Security Assessment

Intelligent Mining

Jun 14th, 2021

Table of Contents

Summary

This report has been prepared for Intelligent Mining to discover issues and vulnerabilities in the source code of the Intelligent Mining 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	Intelligent Mining
Description	Intelligent Mining
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/IM-Intelligent-Mining/contracts
Commit	<04df9915df4b4be2a0a28575c302d483fe5a4ac4> <59c6002e910801e9c3395f0bb679ee434069ea38>

Audit Summary

Delivery Date	Jun 14, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

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

Audit Scope

ID			File	SHA256 Checksum

Understandings

The Intelligent Mining Systems consists of three major components: IM token, the Staking contract, and the timelock contract. The following section demonstrates the main function and purpose of each component.

ImToken

The Intelligent Mining(IM) token is a standard ERC-20 token. When initializing the token contract, the contract mints 90000000 IM token, and send them to the contract deployer. The IM token is not available to mint or burn after contract deployment.

ImStaking

The staking contract contains four major functions:

1. `calcReward`

The `calcReward` function is called whenever a user deposits, withdraws or claims rewards of the staking pool. It first fetches the timestamp of the most recent update reward action and updates the variable `lastRewardTime[_msgSender()]` to the current timestamp. If the user has deposited before, calculate the user reward with an annual percentage yield(APY) of 12.

2. `stake`

The `stake` function is used for depositing IM tokens to the staking contract. It transfers the staking token to the staking contract and updates the user balance and lock start time.

3. `withdraw`

The `withdrawn` function sends the staking token back to the user. It first checks if the amount of token to withdraw is smaller than the user balance and if the lock time has passed. Then, it updates the user balance and sends the staking token back to the user.

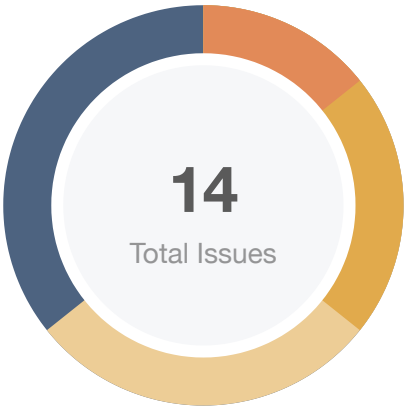
4. `withdrawnReward`

The `withdrawnReward` function sends staking rewards to the user. It calculates the reward, sets user reward to zero, and sends reward from address `fundsWallet` to the user.

TokenTimelock

Contract `TokenTimelock` is a token holder contract that will allow a beneficiary to extract the tokens after a given release time.

Findings



Critical	0 (0.00%)
Major	2 (14.29%)
Medium	3 (21.43%)
Minor	4 (28.57%)
Informational	5 (35.71%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
ITI-01	Potential attack vector on ERC20 API (approve/transferFrom methods)	Volatile Code	Major	Partially Resolved
ISI-01	Return value not handled	Volatile Code	Medium	Resolved
ISI-02	Division before multiplication	Mathematical Operations	Minor	Resolved
TTI-01	Missing event emitting	Coding Style	Informational	Resolved
ISI-03	Missing event emitting	Coding Style	Informational	Resolved
TTI-02	Ambiguous use of `virtual`	Volatile Code	Minor	Resolved
ISI-04	Ambiguous use of `virtual`	Volatile Code	Minor	Resolved
TTI-03	`external` over `public` function	Gas Optimization	Informational	Resolved
ISI-05	`external` over `public` function	Gas Optimization	Informational	Resolved
ISI-06	Insufficient check when withdrawing shares	Logical Issue	Medium	Resolved
ISI-07	Potential reentrancy attack	Volatile Code	Minor	Resolved
ISI-08	Possible to lock funds in Staking contract	Logical Issue	Medium	Resolved
ISI-09	Insufficient check in function `withdrawnReward`	Control Flow	Major	Resolved
ISI-10	Variable could be declared as `constant`	Gas Optimization	Informational	Resolved

ITI-01 | Potential attack vector on ERC20 API (approve/transferFrom methods)

Category	Severity	Location	Status
Volatile Code	Major	Global	Partially Resolved

Description

Approve method in ERC20.sol overrides current allowance regardless of whether spender already used it or not, so there is no way to increase or decrease allowance by certain value automatically unless token owner is a smart contract, not an account.

Here is a possible attack scenario:

1. Alice allows Bob to transfer N of Alice's tokens (N>0) by calling approve method on Token smart contract passing Bob's address and N as method arguments
2. After some time, Alice decides to change from N to M (M>0) the number of Alice's tokens Bob is allowed to transfer, so she calls approve method again, this time passing Bob's address and M as method arguments
3. Bob notices Alice's second transaction before it was mined and quickly sends another transaction that calls transferFrom method to transfer N Alice's tokens somewhere
4. If Bob's transaction will be executed before Alice's transaction, then Bob will successfully transfer N Alice's tokens and will gain an ability to transfer another M tokens
5. Before Alice noticed that something went wrong, Bob calls transferFrom method again, this time to transfer M Alice's tokens.

So, Alice's attempt to change Bob's allowance from N to M (N>0 and M>0) made it possible for Bob to transfer N+M of Alice's tokens, while Alice never wanted to allow so many of her tokens to be transferred by Bob.

