SolidityScan | Powered by CRED SHIELDS

Security Assessment Report

NFTLendAuction v1

8 Dec 2024

This security assessment report was prepared by SolidityScan.com, a cloud-based Smart Contract Scanner.

Self-published

This automated audit report was Self-published by the user. To learn more about our published reports click here.

Table of Contents

DEFINE CONSTRUCTOR AS PAYABLE
GAS OPTIMIZATION FOR STATE VARIABLES
GAS OPTIMIZATION IN INCREMENTS
LONG REQUIRE/REVERT STRINGS
PUBLIC CONSTANTS CAN BE PRIVATE
SPLITTING REQUIRE STATEMENTS
STORAGE VARIABLE CACHING IN MEMORY
USE SELFBALANCE() INSTEAD OF ADDRESS(THIS).BALANCE

05 Scan History

06 Disclaimer

01. Vulnerability Classification and Severity

Description

To enhance navigability, the document is organized in descending order of severity for easy reference. Issues are categorized as *Fixed*, *Pending Fix*, or *Won't Fix*, indicating their current status. *Won't Fix* denotes that the team is aware of the issue but has chosen not to resolve it. Issues labeled as *Pending Fix* state that the bug is yet to be resolved. Additionally, each issue's severity is assessed based on the risk of exploitation or the potential for other unexpected or unsafe behavior.

Critical

The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

High

High-severity vulnerabilities pose a significant risk to both the Smart Contract and the organization. They can lead to user fund losses, may have conditional requirements, and are challenging to exploit.

Medium

The issue affects the ability of the contract to operate in a way that doesn't significantly hinder its behavior.

Low

The issue has minimal impact on the contract's ability to operate.

Informational

The issue does not affect the contract's operational capability but is considered good practice to address.

Gas

This category deals with optimizing code and refactoring to conserve gas.

02. Executive Summary

NFTLendAuction v1

Uploaded Solidity File(s)

Published on 19 Dec 2024

Language	Audit Methodology	Website
Solidity	**Static Scanning**	**https://vaultlayer.xyz/**
Publishers/Owner Name	**Organization**	**Contact Email**
-	**VaultLayer**	-

Security Score is GREAT

The SolidityScan score is calculated based on lines of code and weights assigned to each issue depending on the severity and confidence. To improve your score, view the detailed result and leverage the remediation solutions provided.

This report has been prepared for NFTLendAuction v1 using SolidityScan to scan and discover vulnerabilities and safe coding practices in their smart contract including the libraries used by the contract that are not officially recognized. The SolidityScan tool runs a comprehensive static analysis on the Solidity code and finds vulnerabilities ranging from minor gas optimizations to major vulnerabilities leading to the loss of funds. The coverage scope pays attention to all the informational and critical vulnerabilities with over (100+) modules. The scanning and auditing process covers the following areas:

Various common and uncommon attack vectors will be investigated to ensure that the smart contracts are secure from malicious actors. The scanner modules find and flag issues related to Gas optimizations that help in reducing the overall Gas cost It scans and evaluates the codebase against industry best practices and standards to ensure compliance It makes sure that the officially recognized libraries used in the code are secure and up to date.

The SolidityScan Team recommends running regular audit scans to identify any vulnerabilities that are introduced after NFTLendAuction v1 introduces new features or refactors the code.

03. Findings Summary

NFTLendAuction v1
File Scan

Security Score	Scan duration	Lines of code
83.80/100	7 secs	446

2	1	0	2	24	30
Crit	High	Med	Low	Info	Gas

This audit report has not been verified by the SolidityScan team. To learn more about our published reports. click here

ACTION TAKEN

0	5	3	56
Fixed	*False Positive*	*Won't Fix*	*Pending Fix*

S. No.	Severity	Bug Type	Instances	Detection Method	Status
C001	Critical	INCORRECT ACCESS CONTROL	2	Automated	*Pending Fix*
C002	Critical	LOCKED ETHER	1	Automated	*False Positive*
H001	High	UNCHECKED TRANSFER	1	Automated	*Pending Fix*
M001	Medium	UNCHECKED ARRAY LENGTH	2	Automated	*False Positive*
L001	Low	MISSING EVENTS	4	Automated	*Partially fixed*
I001	Informational	BLOCK VALUES AS A PROXY FOR TIME	3	Automated	*Pending Fix*
I002	Informational	MISSING INDEXED KEYWORDS IN EVENTS	9	Automated	*Pending Fix*
I003	Informational	MISSING PAYABLE IN CALL FUNCTION	2	Automated	*Pending Fix*
I004	Informational	NAME MAPPING PARAMETERS	4	Automated	*Pending Fix*
I005	Informational	REDUNDANT STATEMENTS	1	Automated	*Pending Fix*
I006	Informational	USE CALL INSTEAD OF TRANSFER OR SEND	5	Automated	*Pending Fix*
G001	Gas	ARRAY LENGTH CACHING	2	Automated	*Pending Fix*
G002	Gas	ASSIGNING TO STRUCTS CAN BE MORE EFFICIENT	1	Automated	*Pending Fix*
G003	Gas	AVOID RE-STORING VALUES	3	Automated	*Pending Fix*
G004	Gas	CACHE ADDRESS(THIS) WHEN USED MORE THAN ONCE	5	Automated	*Pending Fix*

S. No.	Severity	Bug Type	Instances	Detection Method	Status
G005	Gas	CHEAPER CONDITIONAL OPERATORS	2	Automated	*Pending Fix*
G006	Gas	CHEAPER INEQUALITIES IN REQUIRE()	3	Automated	*Pending Fix*
G007	Gas	DEFINE CONSTRUCTOR AS PAYABLE	1	Automated	*Pending Fix*
G008	Gas	GAS OPTIMIZATION FOR STATE VARIABLES	2	Automated	*Pending Fix*
G009	Gas	GAS OPTIMIZATION IN INCREMENTS	1	Automated	*Pending Fix*
G010	Gas	LONG REQUIRE/REVERT STRINGS	2	Automated	*Pending Fix*
G011	Gas	PUBLIC CONSTANTS CAN BE PRIVATE	2	Automated	*Pending Fix*
G012	Gas	SPLITTING REQUIRE STATEMENTS	1	Automated	*Pending Fix*
G013	Gas	STORAGE VARIABLE CACHING IN MEMORY	7	Automated	*Pending Fix*
G014	Gas	USE SELFBALANCE() INSTEAD OF ADDRESS(THIS).BALANCE	1	Automated	*Pending Fix*

04. Vulnerability Details

Issue Type

INCORRECT ACCESS CONTROL

S. No.	Severity	Detection Method	Instances
C001	Critical	Automated	2

Description

Access control plays an important role in the segregation of privileges in smart contracts and other applications. If th is is misconfigured or not properly validated on sensitive functions, it may lead to loss of funds, tokens and in some c ases compromise of the smart contract.

