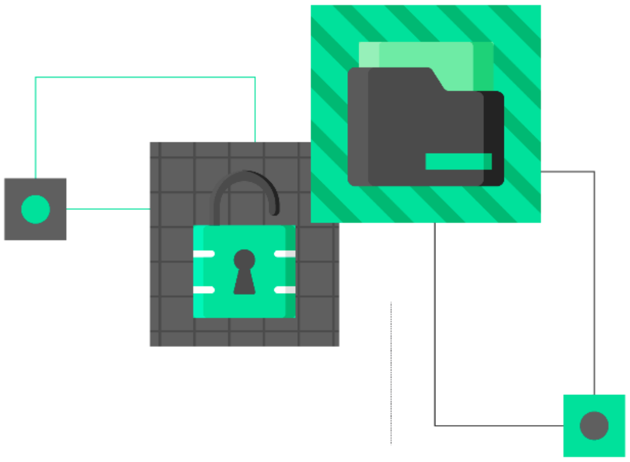
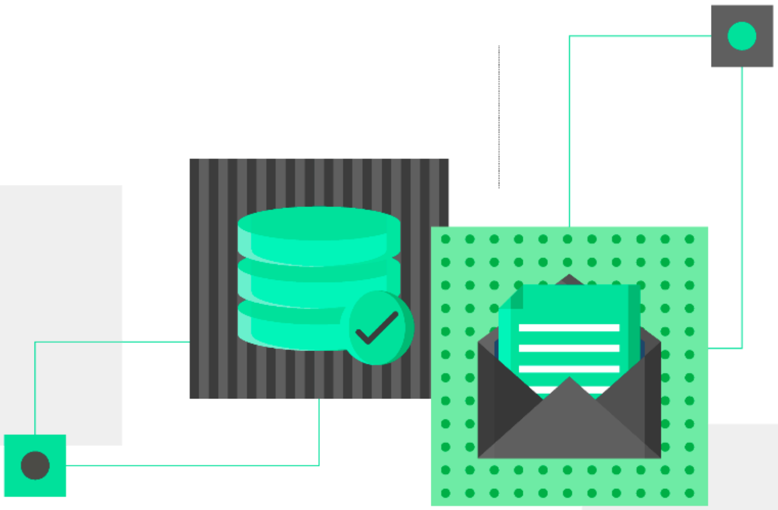
Mantisec Labs

Smart Contract Audit

NFTLending.sol

ethernity.io





Jan 2024

Mantisec Labs

Contents

Disclaimer

Audit Process & Methodology

The Mantisec Labs team carried out a thorough audit for the project, starting with an in-depth analysis of code design patterns. This initial step ensured the smart contract's architecture was well-structured and securely integrated with third-party smart contracts and libraries.Also, our team conducted a thorough line-by-line inspection of the smart contract,seeking out potential issues such as Signature Replay Attacks, Unchecked External Calls, External Contract Referencing, Variable Shadowing, Race conditions, Transaction-ordering dependence, timestamp dependence, DoS attacks, among others.
During the Unit testing phase, we assessed the functions authored by the developer to ascertain their precise functionality. Our Automated Testing procedures leveraged proprietary tools designed in-house to spot vulnerabilities and security flaws within the Smart Contract. The code was subjected to an in-depth audit administered by an independent team of auditors, encompassing the following critical aspects:

- Scrutiny of the smart contract's structural analysis to verify its integrity.
- Extensive automated testing of the contract
- A manual line-by-line Code review, undertaken with the aim of evaluating, analyzing, and identifying potential security risks.
- An evaluation of the contract's intended behavior, encompassing a review of provided documentation to ensure the contract conformed to expectations.
- Rigorous verification of storage layout in upgradeable contracts.
- An integral component of the audit procedure involved the identification and recommendation of enhanced gas optimization techniques for the contract

Audit Purpose

Mantisec Labs was hired by the Ethernity team to review their smart contract.
This audit was conducted in **Jan 2024**.

The main reasons for this review were:

- To find any possible security issues in the smart contract.
- To carefully check the logic behind the given smart contract.

