CERTIK

Security Assessment

The impossible art formula gallery

Nov 3rd, 2021

Table of Contents

Summary

This report has been prepared for The impossible art formula gallery to discover issues and vulnerabilities in the source code of the The impossible art formula gallery project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase 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 codebase 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:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	The impossible art formula gallery
Platform	ethereum
Language	Solidity
Codebase	https://github.com/uni-arts-chain/uniarts-eth-dao/tree/master
Commit	f3683330fa6a00f6b0cf8d3dd652c3c1a8ea700a

Audit Summary

Delivery Date	Nov 03, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

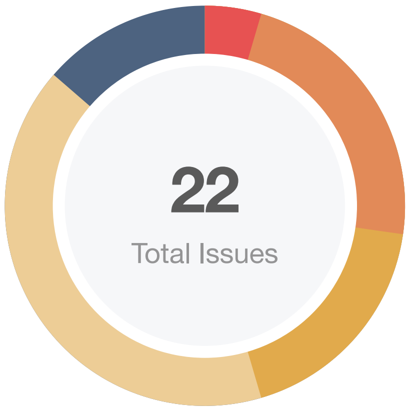
Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	1	0	0	0	0	1
● Major	5	0	0	4	0	1
● Medium	4	0	0	2	0	2
● Minor	9	0	0	3	0	6
● Informational	3	0	0	0	1	2
● Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
AAF	Auction.sol	623636c8e51f5a25a1d659b1733b6bd677e3ed4e4e9430c839cc6fe91174adc4
VMA	VoteMining.sol	cb57f477c405551a303e6a81ba5fdfc56c3951cc0d6166a7e7abdbeb47012db6

It is noted that the group id could be changed in the `addGroup` function by the `operator` account and may cause side-effects on the next operations.

Findings



🟥	**Critical**	**1** (4.55%)
🟧	**Major**	**5** (22.73%)
🟨	**Medium**	**4** (18.18%)
🟨	**Minor**	**9** (40.91%)
🟦	**Informational**	**3** (13.64%)
🟩	**Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
AAF-01	Missing Input Validation	Volatile Code	● Informational	✓ Resolved
AAF-02	Lack of Input Validation	Volatile Code	● Minor	✓ Resolved
AAF-03	Missing Emit Events	Coding Style	● Informational	✓ Resolved
AAF-04	Wrong Event to Emit	Logical Issue	● Minor	ⓘ Acknowledged
VMA-01	Unchecked Value of ERC-20 `transfer()`/`transferFrom()` Call	Volatile Code	● Minor	✓ Resolved
VMA-02	Unstake Fails After `currentGroupId` Changed	Logical Issue	● Major	ⓘ Acknowledged
VMA-03	Exclude Tokens When Rescuing	Logical Issue	● Medium	ⓘ Acknowledged
VMA-04	Potential Reentrancy Attack	Logical Issue	● Minor	✓ Resolved
VMA-05	Redeem Tokens Repeatedly	Logical Issue	● Critical	✓ Resolved
VMA-06	Lack of Input Validation	Volatile Code	● Minor	✓ Resolved
VMA-07	Centralization Risk	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
VMA-08	Centralization Risk	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
VMA-09	Centralization Risk	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged

ID	Title	Category	Severity	Status
VMA-10	Missing Input Validation	Volatile Code	● Minor	Resolved
VMA-11	Missing Emit Events	Coding Style	● Informational	Partially Resolved
VMA-12	`addNFT` Function Issue	Logical Issue	● Medium	Resolved
VMA-13	`claimMintRewards` Function Issue	Logical Issue	● Minor	Acknowledged
VMA-14	Mismatch Between Code and Comment	Volatile Code	● Minor	Resolved
VMA-15	Unknown implementations	Volatile Code	● Minor	Acknowledged
VMA-16	Redeem Tokens Without Subtracting Votes	Logical Issue	● Major	Resolved
VMA-17	Over-write Value of `userNFTVotes`	Logical Issue	● Medium	Resolved
VMA-18	Function `migrate`	Volatile Code	● Medium	Acknowledged

AAF-01 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	● Informational	Auction.sol: 92~94	✓ Resolved

Description

The given input `usdtContractAddress` is missing the check for the non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected error as below:

```
93  require(usdtContractAddress != address(0), "usdtContractAddress should not be address(0)");
```

Alleviation

The development team heeded our advice and resolved this issue in commit aabe2921e8b3539c3a523017b596da1223381d8b.