The contract is importing an access control library but the function is missing the modifier.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_61	/NFTLendAuction.sol	L181 - L206	*Won't Fix*

/NFTLendAuction.sol L181 - L206

```
180         */
181       function placeBid(uint256 loanId, uint256 interestRate)
182           external
183           payable
184           nonReentrant
185           loanExists(loanId)
186           isNotAccepted(loanId)
187       {
188           Loan storage loan = loans[loanId];
189           require(
190               interestRate < loan.currentInterestRate && interestRate <= loan.maxInterestRat
      e,
191               "Bid interest rate invalid"
192           );
193           require(msg.value == loan.loanAmount, "Incorrect loan amount");
194
195           // Refund the previous lender if there is one
196           if (loan.lender != address(0)) {
197               payable(loan.lender).transfer(escrowedFunds[loanId]);
198           }
199
200           // Update loan details
201           loan.lender = msg.sender;
```

/NFTLendAuction.sol L181 - L206

```
200           // Update loan details
201           loan.lender = msg.sender;
202           loan.currentInterestRate = interestRate;
203           escrowedFunds[loanId] = msg.value;
204
205           emit LoanBidPlaced(loanId, loan);
206       }
207
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_62	/NFTLendAuction.sol	L346 - L377	*Won't Fix*

/NFTLendAuction.sol L346 - L377

```
345        */
346       function claimDefaultedLoan(uint256 loanId)
347           external
348           payable
349           nonReentrant
350           loanExists(loanId)
351       {
352           Loan storage loan = loans[loanId];
353           require(loan.isAccepted, "Loan not accepted");
354           require(block.timestamp > loan.startTime + loan.duration, "Loan not expired");
355
356           // Calculate the hypothetical repayment amount
357           uint256 interestAmount = (loan.loanAmount * loan.currentInterestRate) / 10000;
358           uint256 totalRepayment = loan.loanAmount + interestAmount;
359
360           // Calculate the lender's protocol fee
361           uint256 lenderProtocolFee = (totalRepayment * protocolFeeRate) / 10000;
362
363           // Ensure the lender sends the required protocol fee
364           require(msg.value == lenderProtocolFee, "Incorrect protocol fee sent");
365
366           // Update protocol fee balance
367           protocolFeeBalance += lenderProtocolFee;
```

Comments

These functions are opened to access for everyone intentionally.

/NFTLendAuction.sol

L346 - L377

```
365
366             // Update protocol fee balance
367             protocolFeeBalance += lenderProtocolFee;
368
369             // Transfer NFT to the lender
370             loan.isAccepted = false;
371             IERC721(loan.nftAddress).safeTransferFrom(address(this), loan.lender, loan.tokenId);
372
373
374             emit LoanDefaulted(loanId, loan);
375
376             _removeActiveLoan(loanId);
377         }
378
379
```

Issue Type

LOCKED ETHER

S. No.	Severity	Detection Method	Instances
C002	Critical	Automated	1

Description

The smart contract is accepting Ether at its address. This ether can be stored but due to misconfigurations or missin g functions, there is no way to transfer this Ether out of the contract's address. This causes the Ether to be locked in side the contract.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_42	/NFTLendAuction.sol	L14 - L14	*False Positive*

/NFTLendAuction.sol L14 - L14

```
13     */
14    contract NFTLendAuction is ReentrancyGuard, AccessControl {
15        bytes32 public constant OWNER_ROLE = keccak256("OWNER_ROLE");
16        bytes32 public constant MANAGER_ROLE = keccak256("MANAGER_ROLE");
```

Issue Type

UNCHECKED TRANSFER

S. No.	Severity	Detection Method	Instances
H001	High	Automated	1

Description

Some tokens do not revert the transaction when the transfer or transferFrom fails and returns False. Hence we must check the return value after calling the `transfer` or `transferFrom` function.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_8	/NFTLendAuction.sol	L151 - L151	*Won't Fix*

/NFTLendAuction.sol L151 - L151

```
150         // Transfer the NFT to the contract
151         IERC721(nftAddress).transferFrom(msg.sender, address(this), tokenId);
152
153         // Create a new loan
```

Comments

No need to change here since it already uses transferFrom with msg.sender initiating the transfer

Issue Type

UNCHECKED ARRAY LENGTH

S. No.	Severity	Detection Method	Instances
M001	Medium	Automated	2

Description

Ethereum is a very resource-constrained environment. Prices per computational step are orders of magnitude higher than with centralized providers. Moreover, Ethereum miners impose a limit on the total number of Gas consumed in a block. If `array.length` is large enough, the function exceeds the block gas limit, and transactions calling it will nev er be confirmed.

```
for (uint256 i = 0; i < array.length ; i++) { cosltyFunc(); }
```

This becomes a security issue if an external actor influences `array.length`.
E.g., if an array enumerates all registered addresses, an adversary can register many addresses, causing the problem described above.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_24	/NFTLendAuction.sol	L393 - L393	*False Positive*

/NFTLendAuction.sol L393 - L393

```
392             delete activeLoans[loanId];
393             for (uint256 i = 0; i < activeLoanIds.length; ) {
394                 if (activeLoanIds[i] == loanId) {
395                     activeLoanIds[i] = activeLoanIds[activeLoanIds.length - 1];
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_25	/NFTLendAuction.sol	L430 - L430	*False Positive*

/NFTLendAuction.sol L430 - L430

```
429         uint256 totalEscrowedFunds = 0;
430         for (uint256 i = 0; i < activeLoanIds.length; i++) {
431             totalEscrowedFunds += escrowedFunds[activeLoanIds[i]];
432         }
```

Issue Type

MISSING EVENTS

S. No.	Severity	Detection Method	Instances
L001	Low	Automated	4

Description

Events are inheritable members of contracts. When you call them, they cause the arguments to be stored in the trans action's log—a special data structure in the blockchain.
These logs are associated with the address of the contract which can then be used by developers and auditors to ke ep track of the transactions.
The contract was found to be missing these events on the function which would make it difficult or impossible to trac k these transactions off-chain.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_26	/NFTLendAuction.sol	L98 - L100	*False Positive*