This report provides valuable information for assessing the level of risk associated with this smart contract and offers suggestions on how to improve its security by addressing any identified issues.

Contract Details

Project Name	ethernity.io
Contract link	Private
Language	Solidity
Type	ERC20

Mantisec Labs

Security Level Reference

Issues	Critical	High	Medium
Open	1	2	0
Closed	1	2	0

Issues	Low	Informational	Gas Opt
Open	2	1	5
Closed	2	1	5

Findings

Contract: NFTLending.sol

G001 - Don't Initialize Variables with Default Value

If a variable is not set/initialized, it is assumed to have the default value (0, false, 0x0 etc depending on the data type). If you explicitly initialize it with its default value, you are just wasting gas.

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::63
~/nft-lending-contract-main/contracts/NFTLending.sol::295
~/nft-lending-contract-main/contracts/NFTLending.sol::382
~/nft-lending-contract-main/contracts/NFTLending.sol::442
~/nft-lending-contract-main/contracts/NFTLending.sol::476

G002 - Cache Array Length Outside of Loop

Reading array length at each iteration of the loop takes 6 gas (3 for mload and 3 to place memory_offset) in the stack.

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::295
~/nft-lending-contract-main/contracts/NFTLending.sol::382
~/nft-lending-contract-main/contracts/NFTLending.sol::442
~/nft-lending-contract-main/contracts/NFTLending.sol::476

G003 - Use != 0 instead of > 0 for Unsigned Integer Comparison

When dealing with unsigned integer types, comparisons with != 0 are cheaper than with > 0

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::278
~/nft-lending-contract-main/contracts/NFTLending.sol::279
~/nft-lending-contract-main/contracts/NFTLending.sol::280
~/nft-lending-contract-main/contracts/NFTLending.sol::281
~/nft-lending-contract-main/contracts/NFTLending.sol::297
~/nft-lending-contract-main/contracts/NFTLending.sol::401

Mantisec Labs

G004 - Remove Unused Variable

Delete unused variables to reduce the deployment costs.

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::62

G005 - ++eventId Costs Less Gas Compared to eventId++ or eventId += 1

++eventId costs less gas compared to eventId += 1 for unsigned integers, as pre-increment is cheaper (about 11 gas less than i += 1). This statement is true even with the optimizer enabled.

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::129
~/nft-lending-contract-main/contracts/NFTLending.sol::143
~/nft-lending-contract-main/contracts/NFTLending.sol::164
~/nft-lending-contract-main/contracts/NFTLending.sol::176
~/nft-lending-contract-main/contracts/NFTLending.sol::181
~/nft-lending-contract-main/contracts/NFTLending.sol::195
~/nft-lending-contract-main/contracts/NFTLending.sol::211
~/nft-lending-contract-main/contracts/NFTLending.sol::216

L001 - Unsafe ERC20 Operation(s)

ERC20 operations can be unsafe due to different implementations and vulnerabilities in the standard.
It is therefore recommended to always either use OpenZeppelin's SafeERC20 library.

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::379
~/nft-lending-contract-main/contracts/NFTLending.sol::397
~/nft-lending-contract-main/contracts/NFTLending.sol::433
~/nft-lending-contract-main/contracts/NFTLending.sol::439
~/nft-lending-contract-main/contracts/NFTLending.sol::447
~/nft-lending-contract-main/contracts/NFTLending.sol::481

L002 - Add Zero Address Validation

Parameter used to set the state variable, error in these can lead to contract misbehave.

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::119
~/nft-lending-contract-main/contracts/NFTLending.sol::224
~/nft-lending-contract-main/contracts/NFTLending.sol::238
~/nft-lending-contract-main/contracts/NFTLending.sol::255

I001-Use Of Deprecated Setuprole Function

The deprecated function _setupRole from the AccessControl contract. As per the AccessControl.sol contract documentation, this function is deprecated:

https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/access/AccessControl.sol#L183

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::121

H001 - Lack of Authentication for Offer Cancellation in the cancelOffer Function

The cancelOffer function does not enforce any authentication check to ensure that only the creator of the offer can cancel it.
This oversight may allow unauthorized users to cancel loan offers created by others, potentially leading to disruption and misuse of the lending system.

