



NFTfi - Ethereum

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **March 7th, 2022 – March 25th, 2022**

Visit: Halborn.com

DOCUMENT REVISION HISTORY	3
CONTACTS	3
1 EXECUTIVE OVERVIEW	4
1.1 INTRODUCTION	5
1.2 AUDIT SUMMARY	5
1.3 TEST APPROACH & METHODOLOGY	5
RISK METHODOLOGY	6
1.4 SCOPE	8
2 ASSESSMENT SUMMARY & FINDINGS OVERVIEW	10
3 FINDINGS & TECH DETAILS	11
3.1 (HAL-01) CAUSE-EFFECT CHECKS – INFORMATIONAL	13
Description	13
Risk Level	13
Recommendation	13
Remediation Plan	13
4 MANUAL ANALYSIS	14
4.1 NftfiHub	16
4.2 DirectLoanCoordinator	16
4.3 PermittedAirdrop	16
4.4 PermittedNFTsAndTypeRegistry	17
4.5 nftTypeRegistry/nftTypes	17
4.6 SmartNFT	17
5 CALL GRAPH AND INHERITANCE	18
AirdropFlashLoan	20

AirdropReceiver	21
DirectLoanBaseMinimal	22
DirectLoanFixedOffer	23
DirectLoanCoordinator	24
SmartNFT	25
NFTfiBundler	26
6 AUTOMATED TESTING	27
6.1 STATIC ANALYSIS REPORT	28
Description	28
Results	28

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	03/23/2022	Ferran.Celades
0.2	Document Edits	03/28/2022	Ferran.Celades
0.3	Draft Review	03/28/2022	Gabi Urrutia
1.0	Remediation Plan	03/29/2022	Ferran Celades
1.1	Remediation Plan Review	03/30/2022	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Ferran Celades	Halborn	Ferran.Celades@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

NFTfi engaged Halborn to conduct a security audit on their smart contracts beginning on March 7th, 2022 and ending on March 25th, 2022 . The security assessment was scoped to the smart contracts provided to the Halborn team.

1.2 AUDIT SUMMARY

The team at Halborn was provided two weeks for the engagement and assigned a full-time security engineer to audit the security of the smart contract. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that smart contract functions operate as intended
- Identify potential security issues with the smart contracts

In summary, Halborn identified a security risk that was acknowledged by the NFTfi team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the NFTfi develop-v2.1-audit branch and Pool V5 contract solidity code and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

- Research into architecture and purpose.
- Smart contract manual code review and walkthrough.
- Graphing out functionality and contract logic/connectivity/functions ([solgraph](#))
- Manual assessment of use and safety for the critical Solidity variables and functions in scope to identify any arithmetic related vulnerability classes.
- Manual testing by custom scripts .
- Scanning of solidity files for vulnerabilities, security hotspots or bugs. ([MythX](#))
- Static Analysis of security for scoped contract, and imported functions. ([Slither](#))
- Testnet deployment ([Remix IDE](#))

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

- 5 - Almost certain an incident will occur.
- 4 - High probability of an incident occurring.
- 3 - Potential of a security incident in the long term.
- 2 - Low probability of an incident occurring.
- 1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

- 5 - May cause devastating and unrecoverable impact or loss.
- 4 - May cause a significant level of impact or loss.

- 3 - May cause a partial impact or loss to many.
- 2 - May cause temporary impact or loss.
- 1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of **10** to **1** with **10** being the highest level of security risk.

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
----------	------	--------	-----	---------------

- 10** - CRITICAL
- 9** - **8** - HIGH
- 7** - **6** - MEDIUM
- 5** - **4** - LOW
- 3** - **1** - VERY LOW AND INFORMATIONAL

1.4 SCOPE

IN-SCOPE : NFTfi v2.2-07-03-2022-audit branch contracts

The security assessment was scoped to the following smart contract:

Scope-Commit-Id : 77d8476ff40e048dcca108265e863a3ebdfb261a

- contracts/NftfiHub.sol
- contracts/airdrop/AirdropFlashLoan.sol
- contracts/airdrop/AirdropReceiver.sol
- contracts/airdrop/AirdropReceiverFactory.sol
- contracts/airdrop/IAirdropReceiverFactory.sol
- contracts/composable/ERC9981155Extension.sol
- contracts/composable/ERC998ERC20Extension.sol
- contracts/composable/ERC998TopDown.sol
- contracts/composable/NftfiBundler.sol
- contracts/interfaces/*.sol
- contracts/loans/BaseLoan.sol
- contracts/loans/direct/DirectLoanCoordinator.sol
- contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol
- contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol
- contracts/loans/direct/loanTypes/IDirectLoanBase.sol
- contracts/loans/direct/loanTypes/LoanAirdropUtils.sol
- contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol
- contracts/loans/direct/loanTypes/LoanData.sol
- contracts/nftTypeRegistry/nftTypes/CryptoKittiesWrapper.sol
- contracts/nftTypeRegistry/nftTypes/ERC1155Wrapper.sol
- contracts/nftTypeRegistry/nftTypes/ERC721Wrapper.sol
- contracts/permitedLists/PermittedAirdrops.sol
- contracts/permitedLists/PermittedERC20s.sol
- contracts/permitedLists/PermittedNFTsAndTypeRegistry.sol
- contracts/permitedLists/PermittedPartners.sol
- contracts/smartNft/SmartNft.sol
- contracts/utils/ContractKeys.sol
- contracts/utils/NFTfiSigningUtils.sol
- contracts/utils/NFTfiSigningUtilsContract.sol
- contracts/utils/NftReceiver.sol

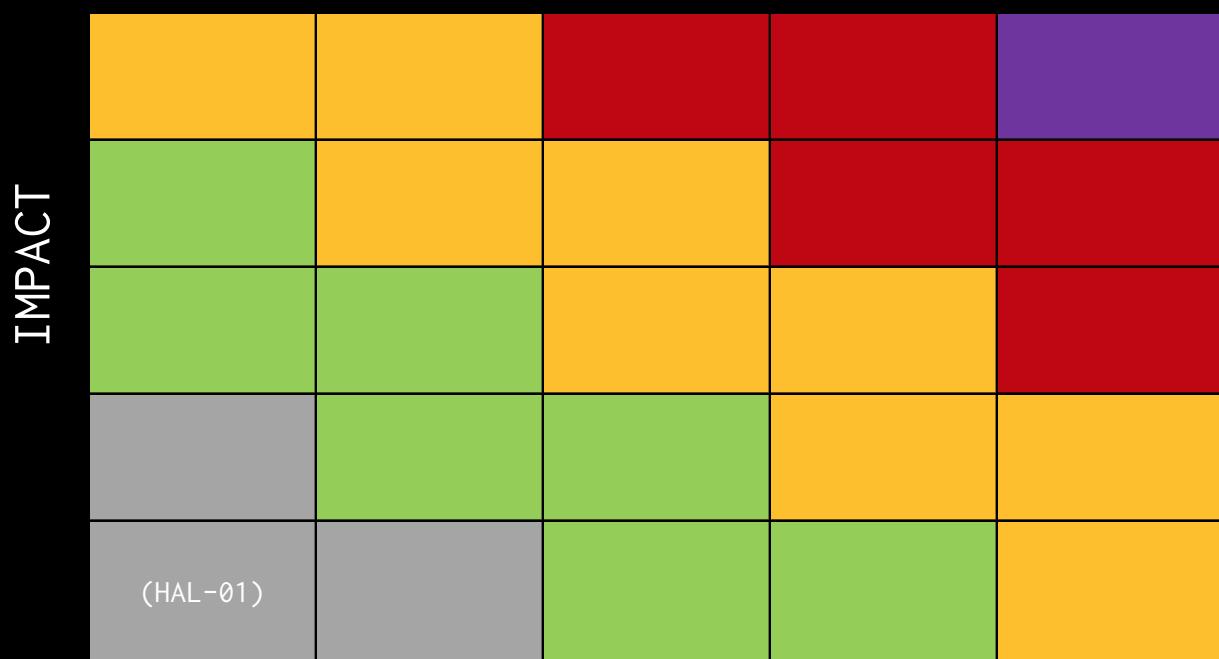
EXECUTIVE OVERVIEW

- contracts/utils/Ownable.sol
- contracts/utils/TokenTrade.sol

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	0	1

LIKELIHOOD



EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
CAUSE-EFFECT CHECKS	Informational	ACKNOWLEDGED



FINDINGS & TECH DETAILS



3.1 (HAL-01) CAUSE-EFFECT CHECKS - INFORMATIONAL

Description:

The `_payBackLoan` under `DirectLoanBaseMinimal` does not pre-check the user's balance by being `payoffAmount + adminFee + revenueShare` it does so incrementally.