/NFTLendAuction.sol L98 - L100

```
97        // Admin can grant roles to other addresses
98        function grantManagerRole(address account) external onlyRole(DEFAULT_ADMIN_ROLE) {
99            grantRole(MANAGER_ROLE, account);
100       }
101
102       // Function for the admin to adjust the maximum size of activeLoanIds
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_27	/NFTLendAuction.sol	L103 - L106	*Pending Fix*

/NFTLendAuction.sol L103 - L106

```
102       // Function for the admin to adjust the maximum size of activeLoanIds
103       function setMaxActiveLoans(uint256 newMax) external onlyRole(OWNER_ROLE) {
104           require(newMax > 0, "Max active loans must be greater than zero");
105           maxActiveLoans = newMax;
106       }
107
108       /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_28	/NFTLendAuction.sol	L119 - L121	*False Positive*

/NFTLendAuction.sol L119 - L121

```
118       // Admin can revoke roles from other addresses
119       function revokeManagerRole(address account) external onlyRole(DEFAULT_ADMIN_ROLE) {
120           revokeRole(MANAGER_ROLE, account);
121       }
122
123       /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_29	/NFTLendAuction.sol	L391 - L403	⚠ *Pending Fix*

/NFTLendAuction.sol L391 - L403

```
390         */
391        function _removeActiveLoan(uint256 loanId) private {
392            delete activeLoans[loanId];
393            for (uint256 i = 0; i < activeLoanIds.length; ) {
394                if (activeLoanIds[i] == loanId) {
395                    activeLoanIds[i] = activeLoanIds[activeLoanIds.length - 1];
396                    activeLoanIds.pop();
397                    break;
398                }
399                unchecked {
400                    i++;
401                }
402            }
403        }
404
405        /**
```

Issue Type

BLOCK VALUES AS A PROXY FOR TIME

S. No.	Severity	Detection Method	Instances
I001	Informational	Automated	3

Description

Contracts often need access to time values to perform certain types of functionality. Values such as `block.timestamp` and `block.number` can be used to determine the current time or the time delta. However, they are not recommended for most use cases.

For `block.number`, as Ethereum block times are generally around 14 seconds, the delta between blocks can be predicted. The block times, on the other hand, do not remain constant and are subject to change for a number of reasons, e.g., fork reorganizations and the difficulty bomb.

Due to variable block times, `block.number` should not be relied on for precise calculations of time.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_30	/NFTLendAuction.sol	L247 - L247	*Pending Fix*

/NFTLendAuction.sol L247 - L247

```
246         loan.isAccepted = true;
247         loan.startTime = block.timestamp;
248
249         uint256 loanAmount = escrowedFunds[loanId];
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_31	/NFTLendAuction.sol	L293 - L293	*Pending Fix*

/NFTLendAuction.sol L293 - L293

```
292           require(loan.isAccepted, "Loan not accepted yet");
293           require(block.timestamp <= loan.startTime + loan.duration, "Loan duration expire
      d");
294
295           uint256 requiredPayment = getRequiredRepayment(loanId);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_32	/NFTLendAuction.sol	L354 - L354	*Pending Fix*

/NFTLendAuction.sol L354 - L354

```
353           require(loan.isAccepted, "Loan not accepted");
354           require(block.timestamp > loan.startTime + loan.duration, "Loan not expired");
355
356           // Calculate the hypothetical repayment amount
```

Issue Type

MISSING INDEXED KEYWORDS IN EVENTS

S. No.	Severity	Detection Method	Instances
I002	Informational	Automated	9

Description

Events are essential for tracking off-chain data and when the event paraemters are `indexed` they can be used as fil ter options which will help getting only the specific data instead of all the logs.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_45	/NFTLendAuction.sol	L45 - L45	*Pending Fix*

/NFTLendAuction.sol L45 - L45

```
44        // Events
45        event LoanListed(uint256 loanId, Loan loan);
46        event LoanDelisted(uint256 loanId, Loan loan);
47        event LoanBidPlaced(uint256 loanId, Loan loan);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_46	/NFTLendAuction.sol	L46 - L46	*Pending Fix*

/NFTLendAuction.sol L46 - L46

```
45        event LoanListed(uint256 loanId, Loan loan);
46        event LoanDelisted(uint256 loanId, Loan loan);
47        event LoanBidPlaced(uint256 loanId, Loan loan);
48        event LoanBidCancelled(uint256 loanId, Loan loan);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_47	/NFTLendAuction.sol	L47 - L47	*Pending Fix*

/NFTLendAuction.sol L47 - L47

```
46        event LoanDelisted(uint256 loanId, Loan loan);
47        event LoanBidPlaced(uint256 loanId, Loan loan);
48        event LoanBidCancelled(uint256 loanId, Loan loan);
49        event LoanAccepted(uint256 loanId, Loan loan);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_48	/NFTLendAuction.sol	L48 - L48	*Pending Fix*

/NFTLendAuction.sol L48 - L48

```
47        event LoanBidPlaced(uint256 loanId, Loan loan);
48        event LoanBidCancelled(uint256 loanId, Loan loan);
49        event LoanAccepted(uint256 loanId, Loan loan);
50        event LoanRepaid(uint256 loanId, Loan loan);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_49	/NFTLendAuction.sol	L49 - L49	*Pending Fix*

/NFTLendAuction.sol L49 - L49

```
48        event LoanBidCancelled(uint256 loanId, Loan loan);
49        event LoanAccepted(uint256 loanId, Loan loan);
50        event LoanRepaid(uint256 loanId, Loan loan);
51        event LoanDefaulted(uint256 loanId, Loan loan);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_50	/NFTLendAuction.sol	L50 - L50	*Pending Fix*

/NFTLendAuction.sol L50 - L50

```
49        event LoanAccepted(uint256 loanId, Loan loan);
50        event LoanRepaid(uint256 loanId, Loan loan);
51        event LoanDefaulted(uint256 loanId, Loan loan);
52        event AllowedNFTUpdated(address nftAddress, bool allowed);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_51	/NFTLendAuction.sol	L51 - L51	*Pending Fix*

/NFTLendAuction.sol L51 - L51

