GUARDIAN AUDITS

SMART CONTRACT SECURITY AUDIT OF

NFTR

Summary

Audit Firm: Guardian Audits

Client Firm: NFTR

Final Report Date - Preliminary Report

Audit Summary

After a line by line manual analysis and automated review, Guardian Audits has concluded that:

- NFTR's smart contracts have a LOW RISK SEVERITY
- NFTR's smart contracts have an **ACTIVE OWNERSHIP**
- Important owner privileges – withdraw, withdrawRNM, curateCollection, updateNamingCreditsProtocolFeeRecipient, shutOffAssignments, assignNamingCredits, setSpecialNames, updateNamingPriceEther, updateNamingPriceRNM, updateProtocolFeeRecipient, updateRnmNamingStartBlock
- NFTR's smart contract owner has multiple "write" privileges. Centralization risk correlated to the active ownership is **MEDIUM**

Notice that the examined smart contracts are not resistant to internal exploit. For a detailed understanding of risk severity, source code vulnerability, and potential attack vectors, refer to the complete audit report below.

🔗 Blockchain network: **Ethereum**

✅ Verify the authenticity of this report on Guardian's GitHub: https://github.com/guardianaudits

Table of Contents

Project Information

Smart Contract Risk Assessment

Report Summary

Addendum

Project Overview

Project Summary

Project Name	NFTR
Language	Solidity
Codebase	https://github.com/greatrat00/NFTRegistryAudit
Commit	8ebfa4136516fa267d3522f2b59138f3f6e222f0

Audit Summary

Delivery Date	Preliminary Report
Audit Methodology	Static Analysis, Manual Review

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● High	0	0	0	0	0	0
● Medium	4	4	0	0	0	0
● Low	7	7	0	0	0	0

Audit Scope & Methodology

Scope

ID	File	SHA-1 Checksum
NFTR	NFTRegistry.sol	0A7AA5273A8514E2A367B40048CA8DFD269956CE

Methodology

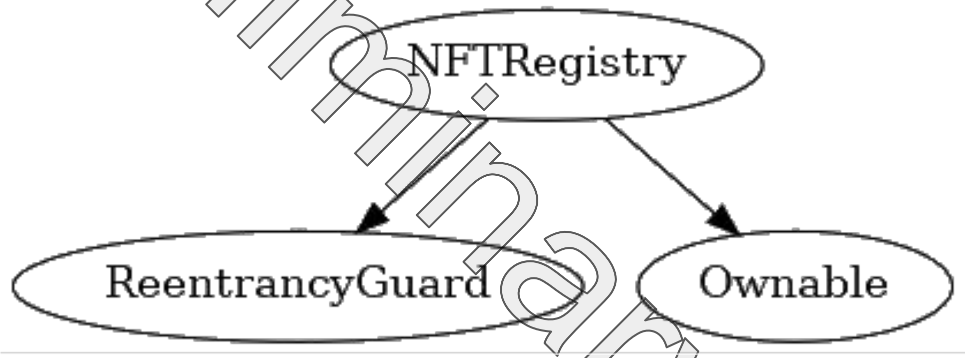
The auditing process pays special attention to the following considerations:

- 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.

Vulnerability Classifications

Vulnerability Level	Classification
● Critical	Easily exploitable by anyone, causing loss/manipulation of assets or data.
● High	Arduously exploitable by a subset of addresses, causing loss/manipulation of assets or data.
● Medium	Inherent risk of future exploits that may or may not impact the smart contract execution.
● Low	Minor deviation from best practices.

Inheritance Graph



Findings & Resolutions

ID	Title	Category	Severity	Status
NFTR-1	Centralization Risk	Centralization / Privilege	● Medium	Unresolved
NFTR-2	Hold Farming Naming Not Free	Logical Error	● Medium	Unresolved
NFTR-3	Simpler Code	Best Practices	● Low	Unresolved
NFTR-4	Transfer vs. TransferFrom	Best Practices	● Low	Unresolved
NFTR-5	Validation Upon Name Transfer	Best Practices	● Low	Unresolved
NFTR-6	Zero Address Checks	Best Practices	● Low	Unresolved
NFTR-7	Unnecessary Boolean Checks	Optimization	● Low	Unresolved
NFTR-8	Unnecessary Casting	Optimization	● Low	Unresolved
NFTR-9	Typo	Typo	● Low	Unresolved
NFTR-10	Potential DoS	Denial-of-Service	● Medium	Unresolved
NFTR-11	Lost Names With Burnable NFT	Logical Error	● Medium	Unresolved

NFTR-1 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● Medium	NFTRegistry.sol	Unresolved

Description

The owner address has the ability to repeatedly change namingPriceRNM by calling updateNamingPriceRNM, but the function lacks any lower and upper bounds on the input. As a result, the owner can modify the RNM price to be as large as possible and frontrun a buyer's transaction. This would lead to the user experiencing a larger decrease of assets than intended.

Additionally, the owner address holds potentially exploitative abilities to: withdraw, withdrawRNM, curateCollection, updateNamingCreditsProtocolFeeRecipient, shutOffAssignments, assignNamingCredits, setSpecialNames, updateNamingPriceEther, updateProtocolFeeRecipient, updateRnmNamingStartBlock.

Recommendation

Consider defining lower and upper bounds on namingPriceRNM.

Furthermore, consider making owner a multi-sig, optionally with a timelock for improved community oversight.

