



Smart contracts security assessment

Final report

[Tariff: Standard](#)

Arbius V4

August 2024



0xguard.com



hello@0xguard.com

Contents

1. Introduction	3
2. Contracts checked	4
3. Procedure	4
4. Known vulnerabilities checked	4
5. Classification of issue severity	5
6. Issues	6
7. Conclusion	8
8. Disclaimer	9

Introduction

The report has been prepared for **Arbius V4**.

This is an incremental audit of the Arbius contracts, including NFT staking and voting governance.

The V2_EngineV4 is an upgradable contract to store models, tasks, and solutions. It also administers rewards, fees, and solution's contestations.

The GovernorV1 is a governance contract inheriting OpenZeppelin's [Governor](#), [GovernorSettings](#), [GovernorCompatibilityBravo](#), and [GovernorTimelockControl](#), can't be upgraded.

The VeGovernorVotes and VeGovernorVotesQuorumFraction contracts are forked from OpenZeppelin's [GovernorVotes](#) and [GovernorVotesQuorumFraction](#) contracts without modifications.

The VotingEscrow contracts is an ERC721 non-fungible token forked from Velodrome Finance [VotingEscrow](#).

The VeStaking contract is staking contract for VotingEscrow NFTs with external rewards from V2_EngineV4 contract.

The code is available at the GitHub [repository](#) and was audited after the commit [712c7621cd478ede2369c135fb4bc3d435ae684a](#).

Name	Arbius V4
Audit date	2024-07-25 - 2024-08-04
Language	Solidity
Platform	Arbitrum Network

🛡️ Contracts checked

Name	Address
------	---------

VotingEscrow

VeGovernorVotes

🛡️ 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
<u>Unencrypted Private Data On-Chain</u>	passed
<u>Code With No Effects</u>	passed
<u>Message call with hardcoded gas amount</u>	passed
<u>Typographical Error</u>	passed
<u>DoS With Block Gas Limit</u>	passed
<u>Presence of unused variables</u>	passed

<u>Incorrect Inheritance Order</u>	passed
<u>Requirement Violation</u>	passed
<u>Weak Sources of Randomness from Chain Attributes</u>	passed
<u>Shadowing State Variables</u>	passed
<u>Incorrect Constructor Name</u>	passed
<u>Block values as a proxy for time</u>	passed
<u>Authorization through tx.origin</u>	passed
<u>DoS with Failed Call</u>	passed
<u>Delegatecall to Untrusted Callee</u>	passed
<u>Use of Deprecated Solidity Functions</u>	passed
<u>Assert Violation</u>	passed
<u>State Variable Default Visibility</u>	passed
<u>Reentrancy</u>	passed
<u>Unprotected SELFDESTRUCT Instruction</u>	passed
<u>Unprotected Ether Withdrawal</u>	passed
<u>Unchecked Call Return Value</u>	passed
<u>Floating Pragma</u>	passed
<u>Outdated Compiler Version</u>	passed
<u>Integer Overflow and Underflow</u>	passed
<u>Function Default Visibility</u>	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. Useless deposit types (VotingEscrow)

Status: Open

VotingEscrow NFT contract locks ERC20 tokens in governance ERC721 NFTs. Deposit can be done in different forms:

```
enum DepositType {
    DEPOSIT_FOR_TYPE,
    CREATE_LOCK_TYPE,
    INCREASE_LOCK_AMOUNT,
    INCREASE_UNLOCK_TIME
}
```

DEPOSIT_FOR_TYPE and **INCREASE_LOCK_AMOUNT** modes share the same code but **increase_amount** function can be called only by NFT owner authorized address.

2. Math underflow (VotingEscrow)

Status: Open

_deposit_for function and all related external functions may experience underflow failures due to incorrect unsigned type of **balanceDiff** variable.

```
// get current balance before checkpoint
uint256 balanceOfNFTBefore = _balanceOfNFT(_tokenId, block.timestamp);

// Possibilities:
// Both old_locked.end could be current or expired (>/< block.timestamp)
// value == 0 (extend lock) or value > 0 (add to lock or extend lock)
// _locked.end > block.timestamp (always)
_checkedpoint(_tokenId, old_locked, _locked);

// get current balance after checkpoint and calculate diff
uint256 balanceOfNFTAfter = _balanceOfNFT(_tokenId, block.timestamp);
uint256 balanceDiff = balanceOfNFTAfter - balanceOfNFTBefore;
```

Recommendation: Use **int256** type for **balanceDiff**.

3. Outdated imports (VeGovernorVotes)

Status: Open

The repository uses v4.9 release of OpenZeppelin's contracts. The [Governor](#) contract has been moderately updated and patched in the [v5 release](#).

Conclusion

Arbius V4 VotingEscrow, VeGovernorVotes contracts were audited. 3 low severity issues were found.

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 OxGuard 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 OxGuard 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.

OxGuard retains exclusive publishing rights for the results of this audit on its website and social networks.



 Guard