

expelee

Building the Futuristic **Blockchain Ecosystem**

Audit Report FOR



Elon Vs Twitter

OVERVIEW

Expelee team has performed a line-by-line manual analysis and automated review of the smart contract.

The smart contract was analysed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit :

 Audit Result	Passed
 KYC Verification	Not Done
 Audit Date	17 Sep 2022

Why Passed?

All the functionalities (depositing, withdrawing) has been reviewed, several non-critical issues were found that cannot disable the system.

- Team Expelee

PROJECT DESCRIPTION

Elon Vs Twitter Staking Contract

Elon Vs Twitter #EVT is setting up to make history in the crypto space. Our #EVT Reward system is unlike anything that has ever been seen before. We are unique, innovative and we are here to take over.

The reach and attention this trial will bring us is beyond anything you can imagine.

We will be presenting Elon Vs Twitter in front of the global media for the whole world to see.

 evt2022.com

 [ElonVsTwitter](#)

 [ElonVsTwitt](#)

*It's always good to check the social profiles of the project,
before making your investment.*

- Team Expelee

CONTRACT DETAILS

Contract Name

StakingPlatform

Optimization

Yes with 200 runs

Contract Address (Verified)

0x01d9a9e1c5720f6c753100785aa9b64ab91998e8

Network

BSC

Language

Solidity

Total Supply

NA

Decimals

NA

Compiler

v0.8.10+commit.fc410830

License

MIT license

Token Type

NA

Staking Contract SHA-256 Checksum

fa131a14dceadad94627b38680202576c5629f13861e3a4b0ed88d7942b7cbf3

AUDIT METHODOLOGY



Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.



Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch, that lead to scams and rugpulls.



Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
- Back-doors
- Vulnerability
- Accuracy
- Readability



Tools

- DE
- Open Zeppelin
- Code Analyzer
- Solidity Code
- Compiler
- Hardhat

RISK CLASSIFICATION

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

AUDIT SUMMARY

Used Tools

Slither, Echidna, etc - we used automated static-analysis tools to check contract for common solidity vulnerability & mistakes.

Manual Review:

we spent most of the audit process time reading the whole contract line by line.

Ownership :

Current owner of staking platform is:

0x9fe338a598ce1dd9dfc28de7b3f5be51c4b27f11

owner has several privileges over contract, all of them are discussed in the report

Contracts & Inheritance Tree:

All of below contracts are in this audit scope

- StakingPlatform.sol



MANUAL AUDIT

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to a methodology based on [OWASP](#) standards.

Vulnerabilities are divided into three primary risk categories: **high**, **medium**, and **low**.

High-level considerations for vulnerabilities span the following key areas when conducting assessments:

- Malicious Input Handling
- Escalation of privileges
- Arithmetic
- Gas use

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Findings Summary

- **High Risk Findings:** 0
- **Medium Risk Findings:** 2
- **Low Risk Findings:** 2
- **Suggestions & discussion:** 2
- **Gas Optimizations :** 2

Medium Risk Findings

Centralizatoin - Owner is able to set **limitFee** variable to an arbitrary number, setting this number to 100% will cause all of the reward tokens to be sent to **DAO** address

```
function setLimitTimeFee(uint256 _feePercent) external onlyOwner {
    limitFee = _feePercent;
}
```

Logical Issue - Potential revert at withdraw(uint256 amount) and withdrawAll() functions:

if Alice & Bob both deposit 100 EVT tokens, there will be 200 EVT tokens inside the contract, if Alice decide to withdraw her EVT tokens (assuming that she earned 5 EVT tokens), Staking contract will payher 105 tokens, if Bob then decide to withdraw all of his tokens, there wont be enough tokens to pay both his rewards and staking amount. **however the team is aware of this issue and will transfer theEVT tokens to stakign contract to pay for rewards after this audit is done.**

Low Risk Findings

Logical - **_userStartTime[_msgSender()]** at deposit function will get overridden at **_updateRewards** function

```
if (_userStartTime[_msgSender()] == 0) {
    _userStartTime[_msgSender()] = block.timestamp;
}
_updateRewards();
```

if **endPeriod** is set to 0 (before starting the staking) then **_updateRewards** will change **_userStartTime[_msgSender()]** to 0 as well

```
function _updateRewards() private {
    _rewardsToClaim[_msgSender()] = _calculateRewards(_msgSender());
    _userStartTime[_msgSender()] = (block.timestamp >= endPeriod)
    ? endPeriod
    : block.timestamp;
}
```

this is required to identify msgSender() as an early staker and giving it more rewards, but if this walletdeposit another amount later, because **_userStartTime[_msgSender()]** is 0 then this condition will be true:

```
if (_userStartTime[_msgSender()] == 0) {
    _userStartTime[_msgSender()] = block.timestamp;
}
```

and `_msgSender()` will not be considered as an early staker anymore

Recommendation:

define a mapping to keep track of early stakers

Centralization - Owner is able to withdraw : Contract BNB Tokens, Contract doge tokens & all otherERC20 tokens **except** EVT

Gas Optimizations

- define **_precision** as constant
- **_numberOfDaysForFee** & its setter **setNumberOfDaysForFee** never used inside the contract, delete them to lower contract size

Suggestions

- this function are changing contract state, but not emitting an event:
setNumberOfDaysForFee, **setDAO**, **setLimitTimeFee**, **setMinTokenStake**, **setMaxTokenStake**
- function **transferOwnership** overriden in StakingPlatform contract is redundant, this function is already implemented in Ownable contract

```
function transferOwnership(address newOwner) public override onlyOwner {  
    require(newOwner != address(0), "Ownable: new owner is the zero  
address");  
    _transferOwnership(newOwner);  
}
```


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

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document.

Always Do your own research and protect yourselves from being scammed. The Expelee team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools.

Under no circumstances did Expelee receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams.

This document should not be presented as a reason to buy or not buy any particular token. The Expelee team disclaims any liability for the resulting losses.