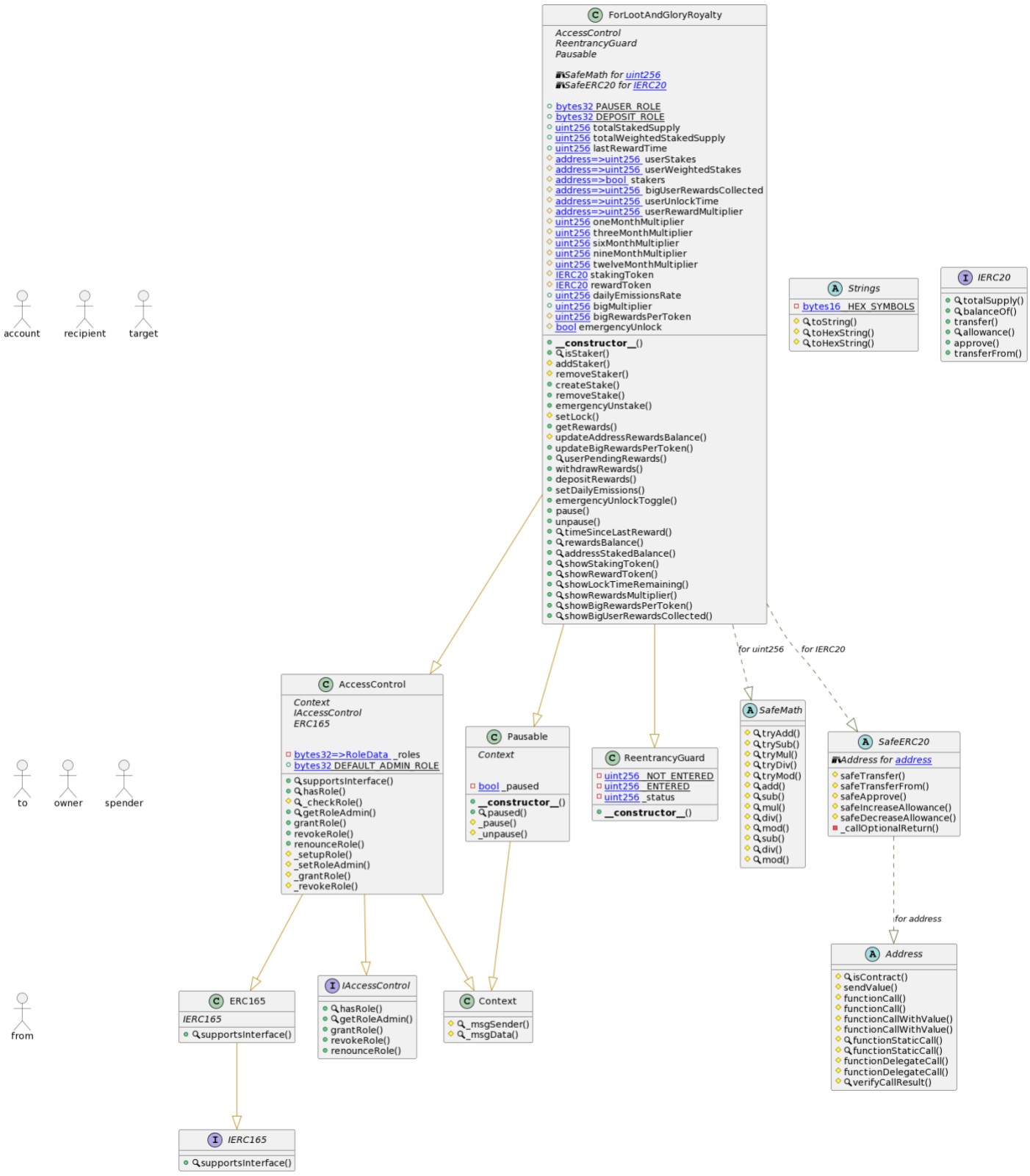
# 4- SOLIDITY UNIT TESTING



5- Call graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Address (lib)
sendValue
functionCallWithValue
functionCall
isContract
functionStaticCall
verifyCallResult
functionDelegateCall
target
staticcall
delegatecall
SafeMath (lib)
tryAdd
trySub
tryMul
tryDiv
tryMod
add
sub
mul
div
mod
Strings (lib)
toString
toHexString
AccessControl
onlyRole
_checkRole
supportsInterface
_setRoleAdmin
getRoleAdmin
revokeRole
_revokeRole
renounceRole
hasRole
_msgSender
grantRole
_setupRole
_grantRole
totalStakedSupply
sub
add
totalWeightedStakedSupply
sub
add
_amount
mul
ReentrancyGuard
<Constructor>
nonReentrant
Context
_msgSender
_msgData
IAccessControl (iface)
hasRole
getRoleAdmin
grantRole
revokeRole
renounceRole
Pausable
<Constructor>
whenNotPaused
paused
whenPaused
_pause
_unpause
_msgSender
bigRewardsPerToken
sub
add
earnedBigRewardsPerToken
mul
ForLootAndGloryRoyalty
<Constructor>
userPendingRewards
emergencyUnstake
isStaker
removeStaker
removeStake
pause
unpause
createStake
isStaking
setDailyEmissions
emergencyUnlockToggle
withdrawRewards
depositRewards
rewardsBalance
showStakingToken
showRewardToken
showLockTimeRemaining
showRewardsMultiplier
showBigRewardsPerToken
showBigUserRewardsCollected
getRewards
updateAddressRewardsBalance
addStaker
setLock
addressStakedBalance
updateBigRewardsPerToken
timeSinceLastReward
_duration
div
IERC20 (iface)
totalSupply
safeTransfer
transfer
safeTransferFrom
approve
transferFrom
balanceOf
allowance
functionCall
dailyEmissionsRate
div
emissionsPerSecond
mul
newEmissionsToAdd
mul
SafeERC20 (lib)
safeIncreaseAllowance
safeDecreaseAllowance
safeTransfer
safeTransferFrom
safeApprove
_callOptionalReturn
ERC165
supportsInterface
type
IERC165 (iface)
supportsInterface

# Unified Modeling Language (UML)



# Functions signature

```
Sighash   |   Function Signature
========================
16279055  =>  isContract(address)
01ffc9a7  =>  supportsInterface(bytes4)
6900a3ae  =>  toString(uint256)
8fba8d5c  =>  toHexString(uint256)
63e1cbea  =>  toHexString(uint256,uint256)
91d14854  =>  hasRole(bytes32,address)
248a9ca3  =>  getRoleAdmin(bytes32)
2f2ff15d  =>  grantRole(bytes32,address)
d547741f  =>  revokeRole(bytes32,address)
36568abe  =>  renounceRole(bytes32,address)
119df25f  =>  _msgSender()
8b49d47e  =>  _msgData()
c2985578  =>  foo()
5b7b2c38  =>  _checkRole(bytes32,address)