AAF-02 | Lack of Input Validation

Category	Severity	Location	Status
Volatile Code	● Minor	Auction.sol: 148~150	✓ Resolved

Description

An NFT can not be added to a finished `Match`.

Recommendation

Consider adding check for the input parameter to prevent unexpected error as below:

```
require(matches[matchId].expiryBlock >= block.number, "the match is finished");
```

Alleviation

The development team heeded our advice and resolved this issue in commit aabe2921e8b3539c3a523017b596da1223381d8b.

AAF-03 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	● Informational	Auction.sol: 314, 283, 262~263	✓ Resolved

Description

The following functions that affect the status of sensitive variables should be able to emit events as notifications to contract users.

- `player_withdraw_bid(string memory matchId, uint tokenIndex)`
- `process_withdraw_nft(string memory matchId, uint tokenIndex)`
- `creator_withdraw_profit()`

Recommendation

Consider adding events for sensitive actions, and emit them in the functions.

Alleviation

The development team heeded our advice and resolved this issue in commit aabe2921e8b3539c3a523017b596da1223381d8b.

AAF-04 | Wrong Event to Emit

Category	Severity	Location	Status
Logical Issue	● Minor	Auction.sol: 176	ⓘ Acknowledged

Description

The event to emit in the `addAuctionNFT` function should not be the `CreateAuctionEvent` event.

Recommendation

Consider emitting the correct event.

Alleviation

No alleviation.

VMA-01 | Unchecked Value of ERC-20 `transfer()`/`transferFrom()` Call

Category	Severity	Location	Status
Volatile Code	● Minor	VoteMining.sol: 459, 454, 437, 417, 387	✓ Resolved

Description

The linked `transfer()`/`transferFrom()` invocations do not check the return value of the function call which should yield a `true` result in case of proper ERC-20 implementation.

Recommendation

"As many tokens do not follow the ERC-20 standard faithfully, they may not return a `bool` variable in this function's execution meaning that simply expecting it can cause incompatibility with these types of tokens. Instead, we advise that OpenZeppelin's `SafeERC20.sol` implementation is utilized for interacting with the `transfer()` and `transferFrom()` functions of ERC-20 tokens. The OZ implementation optionally checks for a return value rendering compatible with all ERC-20 token implementations.

Alleviation

The development team heeded our advice and resolved this issue in commit 566b3f7aaefe52dd080f9ae7f4e21ad98e6d6ed5.

VMA-02 | Unstake Fails After `currentGroupId` Changed

Category	Severity	Location	Status
Logical Issue	● Major	VoteMining.sol: 419, 388	ⓘ Acknowledged

Description

The operator is able to change the value of `currentGroupId` by calling the `addGroup` function. A user may stake tokens with a group id 1, then the operator changed the group id to 2, finally the user fails to unstake tokens from the group whose id is 2.

Alleviation

`The impossible art formula gallery`: Group is added by operator. And check with `require(currentGroupId == 0 || groups[currentGroupId].add(VOTE_DURATION) <= block.timestamp, "Previous group is not over.");`

VMA-03 | Exclude Tokens When Rescuing

Category	Severity	Location	Status
Logical Issue	● Medium	VoteMining.sol: 459	ⓘ Acknowledged

Description

The tokens in the `voteTokens` array should be excluded when rescuing tokens.

Alleviation

No alleviation.

VMA-04 | Potential Reentrancy Attack

Category	Severity	Location	Status
Logical Issue	● Minor	VoteMining.sol: 404	✓ Resolved

Description

A reentrancy attack can occur when the contract creates a function that makes an external call to another untrusted contract before resolving any effects. If the attacker can control the untrusted contract, they can make a recursive call back to the original function, repeating interactions that would have otherwise not run after the external call resolved the effects.

Recommendation

We recommend using the Checks-Effects-Interactions Pattern to avoid the risk of calling unknown contracts or applying OpenZeppelin ReentrancyGuard library - `nonReentrant` modifier for the aforementioned functions to prevent reentrancy attack.

Alleviation

The development team heeded our advice and resolved this issue in commit 566b3f7aaefe52dd080f9ae7f4e21ad98e6d6ed5.

