Guard

Smart contracts security assessment

Final report

Tariff: Standard

Venomous.finance

June 2022

0xguard.com

hello@0xguard.com

Contents

Introduction

This report has been prepared for the Venomous.finance team upon their request.

The audited project is a fork of the Tomb Finance Project.

Further details about Venomous.finance are available at the official website: https://venomous.finance.

Update: The Venomous.finance team, after receiving the initial audit, decided to correct the comments and redeploy the contracts.

Name	Venomous.finance
Audit date	2022-06-06 - 2022-06-07
Language	Solidity
Platform	Avalanche Network

Contracts checked

Name	Address
VenomHostTomb	0x54E90234257F58075C3dA580AB4f02E30A5a2D62
VenomBond	0x45Ba355F4fDE24B143d32AED1B780D4d30629399
SymbiotShare	0x98716351b2660F8Ebbe4bAfB34A07f9c4aA35E14
VShareRewardPool	0x521cEa929C0c6935778d59FE22FEdBd8f2779F03
VTombGenesisRewardPool	0xa383a16f62b6c01703Cd5b06D8348f39081A8045
OracleV2	0xC14c3224BA7316D540fcC7E99E2138a096AB4cCE
Boardroom	0xdd075243e6E88e0Ab2ef147F3C0cd25B1798A177
Treasury	0x386510ec3912E68A0b63cda06B28aaA35a4a5eAf
multisigContractCaller	0x169b08d74afc5ea7639d0b40260e5336bee16ae8
Multiple contracts	

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Comparing the project to the Tomb Finance implementation

Classification of issue severity

High severity	High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.
Medium severity	Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.
Low severity	Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

1. Tax bypass (FIXED) (VenomHostTomb)

Tax avoidance in the Tomb project is the main problem the team faced. The problem is that there is an invariant in the `transferFrom()` function that deducts tax for the transfer of tokens, but there is also an invariant without deduction of tax that calls the `transfer()` function. Using this problem, you can bypass all tax deductions if you use only the `transfer()` function and it is possible to violate the tokenomics of the project.

Recommendation: It is recommended to overload the `transfer()` function to work with a tax or completely remove the tax functionality in contracts.

2. Locked funds (FIXED) (VTombGenesisRewardPool)

If the called `deposit()` function for `pool.token` is equal to the `cake` address, then in 286L the percentage calculated for the fee can never be withdrawn due to the lack of special functions for this. Also, in the `setCakeTokenFee()` function, it is indicated that the maximum fee is 20%, which is too high a percentage for it to be locked on the contract forever. It is also not clear why a fee for depositing a `cake` token is taken if the `cake` token is not a commission token (JoeToken - 0x6e84a6216eA6dACC71eE8E6b0a5B7322EEbC0fDd).

Recommendation: It is recommended to disable taking a fee when depositing a `cake` token.

Medium severity issues

1. Contract ownership (FIXED) (Multiple contracts)

1) An Operator can change `taxTiersTwaps` and `taxTiersRate` up to 100% in `VenomHostTomb` token in `setTaxTiersTwap()` and `setTaxTiersRate()` functions.

2) The `governanceRecoverUnsupported()` function (found in the VenomHostTomb, SymbiotShare and VShareRewardPool contracts) can remove all tokens from the contract balance if

the operator role is compromised.

Recommendation: There is a large number of functions with the `onlyOperator()` modifier, there is a possibility that the operator can be compromised. It is recommended to create multiple roles for different kinds of functions to reduce the operator's problem. It is also recommended to add a time delay to the especially important set functions using the TimelockController. We also recommend that you look through the entire codebase to find functions that are dangerous for you as the owner of the project (mainly set functions), if there are any, then add a call to them via a multisig wallet. This will help avoid the issue of owner compromise.

Low severity issues

1. Contract version not from a verified source (multisigContractCaller)

An implemented contract for multisig signatures can be found at OpenZeppelin. This contract is well tested and has various variations of work (2 out of 3, 3 out of 5, etc.). It is recommended to use this particular contract in order to avoid possible compromises of accounts.

Conclusion

2 high, 1 medium, 1 low severity issues were found.

2 high and 1 medium severity issues have been resolved in the update.

The Venomous.finance Project was compared with the Tomb Project. Venomous.finance has changed the implementation of Token and Pool contracts. All contracts have been updated to the latest version of Solidity. Also, in most contracts, the implementation of the `isHuman()` modifier has been added, which checks the sender of the transaction and does not prohibit other contracts from calling functions, this action will protect against flashloan attacks.

Disclaimer

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 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 without 0xGuard prior written consent.

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 0xGuard 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.

0x Guard