H002 - TransferFrom from address(this) will not work for most of the tokens

Here nft721.transferFrom(msg.sender, address(this), nftTokenId); is used which will not work with most of the tokens without calling approve.

We recommend using SafeTransfer instead of TransferFrom.

Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::397

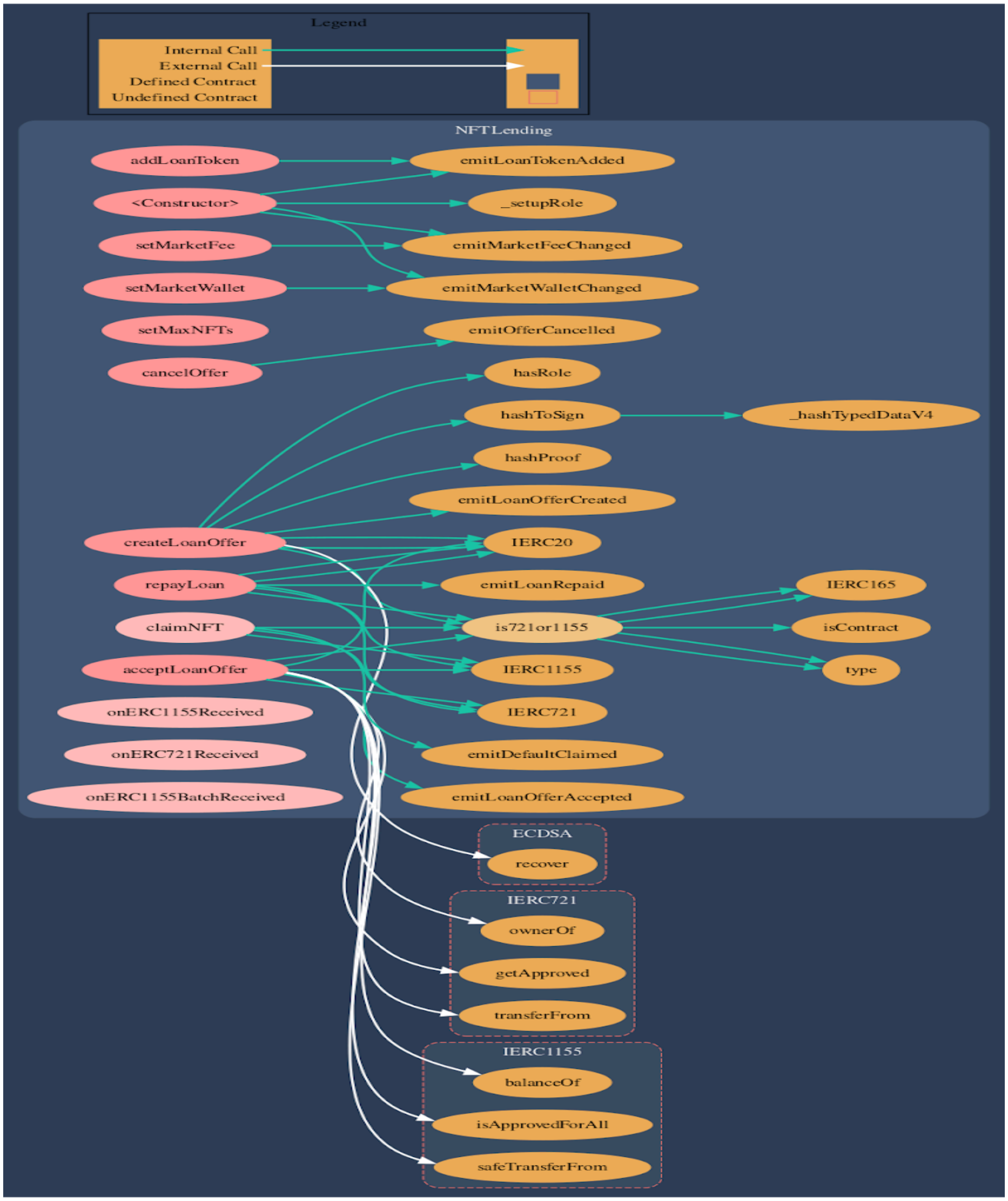
Code Location:
~/nft-lending-contract-main/contracts/NFTLending.sol::397

