VIBRANIUM AUDITS

Security Assessment

The Great Escape

Verified by Vibranium Audits on 17 February 2024

Vibranium Audits Verified on February 17th, 2024

The Great Escape

The security assessment was prepared by Vibranium Audits.

Executive Summary

TYPES
GAMEFI

ECOSYSTEM
Ethereum/EVM

METHODS
Manual Review, penetration testing and Static Analysis

LANGUAGE
Solidity

TIMELINE
Delivered on 17/02/2023

KEY COMPONENTS
IERC721/IERC20

CODEBASE
N/A

COMMITS
N/A

Vulnerability Summary

16	0	0	0	6	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	Unresolved

0 Critical			Critical vulnerabilities are usually straightforward to exploit and can lead to the loss of user funds or contract state manipulation by external or internal actors.
1 High	0 Resolved		High vulnerabilities are usually harder to exploit, requiring specific conditions, or have a more limited scope, but can still lead to the loss of user funds or contract state manipulation by external or internal actors.
1 Medium	0 Resolved		Medium vulnerabilities are usually limited to state manipulations, but cannot lead to assets loss. Major deviations from best practices are also in this category.
2 Low	0 Resolved		Low vulnerabilities are related to outdated and unused code or minor gas optimization. These issues won't have a significant impact on code execution, but affect the code quality.
1 Informational	0 Resolved		Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | THE GREAT ESCAPE

Summary

- Executive Summary
- Vulnerability Summary
- Codebase
- Audit Scope
- Approach & Methods

Findings

- RVA-01/TVA-01 Centralized Ownership
- RVA-02/TVA-02 Faulty Design in WIthdraw functions
- RVA-03/TVA-03 Floating Pragma
- RVA-04/TVA-04 Use of Old Solidity Version
- RVA-05/TVA-05 Unchecked Return Values
- RVA-06/TVA-06 Lack of Event Emitting
- RVA-07/TVA-07 Gas Optimizations

Disclaimer

CODEBASE | THE GREAT ESCAPE

Repository

N/A

Commits

N/A

AUDIT SCOPE | THE GREAT ESCAPE

2 files audited ● 2 file with Acknowledged findings ● 0 files with Resolved findings

ID	Files	Commit Hash
● RVA	rewards.sol (rewardsTGE)	N/A
● TVA	tgev6.sol (TheGreatEscape)	N/A

APPROACH & METHODS | THE GREAT ESCAPE

This report has been prepared for THE GREAT ESCAPE(2024) to discover issues and vulnerabilities in the source code of the THE GREAT ESCAPE project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review, rigorous Penetration Testing and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Pen-Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the code base to ensure compliance with current best practices and industry standards
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire code base by industry experts.

The security assessment resulted in findings that ranged from critical to informational We recommend addressing these findings to ensure a high level of security standards and industry practices.
We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors.
- Enhance general coding practices for better structures of source codes.
- Review unit tests to cover the possible use cases.
- Review functions for readability, especially for future development work.

FINDINGS | THE GREAT ESCAPE

16	1	1	2	1
Total Findings	High	Medium	Low	Informational

This report has been prepared to discover issues and vulnerabilities for THE GREAT ESCAPE. Through this audit, we have uncovered 11 issues ranging from different severity levels. Utilizing the techniques of Manual Review, Penetration Testing & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

ID	Title	Category	Severity	Status
RVA-01 TVA-01	Centralized Ownership	Design Issue	Minor	Acknowledged
RVA-02 TVA-02	Faulty Design in WIthdraw functions	Logical Issue	Medium	Acknowledged
RVA-03 TVA-03	Floating Pragma	Logical Issue	Minor	Acknowledged
RVA-04 TVA-04	Use of old solidity version	Logical Issue	Medium	Acknowledged
RVA-05 TVA-05	Unchecked return values	Logical Issue	High	Acknowledged
RVA-06 TVA-06	Lack of Event emitting	Coding Style	Informational	Acknowledged
RVA-07 TVA-07	Gas Optimizations	Coding Style	Informational	Acknowledged

RVA-01 TVA-01 | Centralized Ownership

Category	Severity	Location	Status
Design Issue	Minor	tgev6.sol rewards.sol	Acknowledged

Description

Both reviewed Smart Contracts rely on Openzeppelin's 'Ownable.sol' to manage ownership of the contracts. Although no particular vulnerability is related to said used smart contract, having a single address gain ownership over the entire architecture could lead to severe issues outside of the project's or developers' control such as:

- Loss of the Owner address.
- Owner address private key gets compromised by external party.

Recommendation

- Implement a multi-sig address as owner of the smart contracts, thus requiring multiple confirmations before executing one of the key and critical functionalities.
- OR implement a multi-owner structure (similar to Access Control) where multiple address on the smart contracts, thus preventing the complete loss of ownership if one owner address is lost.

