VIBRANIUM AUDITS

Security Assessment

Origyn

Verified by Vibranium Audits on 22 July 2024

Vibranium Audits Verified on July 22nd, 2024

Origyn NFT

The security assessment was prepared by Vibranium Audits.

Executive Summary

TYPES	ECOSYSTEM	METHODS
DEFI/NFT	ICP	Manual Review, penetration testing

LANGUAGE	TIMELINE	KEY COMPONENTS
Motoko	Delivered on 22/07/2024	N/A

CODEBASE	COMMITS
https://github.com/ORIGYN-SA/origyn_nft/tree/0.1.6	d512653b9faab5a3f4828835aaac23af557c5910

Vulnerability Summary

5	0	0	0	5	0	0
Total Findings	Resolved	Mitigated	Partially Resolved	Acknowledged	Declined	Unresolved

Count	Severity	Status	Description
0	Critical		Critical vulnerabilities are usually straightforward to exploit and can lead to the loss of user funds or contract state manipulation by external or internal actors.
2	High	0 Resolved	High vulnerabilities are usually harder to exploit, requiring specific conditions, or have a more limited scope, but can still lead to the loss of user funds or contract state manipulation by external or internal actors.
0	Medium	0 Resolved	Medium vulnerabilities are usually limited to state manipulations, but cannot lead to assets loss. Major deviations from best practices are also in this category.
1	Low	0 Resolved	Low vulnerabilities are related to outdated and unused code or minor gas optimization. These issues won't have a significant impact on code execution, but affect the code quality.
2	Informational	0 Resolved	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | ORIGYN NFT

CODEBASE | ORIGYN NFT

Repository

https://github.com/ORIGYN-SA/origyn_nft/tree/0.1.6

Commits

d512653b9faab5a3f4828835aaac23af557c5910

AUDIT SCOPE | ORIGYN NFT

1 repo audited ● 1 repo with Acknowledged findings ● 0 files with Resolved findings

ID	Branch	Commit Hash
VOF	0.1.6	d512653b9faab5a3f4828835aaac23af557c5910

APPROACH & METHODS | ORIGYN NFT

This report has been prepared for ORIGYN NFT(2023) to discover issues and vulnerabilities in the source code of the ORIGYN NFT project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review, rigorous Penetration Testing and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Pen-Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the code base 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 code base 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:

- Testing the smart contracts against both common and uncommon attack vectors.
- Enhance general coding practices for better structures of source codes.
- Review unit tests to cover the possible use cases.
- Review functions for readability, especially for future development work.

FINDINGS | ORIGYN NFT

5	2	0	1	2
Total Findings	High	Medium	Low	Informational

This report has been prepared to discover issues and vulnerabilities for ORIGYN NFT. Through this audit, we have uncovered 5 issues ranging from different severity levels. Utilizing the techniques of Manual Review & Penetration Testing, we discovered the following findings:

ID	Title	Category	Severity	Status
VOF-01	Inter-Canister calls (Reentrancy)	Logical Issue	High	Acknowledged
VOF-02	Large Data Attacks	Logical Issue	High	Acknowledged
VOF-03	Rollback	Logical Issue	Informational	Acknowledged
VOF-04	Redundant Code/Comments	Best Practices	Informational	Acknowledged
VOF-05	Lack of cycle management	Logical Issue	Low	Acknowledged

VOF-01 | Inter-Canister calls (Reentrancy)

Category	Severity	Location	Status
Logical Issue	HIGH	Multiple Findings	Acknowledged

Description

Function announceTransaction (metadata.mo):

The function **announceTransaction** is responsible for sending a transaction announcement to an external canister using **Droute.publish**

If **state.state.collection_data.announce_canister** is read before the inter-canister call, there is an assumption that it will not change during the call.

However, due to the asynchronous nature of inter-canister calls, this state might be altered by another transaction or operation.

If the state changes between the read and the use, it might lead to incorrect logic execution or unauthorized access.

If the global state is modified before the inter-canister call and then again in the callback, there's a risk that other operations might interfere, leading to reentrancy issues or inconsistent state.

Example Scenario:

Initial State:

Let's assume **state.state.collection_data.announce_canister** is someCanister.

Inter-Canister Call Initiation:

The function starts executing and reads the announce_canister.

The inter-canister call to **Droute.publish** is initiated.

State Change by Another Operation:

While waiting for the response from **Droute.publish**, another operation changes **state.state.collection_data.announce_canister** to null or another canister ID.

This can happen if there is another transaction or administrative operation modifying the state.

Callback Execution:

The response from **Droute.publish** returns.

If the response is a failure, the function attempts to handle it, but the state might already be altered by another operation.

Other Cases:

register_escrow_sale_nft_origyn (main.mo)

Recommendation

Proper Failure Handling:

Ensure that any state changes made before the inter-canister call are properly handled or reverted in case of failure.

Example:

```
let originalState = state.state.collection_data.announce_canister;
let result = await Droute.publish(state.state.droute, eventName, payload);
if (result != #ok()) {
  // Revert state changes if necessary
  state.state.collection_data.announce_canister := originalState;
  // Handle failure, log error, etc.
}
```

AND/OR

Use Reentrancy Guards: Implement mechanisms to prevent reentrancy attacks, such as using boolean flags to indicate that a function is in progress.

VOF-02 | Large Data Attacks

Category	Severity	Location	Status
Logical Issue	HIGH	Multiple Findings	Acknowledged

Description

A large data attack involves an attacker sending excessively large inputs to a smart contract function, which can lead to high memory usage, excessive cycle consumption, and potentially cause denial-of-service (DoS) conditions. This type of attack is particularly dangerous because it can exploit the resource limitations of the Internet Computer (IC) canisters, draining their cycles and making them unresponsive.