4fa943a6  =>  _setupRole(bytes32,address)
7612997d  =>  _setRoleAdmin(bytes32,bytes32)
ce2cc1d0  =>  _grantRole(bytes32,address)
2c95bd23  =>  _revokeRole(bytes32,address)
5c975abb  =>  paused()
320b2ad9  =>  _pause()
fc8234cb  =>  _unpause()
884557bf  =>  tryAdd(uint256,uint256)
a29962b1  =>  trySub(uint256,uint256)
6281efa4  =>  tryMul(uint256,uint256)
736ecb18  =>  tryDiv(uint256,uint256)
38dc0867  =>  tryMod(uint256,uint256)
771602f7  =>  add(uint256,uint256)
b67d77c5  =>  sub(uint256,uint256)
c8a4ac9c  =>  mul(uint256,uint256)
a391c15b  =>  div(uint256,uint256)
f43f523a  =>  mod(uint256,uint256)
e31bdc0a  =>  sub(uint256,uint256,string)
b745d336  =>  div(uint256,uint256,string)
71af23e8  =>  mod(uint256,uint256,string)
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)
946b5793  =>  verifyCallResult(bool,bytes,string)
18160ddd  =>  totalSupply()
70a08231  =>  balanceOf(address)
a9059cbb  =>  transfer(address,uint256)
dd62ed3e  =>  allowance(address,address)
095ea7b3  =>  approve(address,uint256)
23b872dd  =>  transferFrom(address,address,uint256)
d0c407e1  =>  safeTransfer(IERC20,address,uint256)
```

```
5beae096  =>  safeTransferFrom(IERC20,address,address,uint256)
d6dcec8d  =>  safeApprove(IERC20,address,uint256)
390cc046  =>  safeIncreaseAllowance(IERC20,address,uint256)
5164ffed  =>  safeDecreaseAllowance(IERC20,address,uint256)
becc5a20  =>  _callOptionalReturn(IERC20,bytes)
6f1e8533  =>  isStaker(address)
2466696e  =>  addStaker(address)
96a9fdc0  =>  removeStaker(address)
d7fa023d  =>  createStake(uint256,uint256)
939624ab  =>  removeStake(uint256)
012ce501  =>  emergencyUnstake(uint256)
d3e15747  =>  setLock(uint256)
0572b0cc  =>  getRewards()
a8ce4a90  =>  updateAddressRewardsBalance(address)
701f53ba  =>  updateBigRewardsPerToken()
fc9d7711  =>  userPendingRewards(address)
9342c8f4  =>  withdrawRewards(uint256)
8bdf67f2  =>  depositRewards(uint256)
366667aa  =>  setDailyEmissions(uint256)
e7228051  =>  emergencyUnlockToggle()
8456cb59  =>  pause()
3f4ba83a  =>  unpause()
fadac5c0  =>  timeSinceLastReward()
131f2e53  =>  rewardsBalance()
d774cdd0  =>  addressStakedBalance(address)
33a00bb8  =>  showStakingToken()
cb3992be  =>  showRewardToken()
29fc7b50  =>  showLockTimeRemaining(address)
b1c52f36  =>  showRewardsMultiplier()
d8a839ed  =>  showBigRewardsPerToken()
4b1cfea0  =>  showBigUserRewardsCollected()
```