Risk Level:

Likelihood - 1

Impact - 1

Recommendation:

It would be nice if the code cloud pre-check if the user had enough balance for all transfers combined instead of checking and transferring incrementally. It is recommended to always check pre/post balances after a transfer.

Remediation Plan:

ACKNOWLEDGED: The NFTfi team acknowledged this issue.

MANUAL ANALYSIS

The `functionDelegateCall` is used on:

- `airdrop/AirdropFlashLoan.sol`
- `airdrop/AirdropReceiver.sol`
- `loans/direct/loanTypes/LoanAirdropUtils.sol`

`_transferNFT` makes a delegate call on the `_nftTransferWrapper` and `_nftWrapper` contexts, calling the `transferNFT` function. This means that all wrappers registered in `PermittedNFTsAndTypeRegistry` should not register any storage variable; otherwise, storage layout issues and collision could occur. Currently, all contracts present in `nftTypeRegistry` do not hold any storage variable, which makes them compatible with the delegate calling system.

- `loans/direct/loanTypes/DirectLoanBaseMinimal.sol`

`_transferNFT` does delegate to the `nftCollateralWrapper` loan in the context of the contract that implements `DirectLoanBaseMinimal`. At this time, there is only one contract that extends `DirectLoanBaseMinimal` which corresponds to the `DirectLoanFixedOffer` contract. The `DirectLoanFixedOffer` contract does not add new storage variables, but will get all the variables present in `DirectLoanBaseMinimal` which extends `BaseLoan`, `NftReceiver` and `LoanData`. Taking this into account, storage collision could be possible between `nftCollateralWrapper` and `DirectLoanBaseMinimal` contract. Although the mappings will be used in the `nftCollateralWrappers` code, it is still possible that some mapping may collide. When transferring the mapping used in the NFT code are the following:

Listing 1

```

1   mapping(address => uint256) private _balances;
2   mapping(uint256 => address) private _tokenApprovals;
```

The `functionCall` is used on:

- `airdrop/AirdropFlashLoan.sol`

- `airdrop/AirdropReceiver.sol`

Both contracts use a call to a user-specified target with any call data. However, both `calldata` and `target` are checked against `IPermittedAirdrops` for validity, and only allow anything other than the one approved by the owners via `setAirdroptPermit` or `setAirdroptPermits`.

4.1 NftfiHub

This contract is a central point for all the contracts registered in the NFTfi system. When a contract needs the address of another contract, it will refer to the hub with a key and the hub will be providing the corresponding address.

- All critical functions like `setContract`, `setContracts` are being checked for ownership
- A contract key that does not exist will return a zero address.

4.2 DirectLoanCoordinator

It will have the `LOAN_COORDINATOR_ROLE` role in the `SmartNFT` contract

- What if the user burns the `promissoryNoteToken` and `obligationReceiptToken` NFTs before the loan period ends?
 - It is not possible to burn tokens, only `resolveLoan` does it.

4.3 PermittedAirdrop

- Properly manage the ownership
- Critical functions such as `setAirdroptPermit` and `setAirdroptPermits` are guarded.

4.4 PermittedNFTsAndTypeRegistry

It is used to register a valid NFT in the NFTfi system. The contract will be used to access NFT wrappers and register new NFT types and permits.

- Critical functions are protected by the owner
- A critical function is protected by using `onlyOwnerOrAirdropFactory` instead of `onlyOwner`. Backtracking the ownership relationship does verify that the check is safe. `AirdropFactory` is cloning new airdrops templates. However, those templates will not have permissions on the `PermittedNFTsAndTypeRegistry` contract.

4.5 nftTypeRegistry/nftTypes

They are registered under the `PermittedNFTsAndTypeRegistry` contract and used with the `AirdropReceiver` contract to wrap the underlying NFT assets. All registered wrappers should not contain any storage variable. They are used by performing a `delegatecall` to the underlying wrapped contract and do not require new storage variables; otherwise, a variable collision could occur. Currently, all contracts present in `nftTypeRegistry` do not hold any new storage variables, which makes them compatible with the delegate calling system.

4.6 SmartNFT

All operations are restricted to the `LOAN_COORDINATOR_ROLE` which is expected to be for the `DirectLoanCoordinator` contract only. However, the `transfer` is allowed to anyone.