Resolution

NFTR-2 | Hold Farming Naming Not Free

Category	Severity	Location	Status
Logical Error	● Medium	NFTRegistry.sol	Unresolved

Description

According to the docs, if an NFT is curated and is in the hold farming period, naming can be free. However, the `changeName` function still expects payment in the specified currency even if the NFT collection is curated and in the hold farming period.

Recommendation

Implement logic such that a NFT in the holding period can be named for free.

Resolution

NFTR-3 | Simpler Code

Category	Severity	Location	Status
Best Practices	● Low	NFTRegistry.sol: 336, 363	Unresolved

Description

Line 336: Because checkOwnership must get the owner and compare it against the msg.sender, the function can simply utilize getOwner instead of duplicating the logic for retrieving the NFT owner.

Line 363: isTokenStructEmpty can simply be return token_in.collectionAddress == address(0) && token_in.tokenId == 0; The statement itself returns a boolean so an if-else is not needed.

Recommendation

Implement the above simplifications.

Resolution

NFTR-4 | Transfer vs TransferFrom

Category	Severity	Location	Status
Best Practices	● Low	NFTRegistry.sol: 507	Unresolved

Description

transferFrom is used to transfer RNM from the NFTRegistry contract to the owner, but a transfer could be used instead so approvals can be avoided.

Recommendation

Consider using the transfer function if the RNM token allows it.

Resolution

NFTR-5 | Validation Upon Name Transfer

Category	Severity	Location	Status
Best Practices	● Low	NFTRegistry.sol: 172	Unresolved

Description

transferName can potentially lead to an NFT's name being overwritten without permission from the holder. There must be proper validation done by the marketplace to make sure people can't arbitrarily send names to another person's NFT.

Recommendation

Ensure the marketplace contract has the necessary validation checks in place to only transfer a name if the to address acknowledges the transaction whether it is through a name purchase or some other means.

Resolution

NFTR-6 | Zero Address Checks

Category	Severity	Location	Status
Best Practices	● Low	NFTRegistry.sol: 93	Unresolved

Description

The constructor can benefit from zero address checks to help prevent errors during deployment.

Recommendation

Focus on creating seamless deploy scripts and consider adding zero address checks

Resolution

NFTR-7 | Unnecessary Boolean Checks

Category	Severity	Location	Status
Optimization	● Low	NFTRegistry.sol	Unresolved

Description

The contract frequently performs `variable == true` or `variable == false` which is unnecessary and gas inefficient.

Recommendation

Replace `require(variable == true)` with `require(variable)`.

Replace `require(variable == false)` with `require(!variable)`.

Resolution

NFTR-8 | Unnecessary Casting

Category	Severity	Location	Status
Optimization	● Low	NFTRegistry.sol	Unresolved

Description

There is no need to cast holdFarmingAddress to the IHoldFarming interface in functions such as curateCollection as it was already declared with the IHoldFarming type. The variable was not cast in initiateRetroactiveHoldFarming. The same can be said for the namingCreditsAddress variable.

Recommendation

Consider removing the explicit casts to save gas.

Resolution

NFTR-9 | Typo

Category	Severity	Location	Status
Typo	● Low	NFTRegistry.sol: 44	Unresolved

Description

“Naiming” is misspelled in the comment.

Recommendation

Correct the spelling for cleaner docs.

Resolution

NFTR-10 | Potential DoS

Category	Severity	Location	Status
Denial-of-Service	● Medium	NFTRegistry.sol: 486	Unresolved

Description

The initiateRetroactiveHoldFarming function calls the initiateHoldFarmingForNFT function in the Hold Farming contract 10,000 times. This may exceed the block gas limit and prevent any collection from getting curated.

Recommendation

Ensure that the block gas limit limit is not exceed or consider executing calls to initiateHoldFarmingForNFT in batches.

Resolution

NFTR-11 | Lost Names With Burnable NFT

Category	Severity	Location	Status
Logical Error	● Medium	NFTRegistry.sol	Unresolved

Description

Consider the scenario where a user registers a special name for their ERC721-compliant burnable NFT. They then proceed to burn their NFT, and ownership is relinquished. As a result, the name of the NFT cannot be changed nor transferred. The special name, a coveted asset to the NFTR protocol, is now lost.

Recommendation

Consider whether or not this is expected behavior. If unexpected, add a function so that if an owner does not exist for a particular NFT, then that NFT's registered name can be dereserved.

Resolution

Auditor's Verdict

After a line by line manual analysis and automated review, Guardian Audits has concluded that:

- NFTR's smart contracts have a LOW RISK SEVERITY
- NFTR's smart contracts have an **ACTIVE OWNERSHIP**
- Important owner privileges – `withdraw`, `withdrawRNM`, `curateCollection`, `updateNamingCreditsProtocolFeeRecipient`, `shutOffAssignments`, `assignNamingCredits`, `setSpecialNames`, `updateNamingPriceEther`, `updateNamingPriceRNM`, `updateProtocolFeeRecipient`, `updateRnmNamingStartBlock`
- NFTR's smart contract owner has multiple "write" privileges. Centralization risk correlated to the active ownership is **MEDIUM**

Disclaimer

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 Guardian 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. Guardian’s position is that each company and individual are responsible for their own due diligence and continuous security. Guardian’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 Guardian 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.

Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract’s safety and security. Therefore, Guardian does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

About Guardian Audits

Founded in 2022 by DeFi experts, Guardian Audits is a leading audit firm in the DeFi smart contract space. With every audit report, Guardian Audits upholds best-in-class security while achieving our mission to relentlessly secure DeFi.

To learn more, visit https://guardianaudits.com

To view our audit portfolio, visit https://github.com/guardianaudits

To book an audit, message https://t.me/guardianaudits