# Automatic general report

Files Description Table

| File Name | SHA-1 Hash |
|---|---|
| /Users/macbook/Desktop/smart contracts/ForLootAndGloryRoyalty.sol | 34dae672b682830e1019ff433f041be7875f9572 |

Contracts Description Table

| Contract | Type | Bases | | |
|---|---|---|---|---|
| └ | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| **IERC165** | Interface | | | |
| └ | supportsInterface | External ❗ | | NO❗ |
| **ERC165** | Implementation | IERC165 | | |
| └ | supportsInterface | Public ❗ | | NO❗ |
| **Strings** | Library | | | |
| └ | toString | Internal 🔒 | | |
| └ | toHexString | Internal 🔒 | | |
| └ | toHexString | Internal 🔒 | | |
| **IAccessControl** | Interface | | | |
| └ | hasRole | External ❗ | | NO❗ |
| └ | getRoleAdmin | External ❗ | | NO❗ |
| └ | grantRole | External ❗ | 🛑 | NO❗ |
| └ | revokeRole | External ❗ | 🛑 | NO❗ |
| └ | renounceRole | External ❗ | 🛑 | NO❗ |
| **Context** | Implementation | | | |
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | Internal 🔒 | | |
| **AccessControl** | Implementation | Context, IAccessControl, ERC165 | | |
| └ | supportsInterface | Public ❗ | | NO❗ |
| └ | hasRole | Public ❗ | | NO❗ |
| └ | _checkRole | Internal 🔒 | | |
| └ | getRoleAdmin | Public ❗ | | NO❗ |
| └ | grantRole | Public ❗ | 🛑 | onlyRole |
| └ | revokeRole | Public ❗ | 🛑 | onlyRole |
| └ | renounceRole | Public ❗ | 🛑 | NO❗ |
| └ | _setupRole | Internal 🔒 | 🛑 | |
| └ | _setRoleAdmin | Internal 🔒 | 🛑 | |
| └ | _grantRole | Internal 🔒 | 🛑 | |
| └ | _revokeRole | Internal 🔒 | 🛑 | |
| **Pausable** | Implementation | Context | | |

