DeHacker

Security Assessment

Illuvium

October 21th, 2022

Contents

CONTENTS
1SUMMARY
3ISSUE CATEGORIES
4OVERVIEW
5
PROJECT SUMMARY 5
AUDIT SUMMARY 5
VULNERABILITY SUMMARY 6
AUDIT SCOPE 6
UNDERSTANDINGS 7
OVERVIEW 7
PRIVILEGED FUNCTIONS 7
FINDINGS
9MAJOR
10
IGB-01 : Lack of Input Validation
DESCRIPTION 10
RECOMMENDATION 10
ALLEVIATION 11
IGB-02 : Requisite Value of ERC-20 `transferFrom()` 12
DESCRIPTION 12
RECOMMENDATION 12
THE TEAM SHOULD ENSURE USERS CAN GET ALL THE LPADDRESS THEY PURCHASED. 12
ALLEVIATION 12
MEDIUM 13
IGB-03 : Missing Error Messages 13
DESCRIPTION 13
RECOMMENDATION 13
ALLEVIATION 13
MINOR 14
IGT-01 : Missing Emit Events 14
DESCRIPTION 14
RECOMMENDATION 14
ALLEVIATION 14
LSI-01 : Centralization Related Risks 15
DESCRIPTION 15
RECOMMENDATION 15
ALLEVIATION 15
INFORMATIONAL 16
LSI-02 : `pauseDuration`IncorrectlyEmitted 16
DESCRIPTION 16
RECOMMENDATION 16

ALLEVIATION 16
DISCLAIMER 17
APPENDIX 18
FINDING CATEGORIES 18
CHECKSUM CALCULATION METHOD 18
ABOUT 19

Summary

DeHacker's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow/underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service/logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Issue Categories

Every issue in this report was assigned a severity level from the following:

Critical severity issues

A vulnerability that can disrupt the contract functioning in a number of scenarios or creates a risk that the contract may be broken.

Major severity issues

A vulnerability that affects the desired outcome when using a contract or provides the opportunity to use a contract in an unintended way.

Medium severity issues

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

Minor severity issues

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

Informational

A vulnerability that has informational character but is not affecting any of the code.

Overview

Project Summary

Project Name	Illuvium Land Sale Protocol
Platform	Ethereum
website	https://illuvium.io/
Type	GameFi
Language	Solidity
Codebase	https://github.com/IlluviumGame/land-sale-core
Commit	b331088ab710142b9776c053268303b7f189c1a 4a92d89d7e9dbdf4a3af6c3746cee1e0a0b94

Audit Summary

Delivery Date	October 21th, 2022
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

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

Audit scope

ID	File	SHA256 Checksum
LSL	contracts/lib/LandSvgLib.sol	54cfd190738df3a10133a7168d62c6fa5e0a3da45d5e423f944afbcb7155695
ACI	contracts/utils/AccessControl.sol	79d1bef7dabe60a72b67d6c8865d3bd812c417f9d6629773386567b87b99e396
ERE	contracts/interfaces/ERC721SpecExt.sol	049587f274a3d53d35a115ee22d5d2c4a4d2e3050e456078421ce0d4f315fed9
LER	contracts/interfaces/LandERC721Spec.sol	781247e832e3e794b17580e3ce8640e455958eaaf1c3835a9528d5533995a6e5

Understandings

Overview

Illuvium is a project for users to purchase lpAddress token.

Users use fundraising Address token to buy the lpAddress.

There is a fundraising goal that is set by the team. When the fundraising goal is reached, the team will only receive the designated amount mandated for the fundraising goal. The remaining tokens will be returned to users.

There is a startTime and endTime which represents the start time and end time of the purchase activity. There is a claimTime. After claim time, users can get lpAddress they purchased, and reclaim the excess proceeds from the remaining fundraisingAddress. The team can only claim the designated amount described per the fundraising Address not greater than the fundraising goal and extract the surplus lpAddress.