VMA-05 | Redeem Tokens Repeatedly

Category	Severity	Location	Status
Logical Issue	● Critical	VoteMining.sol: 433, 451	✓ Resolved

Description

A user is able to redeem tokens repeatedly until tokens of the `VoteMining` contract are exhausted by the `redeemToken` function and `redeemToken` function. Finally, it will block other operations.

Alleviation

The development team heeded our advice and resolved this issue in commit 6690f2c426449c07cf85706d9f09787e8378e23f

VMA-06 | Lack of Input Validation

Category	Severity	Location	Status
Volatile Code	● Minor	VoteMining.sol: 246	✓ Resolved

Description

The operator can not add an `NFT` to a finished group.

Recommendation

Consider adding check for the input parameter to prevent unexpected errors as below:

```
require(!hasFinished[groupId], "the target group is finished");
```

Alleviation

The development team heeded our advice and resolved this issue in commit 566b3f7aaefe52dd080f9ae7f4e21ad98e6d6ed5.

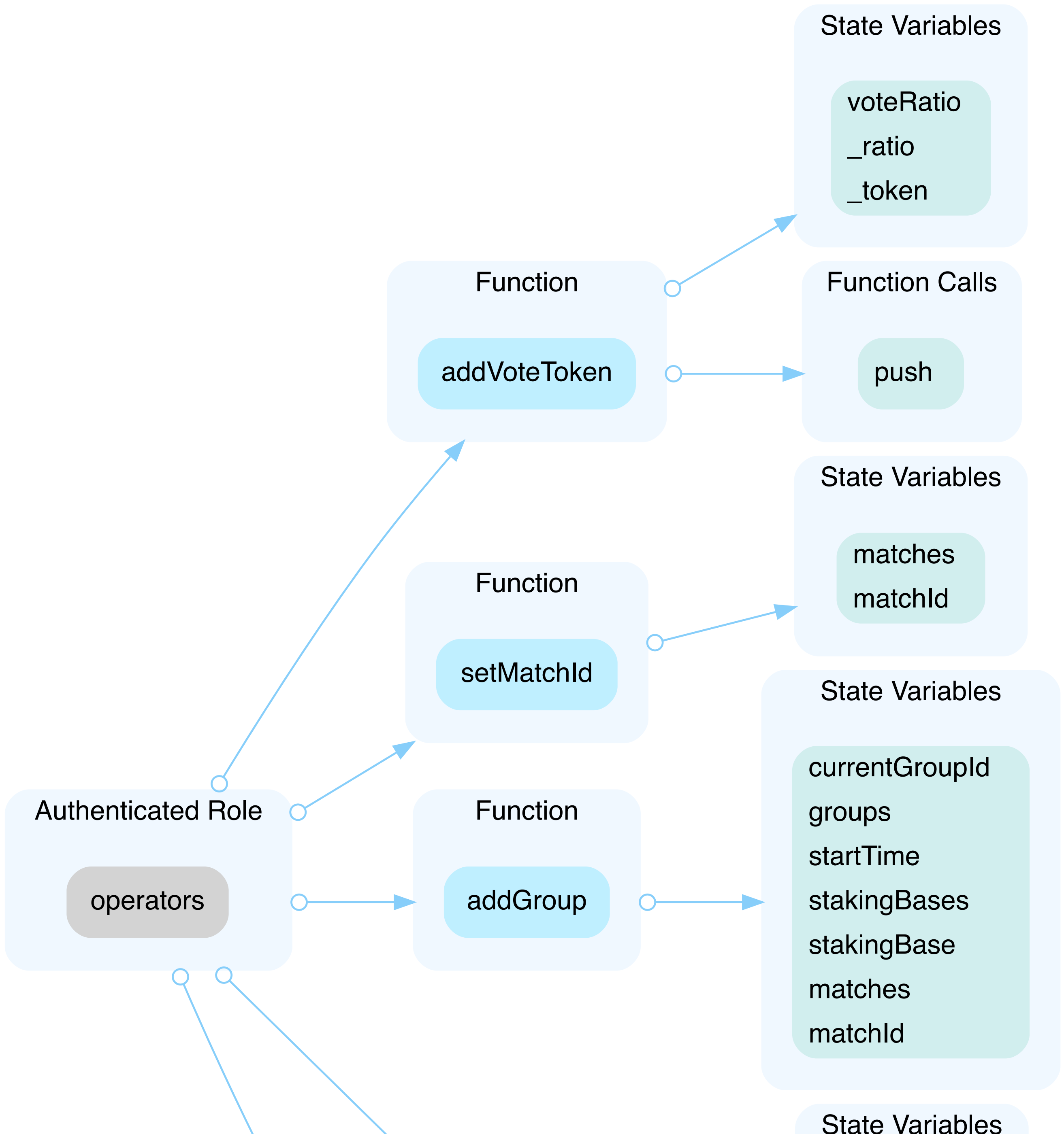
VMA-07 | Centralization Risk

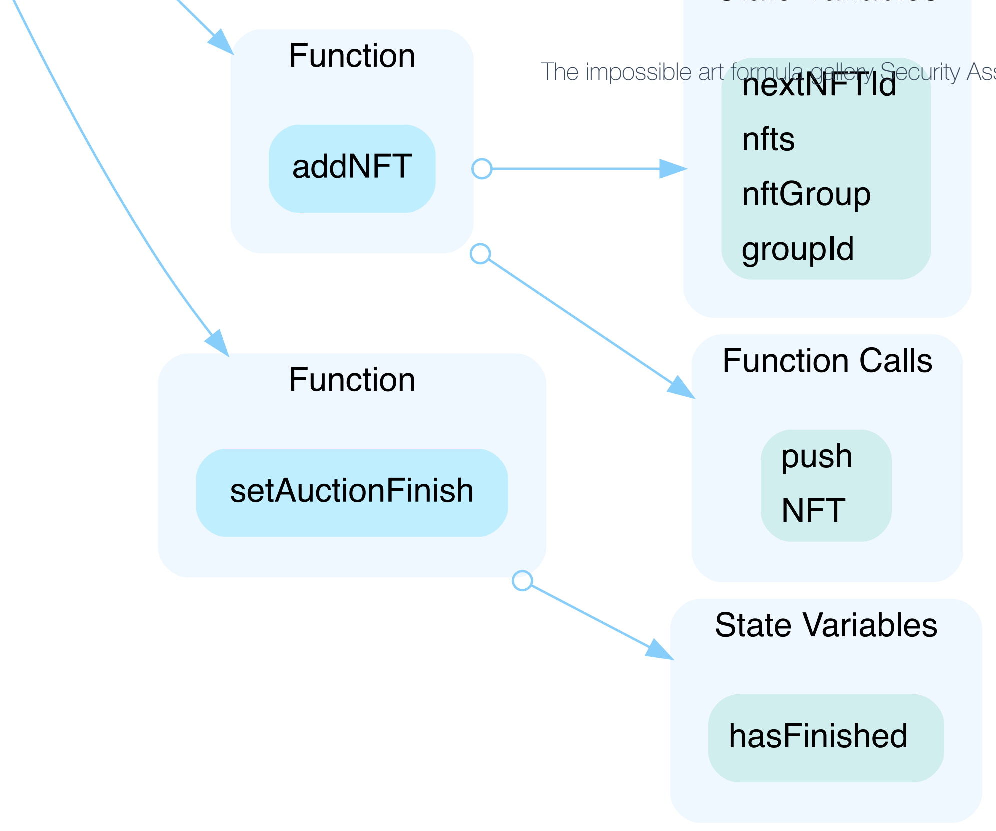
Category	Severity	Location	Status
Centralization / Privilege	● Major	VoteMining.sol: 203~211, 233~235, 237~244, 246~263, 635 ~637	ⓘ Acknowledged

Description

In the contract, `VoteMining`, the role, `operators`, has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `operators` may allow the hacker to take advantage of this.





Recommendation

We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked.

In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

No alleviation.