RVA-02 TVA-02 | Faulty Design in withdraw fucntions

Categor y	Severity	Location	Status
Logical Issue	Medium	sersh-vesting.sol sersh-vesting-escrow.sol	Acknowledged

Description

Both reviewed Smart Contracts implement critical Withdrawal functionalities either for emergency purposes or for profit withdrawal:

- **function emergencyWithdraw (address _token, uint256 _amount)** -> rewards.sol
- **function adminWithdraw (address _token, uint256 _amount)** -> tgev6.sol

These functions allow the owner/admin to input an ERC20 Token Address and the respective amount needed for withdrawal:

```
function emergencyWithdraw (address _token, uint256 _amount) public payable onlyOwner {
    1 reference
    address payable payable_addr = payable(msg.sender);
        if (_token == address(0x0)) {
        payable_addr.transfer(_amount);
        return;
        }
        1 reference
        IERC20 token = IERC20(_token);
        token.transfer(msg.sender, _amount);
}
```

Multiple issues present:

- Function is marked as payable, while there is no purpose to transfer ETH through this function, thus possibly leading to unintended actions leading to unnecessary gas usage and losses.
- By default, tokens are of 18 Decimals, meaning for the parameter **_amount**, if owner wants to withdraw 10 Tokens, function takes it as 10 * 10 ** 18, while for tokens that have 9 decimals, it would need to take 10 * 10 ** 9, thus possibly causing issues and ambiguities in terms of the withdrawn amounts.

Recommended

There are multiple ways to fix this issue, the most efficient and fitting for the purpose of these functionalities would be to abandon the parameter **_amount** altogether, and have the function withdraw whatever is the amount present within the contract altogether, an example of implementation:

```
function emergencyWithdraw (address _token) external onlyOwner {
    1 reference
    address payable payable_addr = payable(msg.sender);
    4 references
    uint256 amount = 0;
        if (_token == address(0x0)) {
        amount = address(this).balance;
        payable_addr.transfer(amount);
        return;
        }
        2 references
        IERC20 token = IERC20(_token);
        amount = token.balanceOf(address(this));
        token.transfer(msg.sender, amount);
}
```

RVA-03 TVA-03 | Floating Pragma

Category	Severity	Location	Status
Logical Issue	Minor	rewards.sol tgev6.sol	Acknowledged

Description

Both reviewed Smart Contracts rely on a floating solidity version. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Recommendation

Use a new specific stable Solidity version e.g 0.8.20, also avoid using the latest version avoiding any potential yet undiscovered bugs.

RVA-04 TVA-04 | Use of Old solidity versions

Category	Severity	Location	Status
Logical Issue	Medium	rewards.sol tgev6.sol	Acknowledged

Description

Both reviewed Smart Contracts use somewhat old versions of Solidity. Old versions of solidity contain bugs/issues that were fixed later on newer versions.
Although this is normally a low-level vulnerability, it has caused the confirmed existence of a potential vulnerability within the rewards.sol smart contract:
Calldata tuple reencoding -> function depositRewards (..)

The bug would cause malformed output of the ABI encoder caused by overwriting some of the encoded values. The detector reports concerned ABI-encoding expressions if a fault compiler version may be used. Functions that when externally called may trigger the bug are also reported. In such cases, the compiler version is not checked since the bug is triggered by the caller.

Recommended

This vulnerability was fixed in version 0.8.16, thus going back to the point of importance of using newer stable versions of solidity such as 0.8.20.

RVA-05 TVA-05 | Unchecked return values

Categor y	Severity	Location	Status
Logical Issue	High	rewards.sol tgev6.sol	Acknowledged

Description

Both revuewed Smart Contracts implement ERC20 fir token withdrawals either by users or admins within external functions. ERC20 functions like transfer(), transferFrom() etc.. do not implement return value checks leading to;

- if insufficient tokens are present, no revert occurs but a result of "false" is returned.
- if the transferFrom() returns false, it would continue the call to withdraw token from the contract and send it to the caller. Thus a user could withdraw free tokens, and eventually some users will be unable to withdraw their tokens.

Reference: https://github.com/code-423n4/2022-06-nested-findings/issues/8

Recommended

Implement SafeERC20 for all ERC20 actions within the function calls in both smart contracts. (i.e safeTransfer(), safeTransferFrom() etc..)

RVA-06 TVA-06 | Lack of event emitting

Categor y	Severity	Location	Status
Logical Issue	Informational	rewards.sol tgev6.sol	Acknowledged

Description

Both reviewed Smart Contracts contain key Admin/Owner functionalities that lack the emittance of events. There is no specific vulnerability associated with this matter.
It just would be better to add events to these functionalities for greater transparency with the community in Transactions and for future reference.

