



Smart contracts security assessment

Final report

[Tariff: Standard](#)

ERC721R

June 2022



0xguard.com



hello@0xguard.com

Contents

1. Introduction	3
2. Contracts checked	3
3. Procedure	3
4. Known vulnerabilities checked	4
5. Classification of issue severity	5
6. Issues	5
7. Conclusion	7
8. Disclaimer	8

Introduction

The report has been prepared for Exodia Labs ERC721R. ERC721R adds trustless refunds to NFT smart contracts allowing minters to return the NFTs minted at a cost within a given refund period. ERC721RExample contract is based on gas efficient ERC721A implementation. The code is available in the Github [repository](#). The code was checked in the [324f41d](#) commit.

Name	ERC721R
Audit date	2022-06-24 - 2022-06-26
Language	Solidity
Platform	Ethereum

Contracts checked

Name	Address
ERC721R	https://github.com/exo-digital-labs/ERC721R/blob/324f41ded89f62b38e5ae41d272d703874e82b8d/contracts/ERC721RExample.sol

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Manually analyze smart contracts for security vulnerabilities

- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
<u>Unencrypted Private Data On-Chain</u>	passed
<u>Code With No Effects</u>	passed
<u>Message call with hardcoded gas amount</u>	passed
<u>Typographical Error</u>	passed
<u>DoS With Block Gas Limit</u>	passed
<u>Presence of unused variables</u>	passed
<u>Incorrect Inheritance Order</u>	passed
<u>Requirement Violation</u>	passed
<u>Weak Sources of Randomness from Chain Attributes</u>	passed
<u>Shadowing State Variables</u>	passed
<u>Incorrect Constructor Name</u>	passed
<u>Block values as a proxy for time</u>	passed
<u>Authorization through tx.origin</u>	passed
<u>DoS with Failed Call</u>	passed
<u>Delegatecall to Untrusted Callee</u>	passed
<u>Use of Deprecated Solidity Functions</u>	passed
<u>Assert Violation</u>	passed
<u>State Variable Default Visibility</u>	passed
<u>Reentrancy</u>	passed
<u>Unprotected SELFDESTRUCT Instruction</u>	passed

<u>Unprotected Ether Withdrawal</u>	passed
<u>Unchecked Call Return Value</u>	passed
<u>FloatingPragma</u>	passed
<u>Outdated Compiler Version</u>	passed
<u>Integer Overflow and Underflow</u>	passed
<u>Function Default Visibility</u>	passed

Classification of issue severity

High severity	High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.
Medium severity	Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.
Low severity	Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

No issues were found

Medium severity issues

No issues were found

Low severity issues

1. Gas optimization (ERC721R)

The `getRefundGuaranteeEndTime()` function can be declared as external to save gas.

2. Lacks validation of input parameters (ERC721R)

The contract function `setRefundAddress()` does not check the address `_refundAddress` against a null address.

3. Few events (ERC721R)

Many functions from the contract lack events:

1. `setRefundAddress()`
2. `setMerkleRoot()`
3. `setBaseURI()`
4. `toggleRefundCountdown()`
5. `togglePresaleStatus()`
6. `togglePublicSaleStatus()`

Conclusion

ERC721R ERC721R contract was audited. 3 low severity issues were found.

Reviewed ERC721RExample contract cannot be considered an ERC standard, since after inheriting from this implementation, there may be problems with the extensibility of the code or changes in any of its parts. This contract is suitable as a kind of auxiliary functionality, but it cannot be considered a standard. The README.md documentation is also insufficient for adequate standard description. To create a standard from this idea, we suggest the Exodia Labs (or community) to write a technical article and documentation and send it to [EIP](#) (Ethereum Improvement Proposals). To understand what needs to be provided and how to arrange everything, you can read this [article](#). There is a [template](#) EIP.

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 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 without 0xGuard prior written consent.

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 0xGuard 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.