There is a whitelist strategy. Variable rand is used to mark users at a whitelist level. rank has four values:0, 1, 2, 3, respectively means: 0: never be a whitelist, 1: normal whitelist, 2: super whitelist, 3: ever a whitelist, and later become not a whitelist. Only users in the whitelist can purchase lpAddress. Users in the normal whitelist have a default purchase limit whitelistQuota. Users in the super whitelist can have a purchase limit that is set by operator.

If the user's mortgage num is greater than the threshold in a third pool poolAddress, he will be set as anormal whitelist when he purchases lpAddress.

Additionally, the contract FundraisingIdoPool can work correctly only when there is enough lpAddress in it. According to the codebase, the amount of lpAddress should be lpQuantitySold when the activity starts.

Privileged Functions

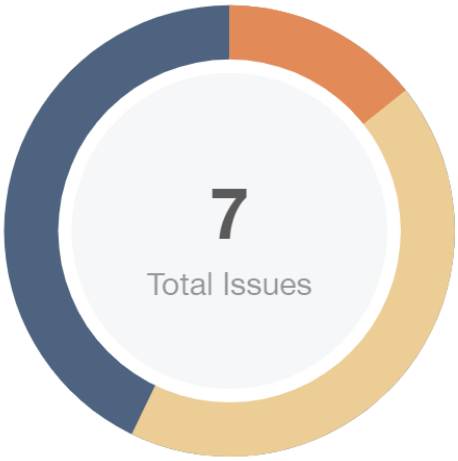
The project contains the following privileged functions. They are used to modify the contract configurationsand address attributes. We grouped these functions below:

The onlyOperator modifier:
contract FundraisingIdoPool.sol:

- function setPoolAddress()
- unction setAdminAddress()
- function setWhiteListQuota()
- function setExchangeRate()
- function setUpperLimit()

- function setEndTime()
- function setClaimTime()
- function addSuperWhiteList()
- function setWhiteList()
- function ownerClaim()
- function extractSurplusLp()

Findings



Critical	0 (0.00%)
Major	1 (14.29%)
Medium	0 (0.00%)
Minor	3 (42.86%)
Informational	3 (42.86%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
IGB-01	Lack Of Input Validation	Volatile Code	Minor	Resolved
IGB-02	Requisite Value Of ERC-20 transferFrom	Logical Issue	Minor	Resolved
IGB-03	Missing Error Messages	Coding Style	Informational	Resolved
IGT-01	Missing Emit Events	Coding Style	Informational	Resolved
LSI-01	Centralization Related Risk	Centralization	Major	Acknowledged
LSI-02	pauseDuration Incorrectly Emitted	Logical Issue	Informational	Resolved

Minor

IGB-01 | Lack Of Input Validation

Category	Severity	Location	Status
Volatile Code	Minor	LandLib.sol: 200	Resolved

Description

In the contract FundraisingIdoPool, the role operator has the authority over the following function:

1. Update poolAddress through setPoolAddress.
2. Update _adminAddress through setAdminAddress function.
3. Update whiteListQuota through setWhiteListQuota function.
4. Update exchangeRate through setExchangeRate function.
5. Update upperLimit[_account] through setUpperLimit function.
6. Update endTime through setEndTime function.
7. Update claimTime through setClaimTime function.
8. Set super whitelist through addSuperWhiteList function.
9. Set whitelist through setWhiteList function.
10. Claim fundRaisingAddress through ownerClaim function.
11. Extract surplus lpAddress through extractSurplusLp function.

without obtaining the consensus of the community.

Recommendation

We advise the client to carefully manage the owner, injector, operator 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., Multi signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at the different levels in terms 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

Consider adding the require() checks for Site Type and Landmark Type ID in setMetadata()

IGB-02| Requisite Value Of ERC-20 transferFrom() / transfer() Call

Category	Severity	Location	Status
LogicalIssue	Minor	UpgradeableERC721.sol:	Resolved

Description

While the ERC-20 implementation does necessitate that the transferFrom() / transfer() functionreturns a bool variable yielding true , many token implementations do not return anything i.e. Tether(USDT) leading to unexpected halts in code execution.