```
50        event LoanRepaid(uint256 loanId, Loan loan);
51        event LoanDefaulted(uint256 loanId, Loan loan);
52        event AllowedNFTUpdated(address nftAddress, bool allowed);
53        event ProtocolFeeRateUpdated(uint256 newFeeRate);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_52	/NFTLendAuction.sol	L52 - L52	*Pending Fix*

/NFTLendAuction.sol L52 - L52

```
51        event LoanDefaulted(uint256 loanId, Loan loan);
52        event AllowedNFTUpdated(address nftAddress, bool allowed);
53        event ProtocolFeeRateUpdated(uint256 newFeeRate);
54        event ProtocolFeesWithdrawn(address to, uint256 amount);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_53	/NFTLendAuction.sol	L54 - L54	*Pending Fix*

/NFTLendAuction.sol L54 - L54

```
53        event ProtocolFeeRateUpdated(uint256 newFeeRate);
54        event ProtocolFeesWithdrawn(address to, uint256 amount);
55
56        // Modifiers
```

Issue Type

MISSING PAYABLE IN CALL FUNCTION

S. No.	Severity	Detection Method	Instances
I003	Informational	Automated	2

Description

The contract is using a `.call()` method to make external calls along with passing some Ether as `msg.value`. Since the function is not marked as `payable`, the transaction might fail if the contract does not have ETH.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_63	/NFTLendAuction.sol	L415 - L415	*Pending Fix*

/NFTLendAuction.sol L415 - L415

```
414
415         (bool success, ) = to.call{value: amount}("");
416         require(success, "Withdrawal failed");
417
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_64	/NFTLendAuction.sol	L439 - L439	⚠ *Pending Fix*

/NFTLendAuction.sol L439 - L439

```
438         // Perform the withdrawal
439         (bool success, ) = to.call{value: withdrawableAmount}("");
440         require(success, "Ether transfer failed");
441
```

Issue Type

NAME MAPPING PARAMETERS

S. No.	Severity	Detection Method	Instances
I004	Informational	Automated	4

Description

After Solidity 0.8.18, a feature was introduced to name mapping parameters. This helps in defining a purpose for each mapping and makes the code more descriptive.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_9	/NFTLendAuction.sol	L32 - L32	*Pending Fix*

/NFTLendAuction.sol L32 - L32

```
31        uint256 public loanCounter; // Counter to track the total number of loans created
32        mapping(uint256 => Loan) public loans; // Mapping of loan IDs to loan details
33        mapping(uint256 => uint256) public escrowedFunds; // Mapping of loan IDs to escrowed l
      ender funds
34        mapping(address => bool) public allowedNFTs; // Tracks which NFT contracts are allowed
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_10	/NFTLendAuction.sol	L33 - L33	⚠ *Pending Fix*

/NFTLendAuction.sol L33 - L33

```
32        mapping(uint256 => Loan) public loans; // Mapping of loan IDs to loan details
33        mapping(uint256 => uint256) public escrowedFunds; // Mapping of loan IDs to escrowed l
      ender funds
34        mapping(address => bool) public allowedNFTs; // Tracks which NFT contracts are allowed
35
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_11	/NFTLendAuction.sol	L34 - L34	⚠ *Pending Fix*

/NFTLendAuction.sol L34 - L34

```
33        mapping(uint256 => uint256) public escrowedFunds; // Mapping of loan IDs to escrowed l
      ender funds
34        mapping(address => bool) public allowedNFTs; // Tracks which NFT contracts are allowed
35
36        uint256 public maxActiveLoans = 1000; // Default maximum size for active loans
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_12	/NFTLendAuction.sol	L38 - L38	⚠ *Pending Fix*

/NFTLendAuction.sol L38 - L38

```
37        uint256[] public activeLoanIds; // List of IDs for currently active loans
38        mapping(uint256 => bool) public activeLoans; // Tracks whether a loan ID is active
39
40        uint256 public protocolFeeRate = 200; // Protocol fee rate in basis points (5%)
```

Issue Type

REDUNDANT STATEMENTS

S. No.	Severity	Detection Method	Instances
I005	Informational	Automated	1

Description

The contract contains redundant statements where types or identifiers are declared but not used, leading to no actio n being performed with them. These statements do not generate any code and can be safely removed to clean up th e contract.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_21	/NFTLendAuction.sol	L393 - L402	*Pending Fix*

/NFTLendAuction.sol L393 - L402

```
392           delete activeLoans[loanId];
393           for (uint256 i = 0; i < activeLoanIds.length; ) {
394               if (activeLoanIds[i] == loanId) {
395                   activeLoanIds[i] = activeLoanIds[activeLoanIds.length - 1];
396                   activeLoanIds.pop();
397                   break;
398               }
399               unchecked {
400                   i++;
401               }
402           }
403       }
404
```

Issue Type

USE CALL INSTEAD OF TRANSFER OR SEND

S. No.	Severity	Detection Method	Instances
I006	Informational	Automated	5

Description

The contract was found to be using `transfer` or `send` function call. This is unsafe as `transfer` has hard coded gas budget and can fail if the user is a smart contract.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_56	/NFTLendAuction.sol	L197 - L197	*Pending Fix*

/NFTLendAuction.sol L197 - L197

```
196         if (loan.lender != address(0)) {
197             payable(loan.lender).transfer(escrowedFunds[loanId]);
198         }
199
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_57	/NFTLendAuction.sol	L219 - L219	*Pending Fix*

/NFTLendAuction.sol L219 - L219

```
218             escrowedFunds[loanId] = 0; // Clear escrow
219             payable(loan.lender).transfer(escrowAmount);
220         }
221
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_58	/NFTLendAuction.sol	L251 - L251	*Pending Fix*

/NFTLendAuction.sol L251 - L251

```
250         escrowedFunds[loanId] = 0;
251         payable(loan.borrower).transfer(loanAmount);
252
253         emit LoanAccepted(loanId, loan);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_59	/NFTLendAuction.sol	L315 - L315	*Pending Fix*

/NFTLendAuction.sol L315 - L315

```
314         // Transfer lender payout
315         payable(loan.lender).transfer(lenderPayout);
316
317         emit LoanRepaid(loanId, loan);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_60	/NFTLendAuction.sol	L332 - L332	*Pending Fix*

/NFTLendAuction.sol L332 - L332

