0x Guard

Smart contracts security assessment

Final report

Tariff: Standard

Champion Finance UVIC

September 2022

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for Champion Finance.

The Champion Finance Protocol allows users to farm UVICTokens. The UVICToken is a rebase token. The EVICToken owner (taxOffice) can set a fee on token trading or set a limit on the amount of the sale of the token.

Contracts UVICTreasury and UVICBoardroom allow keeping a stable price of the UVICToken using the rebase mechanism.

The code is available at the GitHub repository and was audited after the commit 6d309c479c474582a0dff378d34dce318dc69882.

Report Update.

The contract's code was updated according to this report and rechecked after the commit b121ac0ea90d9dd762710bf1846e9a8dc3949c89.

Only 3 contracts with their dependencies were audited: UVICToken, UVICBoardroom, UVICTreasury.

Name	Champion Finance UVIC
Audit date	2022-09-26 - 2022-09-28
Language	Solidity
Platform	Avalanche Network

Contracts checked

Name	Address
UVICToken	
UVICBoardroom	
UVICTreasury	

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

- Manually analyze smart contracts for security vulnerabilities
- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
Unencrypted Private Data On-Chain	passed
Code With No Effects	passed
Message call with hardcoded gas amount	passed
Typographical Error	passed
DoS With Block Gas Limit	passed
Presence of unused variables	passed
Incorrect Inheritance Order	passed
Requirement Violation	passed
Weak Sources of Randomness from Chain Attributes	passed
Shadowing State Variables	passed

Incorrect Constructor Name	passed
Block values as a proxy for time	passed
Authorization through tx.origin	passed
DoS with Failed Call	passed
Delegatecall to Untrusted Callee	passed
Use of Deprecated Solidity Functions	passed
Assert Violation	passed
State Variable Default Visibility	passed
Reentrancy	passed
Unprotected SELFDESTRUCT Instruction	passed
Unprotected Ether Withdrawal	passed
Unchecked Call Return Value	passed
Floating Pragma	passed
Outdated Compiler Version	passed
Integer Overflow and Underflow	passed
Function Default Visibility	passed

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

No issues were found

Medium severity issues

No issues were found

Low severity issues

1. Gas optimization (UVICToken)
Status: Open

1. The visibility of the `setMarketLpPairs()` function can be changed to external to save gas.

2. The `require` check on L328 is redundant because there is no way to set `polWallet` to zero-address.

2. Incorrect commets (UVICTreasury)
Status: Fixed

There are incorrect comments with '`WETH`' token symbol on L188, L189, L415, L416.

3. Payouts to funds 100% rewards (UVICTreasury)
Status: Open

The contract operator can set a 50% reward amount for each fund: `daoFundSharedPercent` and `polFundSharedPercent` (**in total 100%**). This means that all rewards will go to these funds with nothing left for all boardroom contracts.

```
    function _expansionBoardroom(uint256 _tokenPrice) internal {
```

```
        ...
            uint256 _daoFundReward =
boardroomReward.mul(daoFundSharedPercent).div(100);
            uint256 _polReward = boardroomReward.mul(polFundSharedPercent).div(100);
            uint256 _boardroomReward =
boardroomReward.sub(_daoFundReward).sub(_polReward);

            daoFundReward = daoFundReward.add(_daoFundReward);
            polReward = polReward.add(_polReward);

            mainTokenErc20.mint(address(this), _boardroomReward);
            IERC20(mainToken).safeApprove(boardroomAddress, 0);
            IERC20(mainToken).safeApprove(boardroomAddress, _boardroomReward);
            boardroom.allocateSeigniorage(_boardroomReward);
            ...
        }
```

Recommendation: Consider lowering the total payment to the funds.

Conclusion

Champion Finance UVIC UVICToken, UVICBoardroom, UVICTreasury contracts were audited. 3 low severity issues were found.
1 low severity issue has been fixed in the update.

According to this report 1 low issue was fixed by the Champion Finance team.

We strongly recommend writing unit tests to have extensive coverage of the codebase minimize the possibility of bugs and ensure that everything works as expected.

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