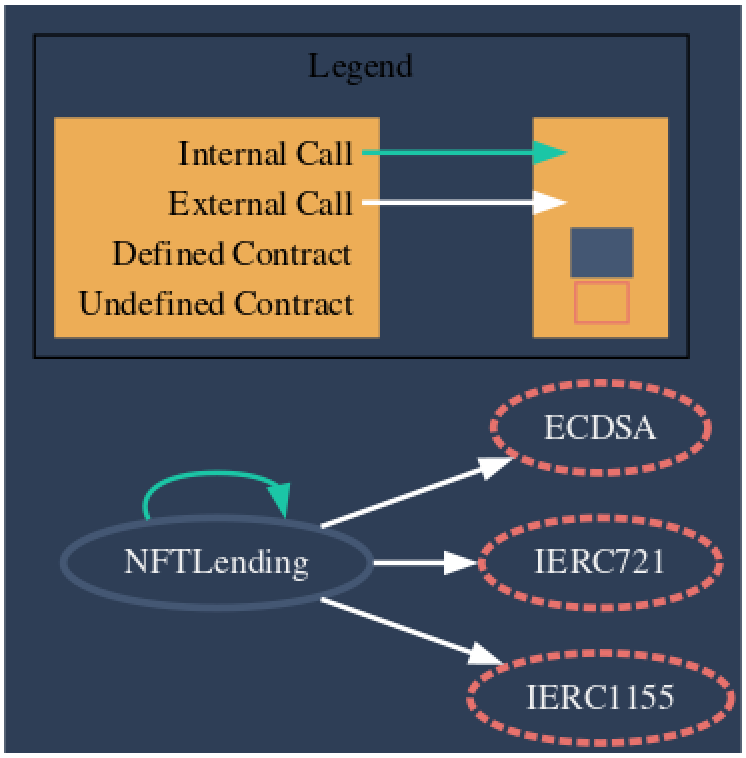
C001 - Transfer of ERC20 tokens from the lender to the borrower occurs before all the necessary validations

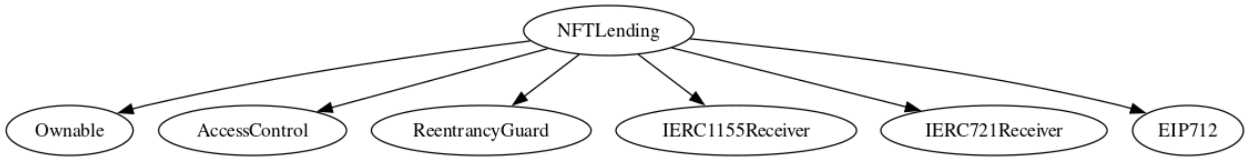
The transfer of ERC20 tokens from the lender to the borrower before completing critical validations, including NFT ownership checks. This order of operations introduces a security vulnerability, as a failure in subsequent checks does not trigger a proper reversion of state changes. An attacker could exploit this flaw by initiating a loan offer, triggering an ERC20 transfer, and then deliberately failing NFT ownership checks, resulting in an unauthorized token transfer.

Mantisec Labs

Additional Details




Legend
Internal Call
External Call
Defined Contract
Undefined Contract
ECDSA
NFTLending
IERC721
IERC1155


NFTLending
Ownable
AccessControl
ReentrancyGuard
IERC1155Receiver
IERC721Receiver
EIP712

Concluding Remarks

To wrap it up, this NFTLending.sol audit has given us a good look at the contract's security and functionality.
We found some Critical,High and low severity issues.
Our auditors confirmed that all the issues are now resolved by the developers.