expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

StakingContract

TABLE OF CONTENTS

OVERVIEW

Contract Address: not deployed yet

Network: ---

Contract Type: Staking

Owner: ---

Deployer: ---

Checksum:
c5517d39c653012ea3a7da25b49dc8fd14e55b7c

Testnet version:
The tests conducted were performed on the contract deployed on the Binance Smart Chain (BSC) Testnet. https://testnet.bscscan.com/token/0x808886660ec773ea5d4b4e3645bdcd5c4afa3a59

OVERVIEW

Key points:

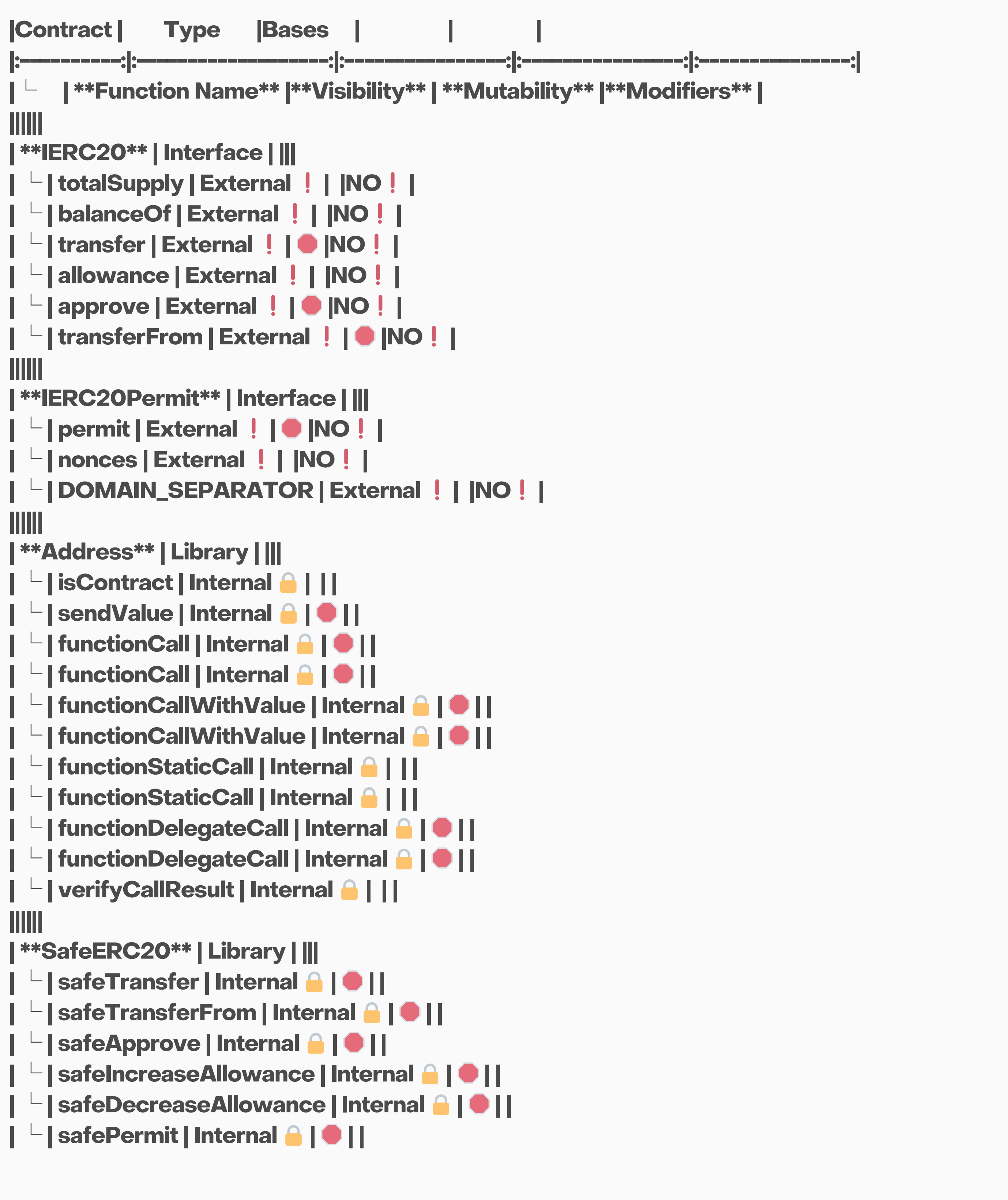
- Contract owner is able to adjust unstake fees within 0-10%
- Contract owner is able to withdraw all type of tokens from the contract
- 110% rewards per month
- staked tokens will be locked for 30 days

