smart state

> Smart Contract Audit

Finnovant

Feb 27
2021

TABLE OF CONTENTS

METHODOLOGY

MAIN TESTS LIST:

- Best code practices
- ERC20/BEP20 compliance (if applicable)
- Logical bugs
- General Denial Of Service(DOS)
- Locked ether
- Private data leaks
- Using components with known vulns
- Weak PRNG
- Unsed vars
- Uncheked call return method
- Code with no effects
- Pool Asset Security (backdoors in the underlying ERC-20)
- Function visibility
- Use of deprecated functions
- Authorization issues
- Re-entrancy
- Arithmetic Over/Under Flows
- Hidden Malicious Code
- External Contract Referencing
- Short Address/ Parameter Attack
- Race Conditions / Front Running
- Uninitialized Storage Pointers
- Floating Points and Precision
- Signatures Replay

STRUCTURE OF CONTRACT

BIOFI_ARRAY.SOL

CHECK SUMMARY:

We recommend to rewrite the token using openzeppelin:

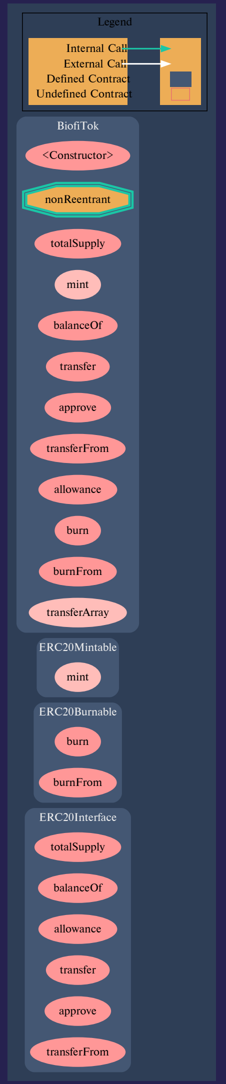
https://github.com/OpenZeppelin/openzeppelin-contracts

CONTRACT METHODS ANALYSIS:

- modifier nonReentrant()
  guardCounter is always incremented by 1 and is permanently increasing. It should be reset after successful execution

WARNING !

- totalSupply()
  total supply doesn't count balance of address 0, so in case someone transfers tokens to 0 address it means they are burned, however they are actually not. Total supply shouldn't have any exclusions and provide a correct information.



Pic. 1.1
biofi_array.sol

- **mint(address _to, uint256 tokens)**
  Reccommended to use onlyOwner modifier from openzeppelin instead of duplicated owner check in every owner method. tokens < _totalSupply check seems redundant.

- **balanceOf(address tokenOwner)**
  Vulnerabilities not detected

WARNING !

- **transfer(address to, uint tokens)**
  isPaused is implelented, however it can not be set true, since setter method is implemented, so it is impossible to put contract on pause. tokens < _totalSupply check should be replaced with the amount of tokens user has on his balance compared to amount of tokens he is going to transfer.

WARNING !

- **approve(address spender, uint tokens)**
  isPaused is implelented, however it can not be set true, since setter method is implemented, so it is impossible to put contract on pause. tokens < _totalSupply seems redundant, because users are basically approving for max(uint256)

WARNING !

- **transferFrom(address from, address to, uint tokens)**
  isPaused is implelented, however it can not be set true, since setter method is implemented, so it is impossible to put contract on pause. tokens < _totalSupply check should be replaced with the amount of tokens user has on his balance compared to amount of tokens he is going to transfer. Allowance is not checked in the method.

- allowance(address tokenOwner, address spender)
  Vulnerabilities not detected

WARNING !

- burn(uint256 tokens)
  isPaused is implelented, however it can not be set true, since setter method is implemented, so it is impossible to put contract on pause. tokens < _totalSupply check should be replaced with the amount of tokens user has on his balance compared to amount of tokens he is going to burn.

WARNING !

- transferArray(address [] memory addresses, uint tokens)
  Reccommended to use onlyOwner modifier from openzeppelin instead of duplicated owner check in every owner method. isPaused is implelented, however it can not be set true, since setter method is implemented, so it is impossible to put contract on pause. tokens * addresses. length < _totalSupply should check for the amount of tokens user has, not for the totalSupply.

STRUCTURE OF CONTRACT

SEQUNETIAL_NFT.SOL

CHECK SUMMARY:

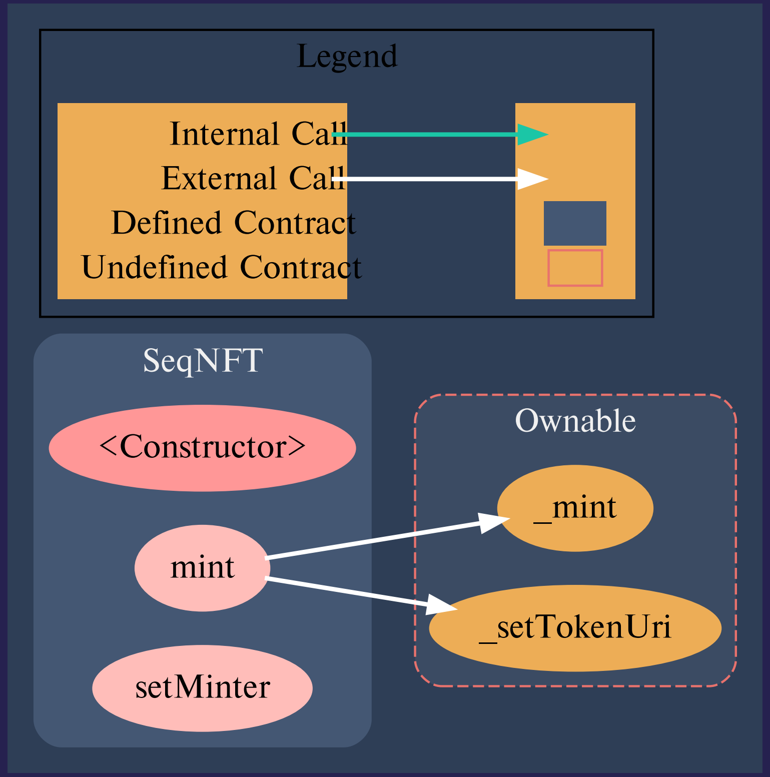
WARNING !

Token_id shouldn't be set to 1 because in that case due to mint method first token id is actually 2. NFT name and symbol should be changed according to the project.

CONTRACT METHODS ANALYSIS:

- mint(address _to, string calldata _uri)
  Vulnerabilities not detected

- setMinter(address newMinter)
  Function should emit an event.
  Recommended to use onlyOwner modifier



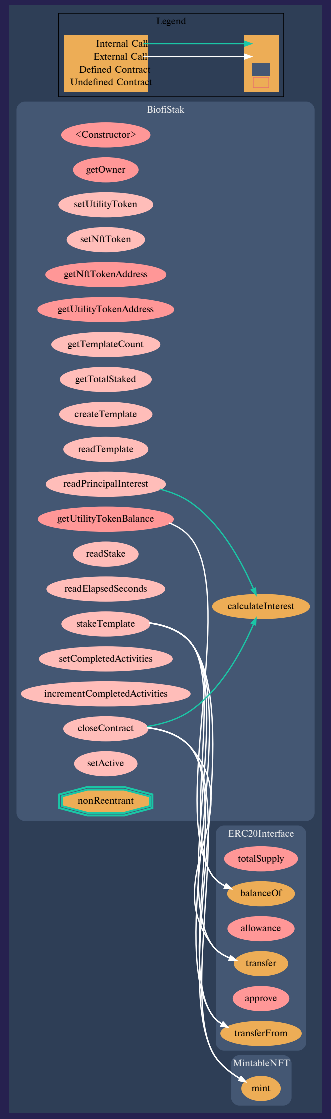
Pic. 1.2
sequnetial_nft.sol

STRUCTURE OF CONTRACT

BIOFI_STAKE.SOL

CONTRACT METHODS ANALYSIS:

- getOwner()
  Vulnerabilities not detected
- setUtilityToken(address token)
  Function should emit an event. Recommended to use onlyOwner modifier
- setNftToken(address token)
  Function should emit an event. Recommended to use onlyOwner modifier
- getNftTokenAddress()
  Vulnerabilities not detected



Pic. 1.3
biofi_stake.sol

- getUtilityTokenAddress()
  Vulnerabilities not detected

- getTemplateCount()
  Vulnerabilities not detected

- getTotalStaked(uint256 stakeIndex)
  Vulnerabilities not detected

- createTemplate(
  string calldata name, Timebox calldata timebox,
  Fraction calldata requiredPenalty, Fraction calldata apr,
  InvestmentRange calldata utilityInvestmentRange,
  string [] calldata beginNfts, string [] calldata endNfts)
  Function should emit an event. Recommended to use onlyOwner modifier

- readTemplate(uint256 templateIndex)
  Vulnerabilities not detected

PAYABLE/WARNING !

- stakeTemplate(uint256 stakeIndex, uint256 utilityTokenAmount)
  Anyone can stake tokens, whitelist or signature check should be implemented

Tokens in, NFT out

- readStake(address staker)
  Vulnerabilities not detected

- readElapsedSeconds(address staker)
  Vulnerabilities not detected

- calculateInterest(address staker)
  Incorrect calculation formula. Due to the fact that aprNumerator is always higher than aprDenumerator and at line 234 denumerator is subbed from numerator, which will basically cause a revert.

- readPrincipalInterest(address staker)
  Vulnerabilities not detected

- setCompletedActivities(address staker, uint256 completedActivities)
  Vulnerabilities not detected

- incrementCompletedActivities(address staker, uint256 completedActivities)
  Vulnerabilities not detected

PAYABLE

- closeContract(address staker)
  Vulnerabilities not detected

Tokens out, NFT out

- setActive(uint256 templateIndex, bool willBeActive)
  Vulnerabilities not detected

VERIFICATION CHECK SUMS

Contract Name	Bytecode hash (SHA 256)
biofi_array	41f0070d1818fedc8b82e60e47a515adade05d45581523255ed4003e95985e7f
sequnetial_nft	69852d2699d4bf4f449eba38376c90301b89b702341d9438fc8ab59d996d5f76
biofi_stake	305f8cf04f7933e7cc36350d016953310a7088755883969542e324e0a609caa8

Get In Touch

info@smartstate.tech

smartstate.tech