For more details about this attack, please visit https://docs.google.com/document/d/1YLPtQxZu1UAvO9cZ1O2RPXBbT0mooh4DYKjA_jp-RLM/edit#.

Recommendation

We recommend using safeIncreaseAllowance and safeDecreaseAllowance in safeERC20.sol to increase and decrease user allowance to mitigate this issue.

Alleviation

The token contracts implements `increaseAllowance` and `decreaseAllowance` to mitigate this issue, but it is still possible to call `approve` to change the allowance.

ISI-01 | Return value not handled

Category	Severity	Location	Status
Volatile Code	Medium	Global	Resolved

Description

Multiple functions in contract `ImStaking` do not handle the return value of function `transferFrom` and `transfer`.

Recommendation

We recommend using variables to receive the return value of the functions mentioned above and handle both success and failure cases if needed by the business logic.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

ISI-02 | Division before multiplication

Category	Severity	Location	Status
Mathematical Operations	Minor	Global	Resolved

Description

In function `calcReward`, the following equation is used to calculate user reward: `_stakeBalances[_msgSender()].mul(staking_apy).div(100).mul(duration).div(365 days);`. The equation performs division before multiplication, which may cause rounding error.

Recommendation

We recommend perform multiplication before division.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

TTI-01 | Missing event emitting

Category	Severity	Location	Status
Coding Style	Informational	Global	Resolved

Description

There are a bunch of functions can change state variables. However, these functions do not emit event to pass the changes out of chain.

Recommendation

Recommend emitting events, for all the essential state variables that are possible to be changed during runtime.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

ISI-03 | Missing event emitting

Category	Severity	Location	Status
Coding Style	Informational	Global	Resolved

Description

There are a bunch of functions can change state variables. However, these functions do not emit event to pass the changes out of chain.

Recommendation

Recommend emitting events, for all the essential state variables that are possible to be changed during runtime.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

TTI-02 | Ambiguous use of `virtual`

Category	Severity	Location	Status
Volatile Code	Minor	Global	Resolved

Description

The linked functions ambiguously use the keyword virtual, yet they expected to be overridden.

Recommendation

We advise to remove the keyword virtual from the linked functions.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

ISI-04 | Ambiguous use of `virtual`

Category	Severity	Location	Status
Volatile Code	Minor	Global	Resolved

Description

The linked functions ambiguously use the keyword virtual, yet they expected to be overridden.

Recommendation

We advise to remove the keyword virtual from the linked functions.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

TTI-03 | `external` over `public` function

Category	Severity	Location	Status
Gas Optimization	Informational	Global	Resolved

Description

The linked functions remain unused by the contract.

Recommendation

We advise that the linked functions have their visiliblity changed to external to save gas.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

ISI-05 | `external` over `public` function

Category	Severity	Location	Status
Gas Optimization	Informational	Global	Resolved

Description

The linked functions remain unused by the contract.

Recommendation

We advise that the linked functions have their visiliblity changed to external to save gas.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

ISI-06 | Insufficient check when withdrawing shares

Category	Severity	Location	Status
Logical Issue	Medium	Global	Resolved

Description

The `withdrawn` function in contract `ImStaking` is used by users to withdraw their token from the staking contract. However, the function does not check if the argument `share` equals zero.

Recommendation

We recommend add sufficient checks for user-controlled data to prevent abnormal operations and potential attacks.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

ISI-07 | Potential reentrancy attack

Category	Severity	Location	Status
Volatile Code	Minor	Global	Resolved

Description

function `stake` calls `transferFrom` to transfer user token to staking contract. After calling `transferFrom`, variable `startLockTime[_msgSender()]` and `_stakeBalances[_msgSender()]` is updated, which violates Checks-Effects-Interactions Pattern. If the implementation of function `transferFrom` in token `im` is unknown, reentrancy is possible to take place.

Recommendation

Recommend using the Checks-Effects-Interactions Pattern to avoid the risk of calling unknown contracts.

Alleviation

The issue is resolved in commit "59c6002e910801e9c3395f0bb679ee434069ea38".

ISI-08 | Possible to lock funds in Staking contract

Category	Severity	Location	Status
Logical Issue	Medium	Global	Resolved

Description

If a user accidentally sends tokens directly to contract `ImStaking`, there is no way to withdraw such tokens. They will be locked in the contract forever.

Recommendation

We recommend adding a function to withdraw tokens in case of error operation or emergency. Note that such function should only be used by the contract operator and should not be able to withdraw funds used for staking.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

ISI-09 | Insufficient check in function `withdrawnReward`

Category	Severity	Location	Status
Control Flow	Major	Global	Resolved

Description

The function `withdrawnReward` does not check if a user has rewards for claiming before calling function `transferFrom` to transfer rewards to the user. A malicious user can call this function multiple times and eventually consumes all of the platform tokens(i.e., BNB/ETH) so that the staking contract cannot perform any operation.

Recommendation

We advise to checking the reward amount before making the transfer. If there is no reward, there is no need to make the transaction.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

ISI-10 | Variable could be declared as `constant`

Category	Severity	Location	Status
Gas Optimization	Informational	Global	Resolved

Description

Variables `lockTime` and `staking_apy` could be declared as `constant` since these state variables are never to be changed.

Recommendation

We recommend declaring those variables as `constant`.

Alleviation

The issue is resolved in commit "c4c2a0fd62d6d79bc726fe34a8ece861b2264438".

Appendix

Finding Categories

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

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.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

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.