DeHacker

Code Security Assessment

Voxel

May27th, 2023

Contents

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	Voxel
Platform	polygon
Website	https://www.voxies.io/
Type	Gaming
Language	Solidity

Vulnerability Summary

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

Audit scope

ID	File	SHA256 Checksum
PVV	PolygonVoxel.sol	be2b61a3b4aea71bc302ae988d2d91bb5c9bac7c7404e7f0cd7771bf57fef5b5
VVC	Voxel.sol	dbfd51b37b7175e110f9421f7c842600db52846e8d864f32d1e88693dc036311
VDV	VoxelDistribution.sol	5b9037547d4a955a8709d6720086bfe0e6830e73878af70935530d989b037bf1

Findings

ID	Category	Severity	Status
EVE-01	Centralization /Privilege	Major	Acknowledged
VDV-01	Centralization /Privilege	Major	Acknowledged
VVC-01	Centralization /Privilege	Major	Resolved

Major

PVV-01 | Privileged Ownership

Category	Severity	Location	Status
Centralization / Privilege	Major	PolygonVoxel.sol: 580	Acknowledged

Description

In the contract PolygonVoxel , the owner has the authority to update the address of_ childChainManagerProxy through function updateChildChainManagerProxy() . And only the address_childChainManagerProxy can call function deposit() to get minted tokens.
Any compromise to the owner may allow the hacker to take advantage of this, gaining access to funds andthe power to manipulate the critical contract. Misuses of the function could cause financial losses.

Recommendation

We recommend carefully managing the _owner and _childChainManagerProxy accounts' private keys toavoid any potential risks of being hacked.
In general, we strongly recommend centralized privileges or roles in the protocol to be improved via adecentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g.,Multisignature wallets.
Indicatively, here are some feasible suggestions that would also mitigate the potential risk at the differentlevels in terms of short-term and long-term:
Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to theprivate key;
Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Major

VDV-01 | Privileged Ownership

Category	Severity	Location	Status
Centralization / Privilege	Major	VoxelDistribution.sol	Acknowledged

Description

In the contract VoxelDistribution , the owner can set admin accounts, and the owner or the adminaccounts has the authority over the following functions:
setMinimumBuyAmount()
setPrice()
withdraw()
withdrawToken()
pause()
unpause()

The owner and the admins of the contract have significant privileges. Admins can set the price of thetoken, and they can set the variable minimumBuyAmount . The owner has the ability to forbid or allow users tobuy tokens and the owner can withdraw the accumulated funds in the contract.

Any compromise to the admins and the owner may allow the hacker to take advantage of this contract,gaining access to funds and the power to manipulate the critical contract and the price. Misuses of thesefunctions could cause financial losses.

Recommendation

We recommend carefully managing the admins and the _owner accounts' private keys to avoid anypotential risks of being hacked.
In general, we strongly recommend centralized privileges or roles in the protocol to be improved via adecentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g.,Multisignature wallets.
Indicatively, here are some feasible suggestions that would also mitigate the potential risk at the differentlevels in terms of short-term and long-term:
Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to theprivate key;
Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Major

VVC-01 | Centralized Token Holding Position

Category	Severity	Location	Status
Centralization / Privilege	Major	Voxel.sol: 565	Resolved

Description

```
565     _mint(_msgSender(), 1000000 ether); // 1 mil
```

All of the tokens are distributed to the contract owner when deploying the contract. And the owner candistribute tokens without obtaining the consensus of the community.

Recommendation

Once the token goes live, we assume many transactions would involve the wallet unlock of the owneraddress and the team shall make enough efforts to restrict the access of the private key.

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

May 2023