OVERVIEW

Risk Findings

Severity	Found	Pending	Resolved
● High	2	0	0
● Medium	0	0	0
● Low	0	2	0
● Informational	8	0	0
● Total	10	0	0

FUNCTION DETAILS



```
|Contract |      Type      |Bases    |           |           |
|:----------:|:-------------------:|:----------------:|:----------------:|:----------------:|
| └   | **Function Name** |**Visibility** | **Mutability** |**Modifiers** |
||||||
| **IERC20** | Interface | |||
| └ | totalSupply | External ❗ | |NO ❗ |
| └ | balanceOf | External ❗ | |NO ❗ |
| └ | transfer | External ❗ | 🛑 |NO ❗ |
| └ | allowance | External ❗ | |NO ❗ |
| └ | approve | External ❗ | 🛑 |NO ❗ |
| └ | transferFrom | External ❗ | 🛑 |NO ❗ |
||||||
| **IERC20Permit** | Interface | |||
| └ | permit | External ❗ | 🛑 |NO ❗ |
| └ | nonces | External ❗ | |NO ❗ |
| └ | DOMAIN_SEPARATOR | 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 | Internal 🔒 | | |
||||||
| **SafeERC20** | Library | |||
| └ | safeTransfer | Internal 🔒 | 🛑 | |
| └ | safeTransferFrom | Internal 🔒 | 🛑 | |
| └ | safeApprove | Internal 🔒 | 🛑 | |
| └ | safeIncreaseAllowance | Internal 🔒 | 🛑 | |
| └ | safeDecreaseAllowance | Internal 🔒 | 🛑 | |
| └ | safePermit | Internal 🔒 | 🛑 | |
```

FUNCTION DETAILS

```
| └ | _callOptionalReturn | Private 🔐 | 🛑 | |
||||||
| **Context** | Implementation | |||
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | Internal 🔒 | | |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗ | 🛑 |NO ❗ |
| └ | owner | Public ❗ | |NO ❗ |
| └ | renounceOwnership | Public ❗ | 🛑 | onlyOwner |
| └ | transferOwnership | Public ❗ | 🛑 | onlyOwner |
| └ | _transferOwnership | Internal 🔒 | 🛑 | |
||||||
| **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 🔒 | | |
||||||
| **StakingContract** | Implementation | Ownable, ReentrancyGuard |||
| └ | <Constructor> | Public ❗ | 🛑 |NO ❗ |
| └ | setFeeWallet | Public ❗ | 🛑 | onlyOwner |
| └ | _getNextDepositID | Private 🔐 | | |
| └ | _incrementDepositID | Private 🔐 | 🛑 | |
| └ | setLockPeriod | Public ❗ | 🛑 | onlyOwner |
| └ | deposit | External ❗ | 🛑 | nonReentrant |
| └ | setApr | Public ❗ | 🛑 | onlyOwner |
| └ | claimAllReward | Public ❗ | 🛑 | nonReentrant |
| └ | withdraw | Public ❗ | 🛑 | nonReentrant |
| └ | getOwnedDeposits | Public ❗ | |NO ❗ |
| └ | getAllClaimableReward | Public ❗ | |NO ❗ |
| └ | getApr | Public ❗ | |NO ❗ |
| └ | getBalance | Public ❗ | |NO ❗ |
| └ | getTotalRewards | Public ❗ | |NO ❗ |
```

FUNCTION DETAILS

```
| └ | sendRewards | Public ❗ | 🛑 | onlyOwner |
| └ | rescueTokens | External ❗ | 🛑 | onlyOwner |
| └ | setWithdrawFee | External ❗ | 🛑 | onlyOwner |
| └ | changePoolStatus | External ❗ | 🛑 | onlyOwner |
| └ | getTotalInvests | Public ❗ | |NO❗ |
### Legend
|Symbol | Meaning|
|:--------:|-----------|
| 🛑 | Function can modify state |
| 💵 | Function is payable |
```

TESTNET VERSION

Staking ✅
Tx:
- 2% sent to referrer
- Rewards saved after staking again
https://testnet.bscscan.com/tx/0xd722c205aa9096411b451fe4e9ce31dd8b904ef8d72bffa99d94c9d27eaea290

==

Claiming rewards ✅
- 795700% APR
- Rewards reset after claiming
https://testnet.bscscan.com/tx/0xaa0e652faab4f5d8a8b9e3ca1bd5c502d6acbaac4e7fd10d1f43b212cc961663

==

Withdrawing ✅
- Withdraw available after locking period
- 2% fee (currently) sent to fee receiver
- Rewards sent to staker
- Could not double withdraw same ID
https://testnet.bscscan.com/tx/0x376a3f2585ae8cc93a3c63ce9122c60a054840d0abd97c36589084556b46b870

RISK CLASSIFICATION

Risk Classification

High

Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, of the contract and its functions. Must be fixed as soon as possible.