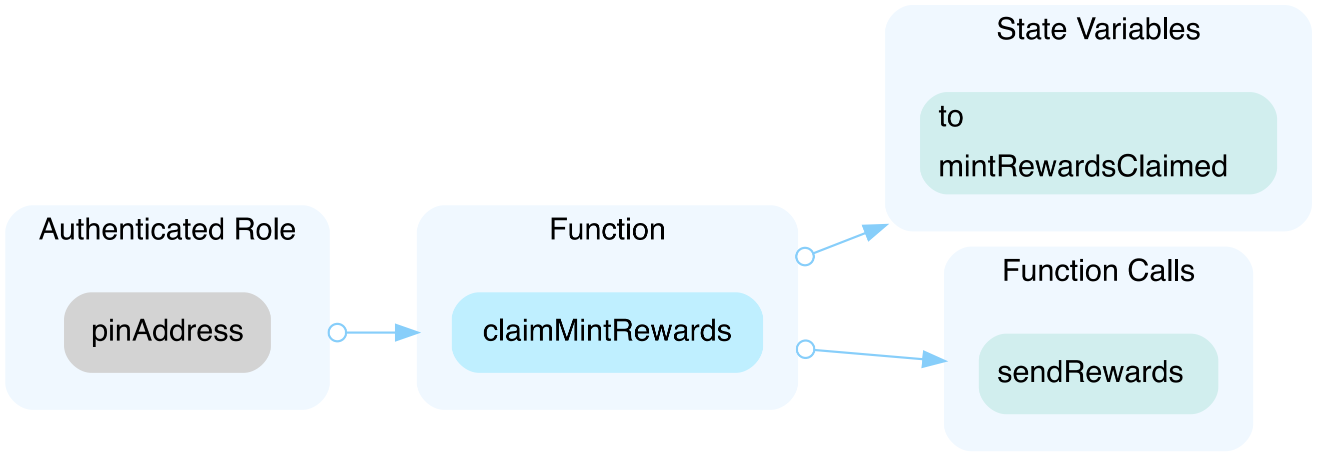
VMA-08 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	VoteMining.sol: 604~620	ⓘ Acknowledged

Description

In the contract, `VoteMining`, the role, `pinAddress`, has the authority over the functions shown in the diagram below.

Any compromise to the privileged account which has access to `pinAddress` may allow the hacker to take advantage of this and send minted rewards to any address.



Recommendation

We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked.

In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;

- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

No alleviation.

VMA-09 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	VoteMining.sol: 213~231	ⓘ Acknowledged

Description

In the contract `VoteMining`, the role `owner` has the authority over the functions shown below:

- `setAuctionAddress(address _auction)`
- `setPinAddress(address _pin)`
- `setTokenLockId(uint lockId)`
- `setOperator(address operator, bool isOperator)`
- `setVoteDays(uint _days)`
- `rescueToken(address token, uint amount)`
- `addVoteToken(address _token, uint _ratio)`
- `setMatchId(uint groupId, string calldata matchId)`
- `addGroup(uint stakingBase, uint startTime, string calldata matchId)`
- `addNFT(uint groupId, address[] calldata nftAddrs, uint[] calldata nftIds)`
- `setAuctionFinish(uint groupId)`

Any compromise to the privileged account which has access to `owner` may allow the hacker to take advantage of this and send minted rewards to any address.

Recommendation

We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked.

In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;

- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

No alleviation.

VMA-10 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	● Minor	VoteMining.sol: 194~201	✓ Resolved

Description

The given inputs are missing the checks for the non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected errors as below:

```
require(_treasury != address(0), "_treasury should not be address(0)");
require(_tokenLocker != address(0), "_tokenLocker should not be address(0)");
```

Alleviation

The development team heeded our advice and resolved this issue in commit 566b3f7aaefe52dd080f9ae7f4e21ad98e6d6ed5.

VMA-11 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	● Informational	VoteMining.sol: 635, 604, 548, 524, 404, 377, 246, 237, 229, 221 , 213, 203	Partially Resolved

Description

The functions that affect the status of sensitive variables should be able to emit events as notifications to contract users.

- `addVoteToken(address _token, uint _ratio)`
- `setAuctionAddress(address _auction)`
- `setTokenLockId(uint lockId)`
- `setVoteDays(uint _days)`
- `addGroup(uint stakingBase, uint startTime, string calldata matchId)`
- `addNFT(uint groupId, address[] calldata nftAddrs, uint[] calldata nftIds)`
- `stake(address nftAddr, uint nftId, address token, uint amount)`
- `unstake(address nftAddr, uint nftId, address token, uint amount)`
- `voteBonded(address nftAddr, uint nftId, uint amount)`
- `unvoteBonded(address nftAddr, uint nftId, uint amount)`
- `claimMintRewards(address nftAddr, uint nftId, address to)`
- `setAuctionFinish(uint groupId)`

Recommendation

Consider adding events for sensitive actions, and emit them in the functions.