In the context of **get_nat_as_token_id function (utils.mo)**, the vulnerability arises because the function does not impose any restrictions on the size of the Nat value it receives. Since Nat is an unbounded natural number, an attacker could send a very large Nat value, leading to high memory consumption when converting it to bytes and processing it further.

Large Nat Values:

- The tokenNat parameter is of type Nat, which is an unbounded natural number. This means it can represent extremely large values.
- Converting a very large Nat to bytes using Conversions.natToBytes(tokenNat) can lead to a large byte array, consuming significant memory.

Memory Consumption:

- The conversion of the large Nat value to bytes and the subsequent processing (e.g., creating buffers and appending them) can consume a significant amount of memory, leading to potential memory exhaustion.

Cycle Consumption:

- Processing large amounts of data can consume a lot of cycles, draining the canister's cycle balance quickly.

Other cases:

owner.mo: public func transferDip721
public func transferICRC7
ledger_interface.mo: public func send_payment_minus_fee

Recommendation

To mitigate the risk of large data attacks, implement the following strategies:

1. Restrict Input Size:
    - Implement checks to restrict the size of the Nat value before processing it. This prevents excessively large inputs from consuming too many resources.
2. Validate Data Size:
    - Before performing any operations, validate the size of the data to ensure it is within acceptable limits.

VOF-03 | Rollbacks

Categor y	Severity	Location	Status
Logical Issue	Informational	Multiple Findings	Acknowledged

Description

In the context of the Internet Computer and the Motoko programming language, the rollback vulnerability can occur when a canister's state changes and then an error occurs, causing a rollback of those changes. The behavior of rollbacks can be surprising and error-prone because not all error conditions trigger a rollback.

- Throwing an Error: When a function throws an error using throw, the state changes that happened before the throw are not rolled back.
- Trapping (e.g., assert or Debug.trap): When a function traps, the state changes before the trap are rolled back. This can lead to inconsistencies if not properly handled.

Although this vulnerability is originally of HIGH level, it does not particularly exist within the codebase but is mentioned within comments, the planned usage of ‘throw’ in the future:

owner.mo Line 157
mint.mo Line 1375
market.mo Line 3173/3364

Recommendation

- **Understand the Difference between Throw and Trap:**
  Be aware that throw does not roll back state changes, while trap (e.g., assert failures) does.
- **Validate Conditions before State Changes:**
  Perform all necessary validations before making any state changes to ensure that the state changes only occur when all conditions are met.
- **Use Try-Catch for Error Handling:**
  Use try-catch blocks to handle errors gracefully and ensure that any necessary rollback logic is explicitly handled.

VOF-05 | Lack of cycle management

Categor y	Severity	Location	Status
Logical Issue	Low	main.mo	Acknowledged

Description

Certain functions perform multiple tasks and includes several operations that could lead to cycle consumption issues. These issues arise primarily from the complexity and length of the function.

Although generally throughout the codebase, there has been great and consistent checks for the functions' cycle consumption, how certain functions lacked these checks which can potentially cause threats:

main.mo: redeem_allocation_sale_nft_origyn
register_escrow_sale_nft_origyn

Recommendation

- Use Cycle Estimation:

Estimate the cycles required for different parts of the function and monitor cycle consumption. Adjust the logic to ensure it stays within acceptable limits.

VOF-04 | Redundant Code

Categor y	Severity	Location	Status
Logical Issue	Informational	General	Acknowledged

Description

The codebase contains a significant amount of redundant code, primarily consisting of numerous commented-out functions. These commented functions occupy a substantial amount of space, contributing to code bloat and negatively impacting the maintainability and readability of the code. Such redundant code includes:

1. Obsolete Functions: Functions that were once used but are no longer relevant to the current functionality of the application. These have been commented out but not removed from the codebase.
2. Duplicate Functions: Functions that replicate the functionality of other parts of the code. These may have been commented out during refactoring but were not deleted.
3. Unused Experimental Functions: Functions that were written for experimental purposes or as part of a feature that was never fully implemented or integrated. These functions remain in the code as comments.

DISCLAIMER | VIBRANIUM AUDITS

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 Vibranium Audits 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 Vibranium Audits 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 Vibranium Audits position is that each company and individual are responsible for their own due diligence and continuous security Vibranium Audits 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 Vibranium Audits 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 VIBRANIUM AUDITS HEREBY DISCLAIMS ALL WARRANTIES WHETHER EXPRESS IMPLIED STATUTORY OR OTHERWISE WITH RESPECT TO THE SERVICES ASSESSMENT REPORT OR OTHER MATERIALS WITHOUT LIMITING THE FOREGOING VIBRANIUM AUDITS 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 VIBRANIUM AUDITS 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 ANOTHER 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 FORGOING, VIBRANIUM AUDITS PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KINDTHAT 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 VIBRANIUM AUDITS NOR ANY OF VIBRANIUM AUDITS AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND EXPRESS OR IMPLIED AS TO THE ACCURACY RELIABILITY ORCURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE VIBRANIUM AUDITS 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 OWNEROR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS

THE SERVICES ASSESSMENT REPORT AND ANY OTHER MATERIALS HERE UNDER 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 VIBRANIUM AUDITS 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 VIBRANIUM AUDITS WITH RESPECT TO SUCH SERVICES ASSESSMENT REPORT AND ANY ACCOMPANYING MATERIALS

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

Vibranium | Securing the Web3 World

Vibranium Audits is a blockchain security company that was founded in 2021 by professors from the University of Greenwich and cyber-security engineers from ITI Capital. As pioneers in the field, Vibranium Audits utilizes best-in-class Formal Verification and AI technology to secure and monitor blockchains, smart contracts, and Web3 apps.