Medium

Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Must be fxed as soon as possible.

Low

Effects are minimal in isolation and do not pose a signifcant danger to the project or its users. Issues under this classifcation are recommended to be fixed nonetheless.

Informational

A vulnerability that have informational character but is not effecting any of the code

RISK FINDING

Category: Centralization
Subject: Withdrawing ERC20 tokens
Status: Open
Impact: High

Overview:
Owner is able to withdraw any kind of token from the contract. A malicious owner may withdraw all staked tokens.

```
  function rescueTokens(
     address token,
     address to,
     uint256 amount
  ) external onlyOwner {
IERC20(token).transfer(to, amount);
  }
```

Suggestion:
Ensure that staked tokens will not be withdrawn by a malicious owner

```
  function rescueTokens(
     address token,
     address to,
     uint256 amount
  ) external onlyOwner {
     require(token != address(this));
IERC20(token).transfer(to, amount);
  }
```

or only allow withdrawal of tokens deposited by owner (as reward pool):

```
 uint256 public totalRewardsDeposited;

 function sendRewards(uint256 _amount) public onlyOwner {
IERC20(TokenContract).safeTransferFrom(
        msg.sender,
        address(this),
        _amount
     );
        totalRewardsDeposited += _amount;
  }

  function rescueTokens(
     address token,
     address to,
     uint256 amount
  ) external onlyOwner {
        require(amount <= totalRewardsDeposited);
IERC20(token).transfer(to, amount);}
```

RISK FINDING

Category: Logical

Subject: Referrer rewards are not deducted from deposited amount

Status: Open

Impact: High

Overview:

A referrer gets 2% of deposited tokens by staker, but this 2% is not deducted from deposited amount, hence, there will be an inconsistency between total staked tokens by investors and total available tokens:

total staked tokens = actual balance of contract + 2% of balance of contract

Suggestion:

deduct referrer reward from deposited amount

```
    if (investors[msg.sender].referrer != address(0)) {
      uint256 referrerAmount = (_amount * ref).div(percentRate);
investors[investors[msg.sender].referrer].referAmount = investors[
investors[msg.sender].referrer
].referAmount.add(referrerAmount);
IERC20(TokenContract).transfer(
investors[msg.sender].referrer,
referrerAmount
      );
      _depositAmount -= referrerAmount
    }
```

RISK FINDING

Category: **Informational and gas optimizations**
Subject: Several improvements and gas optimizations
Status: Open
Impact: **Low**
Overview:

- **withdraw** and **claimAllReward** functions are not emitting any events
- use **Indexed** keyword for all events to ease query of this events
- **TokenContract** can be declared as immutable
- **totalInvested** is not updated after unstaking tokens
- **totalInvestors** is not updated after unstaking all tokens
- increasing and assigning id can be done in one step
- create a function to allow emergency withdraw of tokens (without rewards and in locking period) but with considering a fee as penalty

uint256 Id = ++_currentDepositID;

- Referrer can be any arbitrary wallet, even a wallet that did not staked before

ABOUT EXPELEE

Expelee is a product-based aspirational Web3 start-up. Coping up with numerous solutions for blockchain security and constructing a Web3 ecosystem from deal making platform to developer hosting open platform, while also developing our own commercial and sustainable blockchain.

www.expelee.com

expeleeofficial	expelee
Expelee	expelee
expelee_official	expelee-co

expelee

Building the Futuristic Blockchain Ecosystem

DISCLAIMER

This audit report is prepared for informational purposes only and does not constitute financial, legal, or professional advice. The audit has been conducted based on the information provided by the project team and is limited to the code and documentation available up to the audit date. The assessment is focused on identifying potential vulnerabilities and security concerns within the provided codebase. It does not guarantee the absence of vulnerabilities or the security of the system. The audit does not include a comprehensive review of the project's business model, economic viability, or other non-technical aspects. The audit report is not an endorsement of the project, and readers should exercise their own judgment and due diligence before using, investing, or participating in the project. The project team and the auditors are not liable for any losses, damages, or expenses that may arise from actions taken based on this audit report. The audit is based on the current state of the code and may not account for future changes, updates, or modifications. Security is an ongoing process, and the project team is responsible for addressing any identified issues and maintaining the security of the system beyond the scope of this audit. Readers are encouraged to consult with their own professional advisors for advice tailored to their individual circumstances before making any decisions related to the project. This audit report is provided on an “as is’ basis and without any warranties, representations, or guarantees.

expelee

Building the Futuristic Blockchain Ecosystem