```
331         escrowedFunds[loanId] = 0; // Clear escrow
332         payable(loan.lender).transfer(escrowAmount);
333
334         // Clear lender information
```

Issue Type

ARRAY LENGTH CACHING

S. No.	Severity	Detection Method	Instances
G001	Gas	Automated	2

Description

During each iteration of the loop, reading the length of the array uses more gas than is necessary. In the most favora ble scenario, in which the length is read from a memory variable, storing the array length in the stack can save about 3 gas per iteration. In the least favorable scenario, in which external calls are made during each iteration, the amount of gas wasted can be significant.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_40	/NFTLendAuction.sol	L393 - L402	*Pending Fix*

/NFTLendAuction.sol L393 - L402

```
392           delete activeLoans[loanId];
393           for (uint256 i = 0; i < activeLoanIds.length; ) {
394               if (activeLoanIds[i] == loanId) {
395                   activeLoanIds[i] = activeLoanIds[activeLoanIds.length - 1];
396                   activeLoanIds.pop();
397                   break;
398               }
399               unchecked {
400                   i++;
401               }
402           }
403       }
404
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_41	/NFTLendAuction.sol	L430 - L432	⚠ *Pending Fix*

/NFTLendAuction.sol L430 - L432

```
429         uint256 totalEscrowedFunds = 0;
430         for (uint256 i = 0; i < activeLoanIds.length; i++) {
431             totalEscrowedFunds += escrowedFunds[activeLoanIds[i]];
432         }
433
434         // Calculate withdrawable amount:
```

Issue Type

ASSIGNING TO STRUCTS CAN BE MORE EFFICIENT

S. No.	Severity	Detection Method	Instances
G002	Gas	Automated	1

Description

The contract is found to contain a struct with multiple variables defined in it. When a struct is assigned in a single op eration, Solidity may perform costly storage operations, which can be inefficient. This often results in increased gas costs due to multiple SLOAD and SSTORE operations happening at once

Bug ID	File Location	Line No.	Action Taken
SSP_40397_20	/NFTLendAuction.sol	L154 - L165	*Pending Fix*

/NFTLendAuction.sol L154 - L165

```
153         // Create a new loan
154         loans[loanCounter] = Loan({
155             borrower: msg.sender,
156             lender: address(0),
157             nftAddress: nftAddress,
158             tokenId: tokenId,
159             loanAmount: loanAmount,
160             maxInterestRate: maxInterestRate,
161             currentInterestRate: maxInterestRate,
162             duration: duration,
163             startTime: 0,
164             isAccepted: false
165         });
166
167         // Track active loan
```

Issue Type

AVOID RE-STORING VALUES

S. No.	Severity	Detection Method	Instances
G003	Gas	Automated	3

Description

The function is found to be allowing re-storing the value in the contract's state variable even when the old value is eq ual to the new value. This practice results in unnecessary gas consumption due to the `Gsreset` operation (2900 ga s), which could be avoided. If the old value and the new value are the same, not updating the storage would avoid thi s cost and could instead incur a `Gcoldsload` (2100 gas) or a `Gwarmaccess` (100 gas), potentially saving gas.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_17	/NFTLendAuction.sol	L103 - L106	*Pending Fix*

/NFTLendAuction.sol L103 - L106

```
102       // Function for the admin to adjust the maximum size of activeLoanIds
103       function setMaxActiveLoans(uint256 newMax) external onlyRole(OWNER_ROLE) {
104           require(newMax > 0, "Max active loans must be greater than zero");
105           maxActiveLoans = newMax;
106       }
107
108       /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_18	/NFTLendAuction.sol	L113 - L116	⚠ *Pending Fix*

/NFTLendAuction.sol L113 - L116

```
112         */
113         function updateAllowedNFT(address nftAddress, bool allowed) external onlyRole(MANAGER_
        ROLE) {
114             allowedNFTs[nftAddress] = allowed;
115             emit AllowedNFTUpdated(nftAddress, allowed);
116         }
117
118         // Admin can revoke roles from other addresses
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_19	/NFTLendAuction.sol	L127 - L131	⚠ *Pending Fix*

/NFTLendAuction.sol L127 - L131

```
126         */
127         function setProtocolFeeRate(uint256 newFeeRate) external onlyRole(OWNER_ROLE) {
128             require(newFeeRate <= 1000, "Fee rate too high"); // Max 10%
129             protocolFeeRate = newFeeRate;
130             emit ProtocolFeeRateUpdated(newFeeRate);
131         }
132
133         /**
```

Issue Type

CACHE ADDRESS(THIS) WHEN USED MORE THAN ONCE

S. No.	Severity	Detection Method	Instances
G004	Gas	Automated	5

Description

The repeated usage of `address(this)` within the contract could result in increased gas costs due to multiple exec utions of the same computation, potentially impacting efficiency and overall transaction expenses.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_13	/NFTLendAuction.sol	L151 - L151	*Pending Fix*

/NFTLendAuction.sol L151 - L151

```
150         // Transfer the NFT to the contract
151         IERC721(nftAddress).transferFrom(msg.sender, address(this), tokenId);
152
153         // Create a new loan
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_14	/NFTLendAuction.sol	L223 - L223	⚠ *Pending Fix*

/NFTLendAuction.sol L223 - L223

```
222          // Return the NFT to the borrower
223          IERC721(loan.nftAddress).safeTransferFrom(address(this), loan.borrower, loan.token
      Id);
224
225          // Clean up loan data
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_14	/NFTLendAuction.sol	L312 - L312	⚠ *Pending Fix*

/NFTLendAuction.sol L312 - L312

```
311          loan.isAccepted = false;
312          IERC721(loan.nftAddress).safeTransferFrom(address(this), loan.borrower, loan.token
      Id);
313
314          // Transfer lender payout
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_15	/NFTLendAuction.sol	L371 - L371	*Pending Fix*

/NFTLendAuction.sol L371 - L371

```
370           loan.isAccepted = false;
371           IERC721(loan.nftAddress).safeTransferFrom(address(this), loan.lender, loan.tokenI
      d);
372
373
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_16	/NFTLendAuction.sol	L426 - L426	*Pending Fix*

/NFTLendAuction.sol L426 - L426

```
425           // Calculate the total balance held by the contract
426           uint256 totalBalance = address(this).balance;
427
428           // Calculate total escrowed funds (sum of all values in escrowedFunds mapping)
```

Issue Type

CHEAPER CONDITIONAL OPERATORS

S. No.	Severity	Detection Method	Instances
G005	Gas	Automated	2

Description

During compilation, `x != 0` is cheaper than `x > 0` for unsigned integers in solidity inside conditional statements.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_54	/NFTLendAuction.sol	L104 - L104	*Pending Fix*