RVA-07 TVA-07 | Gas Optimizations

Categor y	Severity	Location	Status
Logical Issue	Informational	rewards.sol tgev6.sol	Acknowledged

Description

Some core functionalities, contain multiple if statements that return an error message in case certain conditions aren't met.
Due to the number of these statements and the length of the error messages, it will be the source of highly augmented gas fees, and potentially blocking the execution of the transaction in cases of Network congestion due to Gas Limit exceeding.

Recommended

We Recommend the use of custom errors instead of the full length error messages to save up on gas usage; example:

```
error undefinedAddress();
...
if (buyer == address(0)) {
        revert undefinedAddress();
}
...
```

DISCLAIMER | VIBRANIUM AUDITS

This report is subject to the terms and conditions (including without limitation description of services confidentiality disclaimer and limitation of liability) set forth in the Services Agreement or the scope of services and terms and conditions provided to you (“Customer” or the “Company”) in connection with the Agreement This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement This report may not be transmitted disclosed referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Vibranium Audits prior written consent in each instance

This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team This report is not nor should be considered an indication of the economics or value of any “product” or “asset”created by any team or project that contracts Vibranium Audits to perform a security assessment This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed nor do they provide any indication of the technologies proprietors business business model or legal compliance

This report should not be used in any way to make decisions around investment or involvement with any particular project This report in no way provides investment advice nor should be leveraged as investment advice of any sort This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology

Blockchain technology and cryptographic assets present a high level of ongoing risk Vibranium Audits position is that each company and individual are responsible for their own due diligence and continuous security Vibranium Audits goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze

The assessment services provided by Vibranium Audits is subject to dependencies and under continuing development You Agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty The assessment reports could include false positives false negatives and other unpredictable results The services may access and depend upon multiple layers of third-parties

ALL SERVICES THE LABELS THE ASSESSMENT REPORT WORK PRODUCT OR OTHER MATERIALS OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED “AS IS” AND “AS AVAILABLE” AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW VIBRANIUM AUDITS HEREBY DISCLAIMS ALL WARRANTIES WHETHER EXPRESS IMPLIED STATUTORY OR OTHERWISE WITH RESPECT TO THE SERVICES ASSESSMENT REPORT OR OTHER MATERIALS WITHOUT LIMITING THE FOREGOING VIBRANIUM AUDITS SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY FITNESS FOR A PARTICULAR PURPOSE TITLE AND NON-INFRINGEMENT AND ALL WARRANTIES ARISING FROM COURSE OF DEALING USAGE OR TRADE PRACTICE WITHOUT LIMITING THE FOREGOING VIBRANIUM AUDITS MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES THE LABELS THE ASSESSMENT REPORT WORK PRODUCT OR OTHER MATERIALS OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF WILL MEET CUSTOMER’S OR ANOTHER PERSON’S REQUIREMENTS ACHIEVE ANY INTENDED RESULT BE COMPATIBLE OR WORK WITH ANY SOFTWARE SYSTEM OR OTHER SERVICES OR BE SECURE ACCURATE COMPLETE FREE OF HARMFUL CODE

OR ERROR-FREE WITHOUT LIMITATION TO THE FORGOING, VIBRANIUM AUDITS PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KINDTHAT WILL MEET CUSTOMER' S REQUIREMENTS ACHIEVE ANY INTENDED RESULTS BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE APPLICATIONS SYSTEMS OR SERVICES OPERATE WITHOUT INTERRUPTION MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED

WITHOUT LIMITING THE FOREGOING NEITHER VIBRANIUM AUDITS NOR ANY OF VIBRANIUM AUDITS AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND EXPRESS OR IMPLIED AS TO THE ACCURACY RELIABILITY ORCURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE VIBRANIUM AUDITS WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS MISTAKES OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE OF ANY NATURE WHATSOEVER RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES ASSESSMENT REPORT OR OTHER MATERIALS

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNEROR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS

THE SERVICES ASSESSMENT REPORT AND ANY OTHER MATERIALS HERE UNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT NOR MAY COPIES BE DELIVERED TO ANY OTHER PERSON WITHOUT VIBRANIUM AUDITS PRIOR WRITTEN CONSENT IN EACH INSTANCE

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES ASSESSMENT REPORT AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST VIBRANIUM AUDITS WITH RESPECT TO SUCH SERVICES ASSESSMENT REPORT AND ANY ACCOMPANYING MATERIALS

THE REPRESENTATIONS AND WARRANTIES OF VIBRANIUM AUDITS CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER ACCORDINGLY NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST VIBRANIUM AUDITS WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE

FOR AVOIDANCE OF DOUBT THE SERVICES INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS ORMATERIALS SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

Vibranium | Securing the Web3 World

Vibranium Audits is a blockchain security company that was founded in 2021 by professors from the University of Greenwich and cyber-security engineers from ITI Capital. As pioneers in the field, Vibranium Audits utilizes best-in-class Formal Verification and AI technology to secure and monitor blockchains, smart contracts, and Web3 apps.