Alleviation

The development team heeded our advice and partially resolved this issue in commit 566b3f7aaefe52dd080f9ae7f4e21ad98e6d6ed5.

VMA-12 | `addNFT` Function Issue

Category	Severity	Location	Status
Logical Issue	● Medium	VoteMining.sol: 251~253, 270	✓ Resolved

Description

From the aforementioned code, we can infer that an `NFT` can not be added twice and there is no way to add an `NFT` to another group, which will cause an `NFT` to be voted only once. The rest function will be blocked after a new group is added, since the old `NFT` only belongs to the old group, there is no way to add NFT to the new group.

It will also cause a side effect on the `getAuctionPrices` function since the values of `groupNFTs` are wrong for the new group.

Alleviation

The development team responded that NFT is unique in all groups.

VMA-13 | `claimMintRewards` Function Issue

Category	Severity	Location	Status
Logical Issue	● Minor	VoteMining.sol: 604~620	ⓘ Acknowledged

Description

From the aforementioned code, we can infer that the role `Pin` can claim minted rewards from every `NFT` one time at most, no matter how many vote rounds have been taken place.

Recommendation

NFT is unique in all groups

Alleviation

`The impossible art formula gallery`: NFT is unique in all groups.

VMA-14 | Mismatch Between Code and Comment

Category	Severity	Location	Status
Volatile Code	● Minor	VoteMining.sol: 33~34	✓ Resolved

Description

The variable `dailyVoteRewardCap` and `mintRewardCap` is 12.5% of daily cap and 25% of weekly cap, while the comment claim to be 25% of daily cap and 25% of daily.

Recommendation

Consider changing the comments to match the codes.

Alleviation

The development team heeded our advice and resolved this issue in commit 566b3f7aaefe52dd080f9ae7f4e21ad98e6d6ed5.

VMA-15 | Unknown implementations

Category	Severity	Location	Status
Volatile Code	● Minor	VoteMining.sol: 11~22	ⓘ Acknowledged

Description

The implementations of contracts `ITreasury`, `IAuction`, and `ITokenLocker` are unknown.

The implementation of contract `IVoteMiningV1` is also unknown in commit f3683330fa6a00f6b0cf8d3dd652c3c1a8ea700a.

Alleviation

No alleviation.

VMA-16 | Redeem Tokens Without Subtracting Votes

Category	Severity	Location	Status
Logical Issue	● Major	VoteMining.sol: 451~456	✓ Resolved

Description

An Investor can redeem his staked tokens by calling the `redeemToken(address token)` function before the current voting round finishes, nonetheless, the votes of the redeemed tokens are still effective, the user can still get rewards of the votes.

Recommendation

Consider subtracting the votes when the investor redeems staked tokens.

Alleviation

The development team heeded our advice and resolved this issue by deleting the function in commit 6690f2c426449c07cf85706d9f09787e8378e23f.

VMA-17 | Over-write Value of `userNFTVotes`

Category	Severity	Location	Status
Logical Issue	● Medium	VoteMining.sol: 493, 349, 303	✓ Resolved

Description

The value of `userNFTVotes` could be updated by the `_unvote` function and `vote` function in each group, however, its value is used in the `getAuctionRewards` function for calculating the rewards of every group in the `getTotalRewards` function.

The value of `userNFTVotes` will be increased after each group is finished, then will cause some side effects on the `getAuctionRewards` function, `getTotalRewards` function, `getBondedBalance` function, and `unbond` function.

Recommendation

Consider taking the group into account for the `userNFTVotes` variable.

Alleviation

The development team responded that NFT is unique in all groups.

VMA-18 | Function `migrate`

Category	Severity	Location	Status
Volatile Code	● Medium	VoteMining.sol	ⓘ Acknowledged

Description

The newly added `migrate` function, in commit f3683330fa6a00f6b0cf8d3dd652c3c1a8ea700a, has influence on the state variables `migrated`, `bondedBalances`, and `userTokenBalances`. As a migrating function, there is no updating on the data to the old `v1` contract.

Alleviation

No alleviation.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

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 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 CertiK's 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 CertiK 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. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's 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 CertiK 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, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK 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, CERTIK 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 ANY OTHER 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 FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE 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 CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK 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 OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER 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 CERTIK'S 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 CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK 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 CERTIK 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 OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.