/NFTLendAuction.sol L104 - L104

```
103       function setMaxActiveLoans(uint256 newMax) external onlyRole(OWNER_ROLE) {
104           require(newMax > 0, "Max active loans must be greater than zero");
105           maxActiveLoans = newMax;
106       }
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_55	/NFTLendAuction.sol	L436 - L436	⚠ *Pending Fix*

/NFTLendAuction.sol L436 - L436

```
435           uint256 withdrawableAmount = totalBalance - totalEscrowedFunds - protocolFeeBalanc
      e;
436           require(withdrawableAmount > 0, "No funds available for withdrawal");
437
438           // Perform the withdrawal
```

Issue Type

CHEAPER INEQUALITIES IN REQUIRE()

S. No.	Severity	Detection Method	Instances
G006	Gas	Automated	3

Description

The contract was found to be performing comparisons using inequalities inside the `require` statement. When inside the `require` statements, non-strict inequalities `(>=, <=)` are usually costlier than strict equalities `(>, <)`.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_33	/NFTLendAuction.sol	L128 - L128	*Pending Fix*

/NFTLendAuction.sol L128 - L128

```
127       function setProtocolFeeRate(uint256 newFeeRate) external onlyRole(OWNER_ROLE) {
128           require(newFeeRate <= 1000, "Fee rate too high"); // Max 10%
129           protocolFeeRate = newFeeRate;
130           emit ProtocolFeeRateUpdated(newFeeRate);
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_34	/NFTLendAuction.sol	L190 - L190	*Pending Fix*

/NFTLendAuction.sol L190 - L190

```
189           require(
190               interestRate < loan.currentInterestRate && interestRate <= loan.maxInterestRat
      e,
191               "Bid interest rate invalid"
192           );
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_35	/NFTLendAuction.sol	L293 - L293	*Pending Fix*

/NFTLendAuction.sol L293 - L293

```
292           require(loan.isAccepted, "Loan not accepted yet");
293           require(block.timestamp <= loan.startTime + loan.duration, "Loan duration expire
      d");
294
295           uint256 requiredPayment = getRequiredRepayment(loanId);
```

Issue Type

DEFINE CONSTRUCTOR AS PAYABLE

S. No.	Severity	Detection Method	Instances
G007	Gas	Automated	1

Description

Developers can save around 10 opcodes and some gas if the constructors are defined as payable.
However, it should be noted that it comes with risks because payable constructors can accept ETH during deployme nt.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_2	/NFTLendAuction.sol	L90 - L95	*Pending Fix*

/NFTLendAuction.sol L90 - L95

```
89          */
90         constructor() {
91             // Grant the initial owner the DEFAULT_ADMIN_ROLE and OWNER_ROLE
92             _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);
93             _grantRole(OWNER_ROLE, msg.sender);
94             _grantRole(MANAGER_ROLE, msg.sender);
95         }
96
97         // Admin can grant roles to other addresses
```

Issue Type

GAS OPTIMIZATION FOR STATE VARIABLES

S. No.	Severity	Detection Method	Instances
G008	Gas	Automated	2

Description

Plus equals (`+=`) costs more gas than addition operator. The same thing happens with minus equals (`-=`).

Bug ID	File Location	Line No.	Action Taken
SSP_40397_38	/NFTLendAuction.sol	L308 - L308	*Pending Fix*

/NFTLendAuction.sol L308 - L308

```
307         uint256 borrowerProtocolFee = (totalRepayment * protocolFeeRate) / 10000;
308         protocolFeeBalance += (borrowerProtocolFee + lenderProtocolFee);
309
310         // Transfer NFT back to borrower
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_39	/NFTLendAuction.sol	L367 - L367	⚠ *Pending Fix*

/NFTLendAuction.sol L367 - L367

```
366         // Update protocol fee balance
367         protocolFeeBalance += lenderProtocolFee;
368
369         // Transfer NFT to the lender
```

Issue Type

GAS OPTIMIZATION IN INCREMENTS

S. No.	Severity	Detection Method	Instances
G009	Gas	Automated	1

Description

`++i` costs less gas compared to `i++` or `i += 1` for unsigned integers. In `i++`, the compiler has to create a temporary variable to store the initial value. This is not the case with `++i` in which the value is directly incremented and returned, thus, making it a cheaper alternative.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_22	/NFTLendAuction.sol	L430 - L430	*Pending Fix*

/NFTLendAuction.sol L430 - L430

```
429         uint256 totalEscrowedFunds = 0;
430         for (uint256 i = 0; i < activeLoanIds.length; i++) {
431             totalEscrowedFunds += escrowedFunds[activeLoanIds[i]];
432         }
```

Issue Type

LONG REQUIRE/REVERT STRINGS

S. No.	Severity	Detection Method	Instances
G010	Gas	Automated	2

Description

The `require()` and `revert()` functions take an input string to show errors if the validation fails.
This strings inside these functions that are longer than `32 bytes` require at least one additional `MSTORE`, along wit
h additional overhead for computing memory offset, and other parameters.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_43	/NFTLendAuction.sol	L104 - L104	*Pending Fix*

/NFTLendAuction.sol L104 - L104

```
103       function setMaxActiveLoans(uint256 newMax) external onlyRole(OWNER_ROLE) {
104           require(newMax > 0, "Max active loans must be greater than zero");
105           maxActiveLoans = newMax;
106       }
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_44	/NFTLendAuction.sol	L436 - L436	⚠ *Pending Fix*

/NFTLendAuction.sol L436 - L436

```
435           uint256 withdrawableAmount = totalBalance - totalEscrowedFunds - protocolFeeBalanc
      e;
436           require(withdrawableAmount > 0, "No funds available for withdrawal");
437
438           // Perform the withdrawal
```

Issue Type

PUBLIC CONSTANTS CAN BE PRIVATE

S. No.	Severity	Detection Method	Instances
G011	Gas	Automated	2

Description

Public constant variables cost more gas because the EVM automatically creates getter functions for them and adds entries to the method ID table. The values can be read from the source code instead.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_36	/NFTLendAuction.sol	L15 - L15	*Pending Fix*

/NFTLendAuction.sol L15 - L15