| └ | <Constructor> | Public ❗ | 🛑 |NO❗ |
| └ | paused | Public ❗ | |NO❗ |
| └ | _pause | Internal 🔒 | 🛑 | whenNotPaused |
| └ | _unpause | Internal 🔒 | 🛑 | whenPaused |
||||||
| **ReentrancyGuard** | Implementation | |||
| └ | <Constructor> | Public ❗ | 🛑 |NO❗ |
||||||
| **SafeMath** | Library | |||
| └ | tryAdd | Internal 🔒 | | |
| └ | trySub | Internal 🔒 | | |
| └ | tryMul | Internal 🔒 | | |
| └ | tryDiv | Internal 🔒 | | |
| └ | tryMod | Internal 🔒 | | |
| └ | add | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | mul | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |
| └ | mod | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |
| └ | mod | Internal 🔒 | | |
||||||
| **Address** | Library | |||
| └ | isContract | Internal 🔒 | | |
| └ | sendValue | Internal 🔒 | 🛑 | |
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | functionStaticCall | Internal 🔒 | | |
| └ | functionStaticCall | Internal 🔒 | | |
| └ | functionDelegateCall | Internal 🔒 | 🛑 | |
| └ | functionDelegateCall | Internal 🔒 | 🛑 | |
| └ | verifyCallResult | Internal 🔒 | | |
||||||
| **IERC20** | Interface | |||
| └ | totalSupply | External ❗ | |NO❗ |
| └ | balanceOf | External ❗ | |NO❗ |
| └ | transfer | External ❗ | 🛑 |NO❗ |
| └ | allowance | External ❗ | |NO❗ |
| └ | approve | External ❗ | 🛑 |NO❗ |
| └ | transferFrom | External ❗ | 🛑 |NO❗ |
||||||
| **SafeERC20** | Library | |||
| └ | safeTransfer | Internal 🔒 | 🛑 | |
| └ | safeTransferFrom | Internal 🔒 | 🛑 | |
| └ | safeApprove | Internal 🔒 | 🛑 | |
| └ | safeIncreaseAllowance | Internal 🔒 | 🛑 | |
| └ | safeDecreaseAllowance | Internal 🔒 | 🛑 | |
| └ | _callOptionalReturn | Private 🔐 | 🛑 | |
||||||
| **ForLootAndGloryRoyalty** | Implementation | AccessControl, ReentrancyGuard, Pausable |||
| └ | <Constructor> | Public ❗ | 🛑 |NO❗ |

| └ | isStaker | Public ❗ | | NO❗ |
| --- | --- | --- | --- | --- |
| └ | addStaker | Internal 🔒 | 🛑 | whenNotPaused |
| └ | removeStaker | Internal 🔒 | 🛑 | whenNotPaused |
| └ | createStake | External ❗ | 🛑 | whenNotPaused |
| └ | removeStake | External ❗ | 🛑 | whenNotPaused isStaking |
| └ | emergencyUnstake | External ❗ | 🛑 | NO❗ |
| └ | setLock | Internal 🔒 | 🛑 | |
| └ | getRewards | Public ❗ | 🛑 | nonReentrant whenNotPaused |
| └ | updateAddressRewardsBalance | Internal 🔒 | 🛑 | |
| └ | updateBigRewardsPerToken | Public ❗ | 🛑 | NO❗ |
| └ | userPendingRewards | Public ❗ | | NO❗ |
| └ | withdrawRewards | External ❗ | 🛑 | onlyRole |
| └ | depositRewards | External ❗ | 🛑 | onlyRole |
| └ | setDailyEmissions | External ❗ | 🛑 | onlyRole |
| └ | emergencyUnlockToggle | External ❗ | 🛑 | onlyRole |
| └ | pause | External ❗ | 🛑 | onlyRole |
| └ | unpause | External ❗ | 🛑 | onlyRole |
| └ | timeSinceLastReward | Public ❗ | | NO❗ |
| └ | rewardsBalance | External ❗ | | NO❗ |
| └ | addressStakedBalance | Public ❗ | | NO❗ |
| └ | showStakingToken | External ❗ | | NO❗ |
| └ | showRewardToken | External ❗ | | NO❗ |
| └ | showLockTimeRemaining | External ❗ | | NO❗ |
| └ | showRewardsMultiplier | External ❗ | | NO❗ |
| └ | showBigRewardsPerToken | External ❗ | | NO❗ |
| └ | showBigUserRewardsCollected | External ❗ | | 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 to 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 “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.