Recommendation

We advise that the SafeERC20.sol library is utilized by OpenZeppelin to ensure that the transferFrom() /transfer() function is safely invoked in all circumstances.

Alleviation

The team heeded our advice and removed the extractSurplusLp function. Added a logic in function ownerClaim to ensure there will be enough lpAddress in the contract for users to claim. Code change was applied in commit: d333c208cc2f1c9f52244f66773eda6f9fd7d3ad.

Informational

IGB-03| Missing Error Messages

Category	Severity	Location	Status
CodingStyle	Informational	AggregatorV3Mock.sol: 102	Resolved

Description

The require can be used to check for conditions and throw an exception if the condition is not met. It isbetter to provide a string message containing details about the error that will be passed back to the caller.

Recommendation

We advise adding error messages to the linked require statements.

Alleviation

[CertiK] : The team heeded the advice and resolved the finding in the commita92d89d7e9dbdf4a3af6c3746cee1e0a0b9403c3

IGT-01 | Missing Emit Events

CodingStyle	Informational	ERC721Impl.sol: 306, 333	Resolved

Description

There should always be events emitted in the sensitive functions that are controlled by centralization roles.

Recommendation

It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

Alleviation

[CertiK] : The team heeded the advice and resolved the finding in the commita92d89d7e9dbdf4a3af6c3746cee1e0a0b9403c3

LSI-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization	Major	LandSale.sol: 1	Acknowledged

Description

In the contract LandSale.sol , the following roles has authority over the following functions:ROLE_DATA_MANAGER role has authority over function setInputDataRoot()

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the securityoperation and level of decentralization, which in most cases cannot be resolved entirely at the presentstage. We advise the client to carefully manage the privileged account's private key to avoid any potentialrisks of being hacked.

Alleviation

[Illuvium] : Current deployment process implies transferring all the roles to Illuvium eDAO mSig wallet(4/6 signatures) It also implies that any permissions which are no longer required to extend, or/and upgradethe protocol to be revoked from the mSig We have a long-term plan to move these permissions to the DAOsmart contract with time-lock feature, controlled by the community in the decentralized way. This design iswell-known to the public and is the same for all the Illuvium smart contracts, including Illuvium Token itself,Staking contracts, and others.

Informational

LSI-02 | _pauseDuration Incorrectly Emitted

Category	Severity	Location	Status
Logical Issue	Informational	LandSale.sol: 629	Resolved

Description

In the function initialize() , when the sale is in paused state, the value of _pauseDuration will beincorrectly emitted in the Resumed event.

Recommendation

Consider emitting pauseDuration + now32() - pausedAt in the event.

Alleviation

[CertiK] : The team heeded the advice and resolved the finding in the commita92d89d7e9dbdf4a3af6c3746cee1e0a0b9403c3

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

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.

Coding Style

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

Volatile Code

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

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.

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.

About

DeHacker is a team of auditors and white hat hackers who perform security audits and assessments. With decades of experience in security and distributed systems, our experts focus on the ins and outs of system security. Our services follow clear and prudent industry standards. Whether it's reviewing the smallest modifications or a new platform, we'll provide an in-depth security survey at every stage of your company's project. We provide comprehensive vulnerability reports and identify structural inefficiencies in smart contract code, combining high-end security research with a real-world attacker mindset to reduce risk and harden code.

BLOCKCHAIINS

- Ethereum
- Cosmos
- Eos
- Substrate

TECH STACK

- Python
- Solidity
- Rust
- c++

CONTACTS

- https://dehacker.io
- https://twitter.com/dehackerio
- https://github.com/dehacker/audits_public
- https://t.me/dehackerio
- https://blog.dehacker.io/

DeHacker

October 2022