```
14    contract NFTLendAuction is ReentrancyGuard, AccessControl {
15        bytes32 public constant OWNER_ROLE = keccak256("OWNER_ROLE");
16        bytes32 public constant MANAGER_ROLE = keccak256("MANAGER_ROLE");
17
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_37	/NFTLendAuction.sol	L16 - L16	⚠ *Pending Fix*

/NFTLendAuction.sol L16 - L16

```
15        bytes32 public constant OWNER_ROLE = keccak256("OWNER_ROLE");
16        bytes32 public constant MANAGER_ROLE = keccak256("MANAGER_ROLE");
17
18        struct Loan {
```

Issue Type

SPLITTING REQUIRE STATEMENTS

S. No.	Severity	Detection Method	Instances
G012	Gas	Automated	1

Description

Require statements when combined using operators in a single statement usually lead to a larger deployment gas cost but with each runtime calls, the whole thing ends up being cheaper by some gas units.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_1	/NFTLendAuction.sol	L189 - L192	*Pending Fix*

/NFTLendAuction.sol L189 - L192

```
188         Loan storage loan = loans[loanId];
189         require(
190             interestRate < loan.currentInterestRate && interestRate <= loan.maxInterestRat
    e,
191             "Bid interest rate invalid"
192         );
193         require(msg.value == loan.loanAmount, "Incorrect loan amount");
194
```

Issue Type

STORAGE VARIABLE CACHING IN MEMORY

S. No.	Severity	Detection Method	Instances
G013	Gas	Automated	7

Description

The contract is using the state variable multiple times in the function.
`SLOADs` are expensive (100 gas after the 1st one) compared to `MLOAD` / `MSTORE` (3 gas each).

Bug ID	File Location	Line No.	Action Taken
SSP_40397_3	/NFTLendAuction.sol	L141 - L174	*Pending Fix*

/NFTLendAuction.sol L141 - L174

```
140         */
141        function listLoan(
142            address nftAddress,
143            uint256 tokenId,
144            uint256 loanAmount,
145            uint256 maxInterestRate,
146            uint256 duration
147        ) external nonReentrant isAllowedNFT(nftAddress) onlyNftOwner(nftAddress, tokenId) {
148            require(activeLoanIds.length < maxActiveLoans, "Active loan limit reached");
149
150            // Transfer the NFT to the contract
151            IERC721(nftAddress).transferFrom(msg.sender, address(this), tokenId);
152
153            // Create a new loan
154            loans[loanCounter] = Loan({
155                borrower: msg.sender,
156                lender: address(0),
157                nftAddress: nftAddress,
158                tokenId: tokenId,
159                loanAmount: loanAmount,
160                maxInterestRate: maxInterestRate,
161                currentInterestRate: maxInterestRate,
```

/NFTLendAuction.sol L141 - L174

```
160                 maxInterestRate: maxInterestRate,
161                 currentInterestRate: maxInterestRate,
162                 duration: duration,
163                 startTime: 0,
164                 isAccepted: false
165             });
166
167             // Track active loan
168             activeLoans[loanCounter] = true;
169             activeLoanIds.push(loanCounter);
170
171             emit LoanListed(loanCounter, loans[loanCounter]);
172
173             loanCounter++;
174         }
175
176         /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_3	/NFTLendAuction.sol	L141 - L174	⚠ *Pending Fix*

/NFTLendAuction.sol L141 - L174

```
140         */
141        function listLoan(
142            address nftAddress,
143            uint256 tokenId,
144            uint256 loanAmount,
145            uint256 maxInterestRate,
146            uint256 duration
147        ) external nonReentrant isAllowedNFT(nftAddress) onlyNftOwner(nftAddress, tokenId) {
148            require(activeLoanIds.length < maxActiveLoans, "Active loan limit reached");
149
150            // Transfer the NFT to the contract
151            IERC721(nftAddress).transferFrom(msg.sender, address(this), tokenId);
152
153            // Create a new loan
154            loans[loanCounter] = Loan({
155                borrower: msg.sender,
156                lender: address(0),
157                nftAddress: nftAddress,
158                tokenId: tokenId,
159                loanAmount: loanAmount,
160                maxInterestRate: maxInterestRate,
161                currentInterestRate: maxInterestRate,
162                duration: duration,
```

/NFTLendAuction.sol L141 - L174

```
160             maxInterestRate: maxInterestRate,
161             currentInterestRate: maxInterestRate,
162             duration: duration,
163             startTime: 0,
164             isAccepted: false
165         });
166
167         // Track active loan
168         activeLoans[loanCounter] = true;
169         activeLoanIds.push(loanCounter);
170
171         emit LoanListed(loanCounter, loans[loanCounter]);
172
173         loanCounter++;
174     }
175
176     /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_4	/NFTLendAuction.sol	L212 - L230	⚠ *Pending Fix*

/NFTLendAuction.sol L212 - L230

```
211         */
212         function delistLoan(uint256 loanId) external nonReentrant loanExists(loanId) isNotAcce
    pted(loanId) onlyBorrower(loanId) {
213             Loan storage loan = loans[loanId];
214 
215             // Refund escrowed funds to the last bidder (if any)
216             if (loan.lender != address(0)) {
217                 uint256 escrowAmount = escrowedFunds[loanId];
218                 escrowedFunds[loanId] = 0; // Clear escrow
219                 payable(loan.lender).transfer(escrowAmount);
220             }
221 
222             // Return the NFT to the borrower
223             IERC721(loan.nftAddress).safeTransferFrom(address(this), loan.borrower, loan.token
    Id);
224 
225             // Clean up loan data
226             delete loans[loanId];
227             _removeActiveLoan(loanId);
228 
229             emit LoanDelisted(loanId, loan);
230         }
231 
232         /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_4	/NFTLendAuction.sol	L212 - L230	⚠ *Pending Fix*

/NFTLendAuction.sol L212 - L230

```
211         */
212         function delistLoan(uint256 loanId) external nonReentrant loanExists(loanId) isNotAcce
    pted(loanId) onlyBorrower(loanId) {
213             Loan storage loan = loans[loanId];
214 
215             // Refund escrowed funds to the last bidder (if any)
216             if (loan.lender != address(0)) {
217                 uint256 escrowAmount = escrowedFunds[loanId];
218                 escrowedFunds[loanId] = 0; // Clear escrow
219                 payable(loan.lender).transfer(escrowAmount);
220             }
221 
222             // Return the NFT to the borrower
223             IERC721(loan.nftAddress).safeTransferFrom(address(this), loan.borrower, loan.token
    Id);
224 
225             // Clean up loan data
226             delete loans[loanId];
227             _removeActiveLoan(loanId);
228 
229             emit LoanDelisted(loanId, loan);
230         }
231 
232         /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_5	/NFTLendAuction.sol	L284 - L320	⚠ *Pending Fix*

/NFTLendAuction.sol L284 - L320

```
283         */
284       function repayLoan(uint256 loanId)
285           external
286           payable
287           nonReentrant
288           loanExists(loanId)
289           onlyBorrower(loanId)
290       {
291           Loan storage loan = loans[loanId];
292           require(loan.isAccepted, "Loan not accepted yet");
293           require(block.timestamp <= loan.startTime + loan.duration, "Loan duration expire
      d");
294
295           uint256 requiredPayment = getRequiredRepayment(loanId);
296           require(msg.value == requiredPayment, "Incorrect repayment amount");
297
298           // Calculate total repayment: principal + interest
299           uint256 interestAmount = (loan.loanAmount * loan.currentInterestRate) / 10000;
300           uint256 totalRepayment = loan.loanAmount + interestAmount;
301
302           // Calculate lender protocol fee
303           uint256 lenderProtocolFee = (totalRepayment * protocolFeeRate) / 10000;
304           uint256 lenderPayout = totalRepayment - lenderProtocolFee;
305
```

/NFTLendAuction.sol L284 - L320

```
303             uint256 lenderProtocolFee = (totalRepayment * protocolFeeRate) / 10000;
304             uint256 lenderPayout = totalRepayment - lenderProtocolFee;
305
306             // Update protocol fee balance
307             uint256 borrowerProtocolFee = (totalRepayment * protocolFeeRate) / 10000;
308             protocolFeeBalance += (borrowerProtocolFee + lenderProtocolFee);
309
310             // Transfer NFT back to borrower
311             loan.isAccepted = false;
312             IERC721(loan.nftAddress).safeTransferFrom(address(this), loan.borrower, loan.tokenId);
313
314             // Transfer lender payout
315             payable(loan.lender).transfer(lenderPayout);
316
317             emit LoanRepaid(loanId, loan);
318
319             _removeActiveLoan(loanId);
320         }
321
322         /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_6	/NFTLendAuction.sol	L391 - L403	*Pending Fix*

/NFTLendAuction.sol L391 - L403

```
390         */
391        function _removeActiveLoan(uint256 loanId) private {
392            delete activeLoans[loanId];
393            for (uint256 i = 0; i < activeLoanIds.length; ) {
394                if (activeLoanIds[i] == loanId) {
395                    activeLoanIds[i] = activeLoanIds[activeLoanIds.length - 1];
396                    activeLoanIds.pop();
397                    break;
398                }
399                unchecked {
400                    i++;
401                }
402            }
403        }
404
405        /**
```

Bug ID	File Location	Line No.	Action Taken
SSP_40397_7	/NFTLendAuction.sol	L422 - L443	⚠ *Pending Fix*

/NFTLendAuction.sol L422 - L443

```
421       // Withdraw locked Ether sent to the contract by mistake
422       function withdrawEther(address payable to) external nonReentrant onlyRole(OWNER_ROLE)
      {
423           require(to != address(0), "Invalid recipient address");
424
425           // Calculate the total balance held by the contract
426           uint256 totalBalance = address(this).balance;
427
428           // Calculate total escrowed funds (sum of all values in escrowedFunds mapping)
429           uint256 totalEscrowedFunds = 0;
430           for (uint256 i = 0; i < activeLoanIds.length; i++) {
431               totalEscrowedFunds += escrowedFunds[activeLoanIds[i]];
432           }
433
434           // Calculate withdrawable amount:
435           uint256 withdrawableAmount = totalBalance - totalEscrowedFunds - protocolFeeBalanc
      e;
436           require(withdrawableAmount > 0, "No funds available for withdrawal");
437
438           // Perform the withdrawal
439           (bool success, ) = to.call{value: withdrawableAmount}("");
440           require(success, "Ether transfer failed");
441
442           emit ProtocolFeesWithdrawn(to, withdrawableAmount);
443       }
```

Issue Type

USE SELFBALANCE() INSTEAD OF ADDRESS(THIS).BALANCE

S. No.	Severity	Detection Method	Instances
G014	Gas	Automated	1

Description

In Solidity, efficient use of gas is paramount to ensure cost-effective execution on the Ethereum blockchain. Gas can be optimized when obtaining contract balance by using `selfbalance()` rather than `address(this).balance` because it bypasses gas costs and refunds, which are not required for obtaining the contract's balance.

Bug ID	File Location	Line No.	Action Taken
SSP_40397_23	/NFTLendAuction.sol	L426 - L426	*Pending Fix*

/NFTLendAuction.sol L426 - L426

```
425         // Calculate the total balance held by the contract
426         uint256 totalBalance = address(this).balance;
427
428         // Calculate total escrowed funds (sum of all values in escrowedFunds mapping)
```

05. Scan History

Critical High Medium Low Informational Gas

No	Date	Security Score	Scan Overview
1.	2024-12-08	83.80	2 1 0 2 24 30

06. Disclaimer

The Reports neither endorse nor condemn any specific project or team, nor do they guarantee the security of any specific project. The contents of this report do not, and should not be interpreted as having any bearing on, the economics of tokens, token sales, or any other goods, services, or assets.

The security audit is not meant to replace functional testing done before a software release.

There is no warranty that all possible security issues of a particular smart contract(s) will be found by the tool, i.e., It is not guaranteed that there will not be any further findings based solely on the results of this evaluation.

Emerging technologies such as Smart Contracts and Solidity carry a high level of technical risk and uncertainty. There is no warranty or representation made by this report to any Third Party in regards to the quality of code, the business model or the proprietors of any such business model, or the legal compliance of any business.

In no way should a third party use these reports to make any decisions about buying or selling a token, product, service, or any other asset. It should be noted that this report is not investment advice, is not intended to be relied on as investment advice, and has no endorsement of this project or team. It does not serve as a guarantee as to the project's absolute security.

The assessment provided by SolidityScan 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. SolidityScan owes no duty to any third party by virtue of publishing these Reports.

As one audit-based assessment cannot be considered comprehensive, we always recommend proceeding with several independent manual audits including manual audit and a public bug bounty program to ensure the security of the smart contracts.