As an example, the obligation token can be transferred to the lender:

MANUAL ANALYSIS

```
<TRANSACTION 0x0856cd4dbd704f321754376af1878413f2514b64335ca26753895a3b758d317c>
>>> promissoryNoteToken.transferFrom(a[0], a[1], 15847410939590356974, {'from':a[0]})  
Transaction sent: 0x0856cd4dbd704f321754376af1878413f2514b64335ca26753895a3b758d317c  
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 7  
SmartNft.transferFrom confirmed Block: 55 Gas used: 50016 (0.42%)  
<Transaction '0x0856cd4dbd704f321754376af1878413f2514b64335ca26753895a3b758d317c'>  
>>> █
```

CALL GRAPH AND INHERITANCE

CALL GRAPH AND INHERITANCE

AirdropFlashLoan:

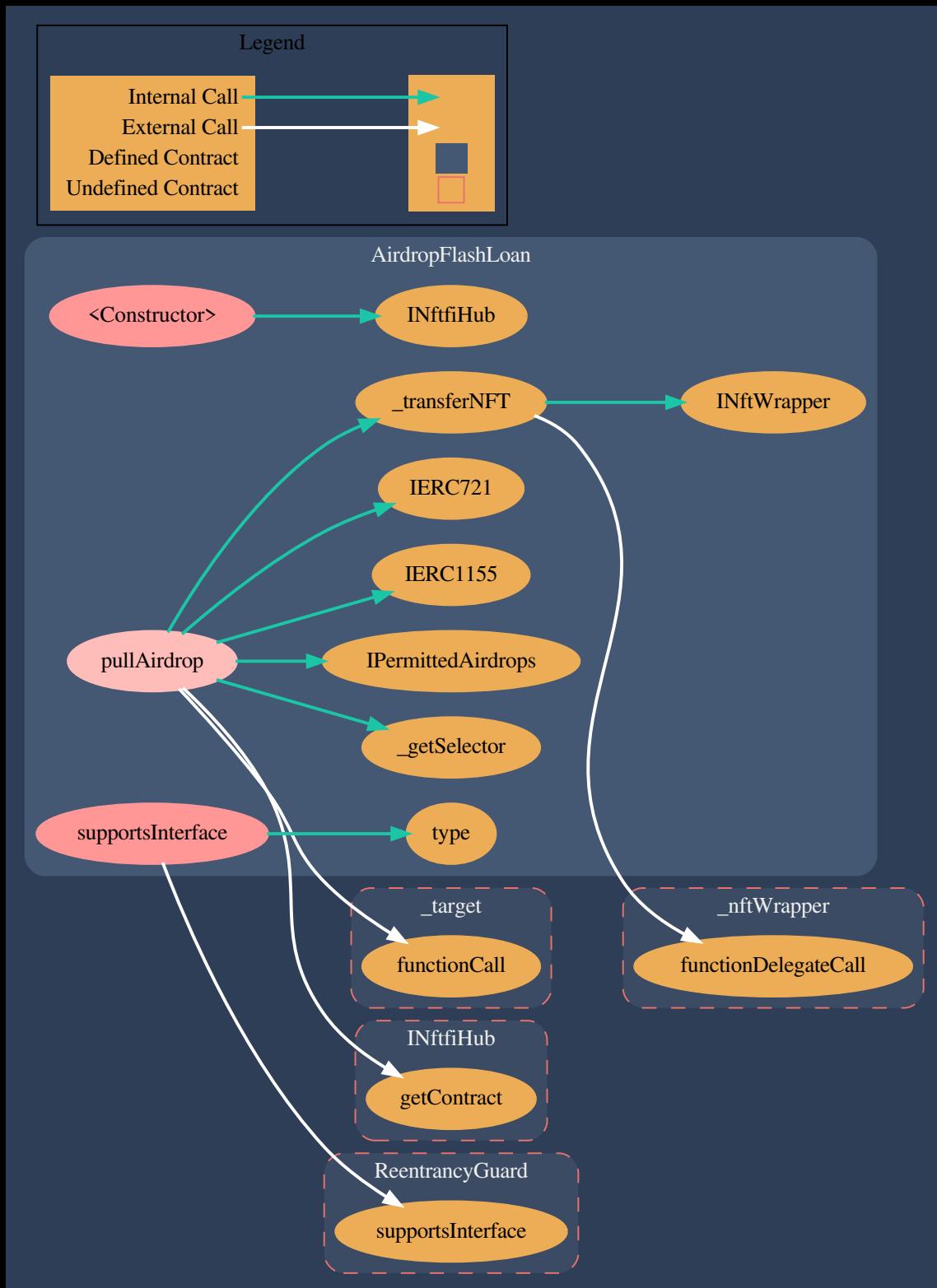


Figure 1: airdrop/AirdropFlashLoan.sol

CALL GRAPH AND INHERITANCE

AirdropReceiver:

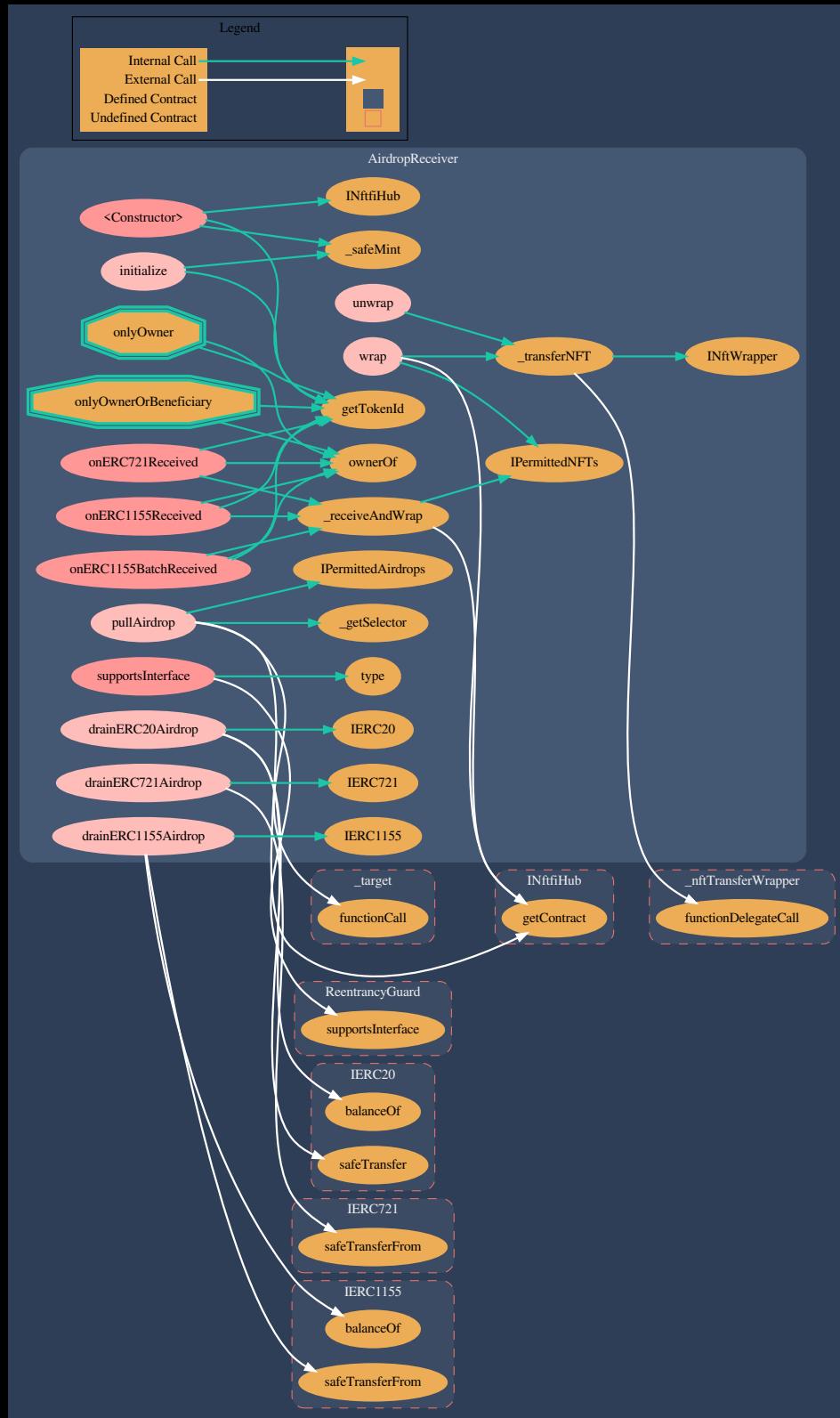


Figure 2: airdrop/AirdropReceiver.sol

CALL GRAPH AND INHERITANCE

DirectLoanBaseMinimal:

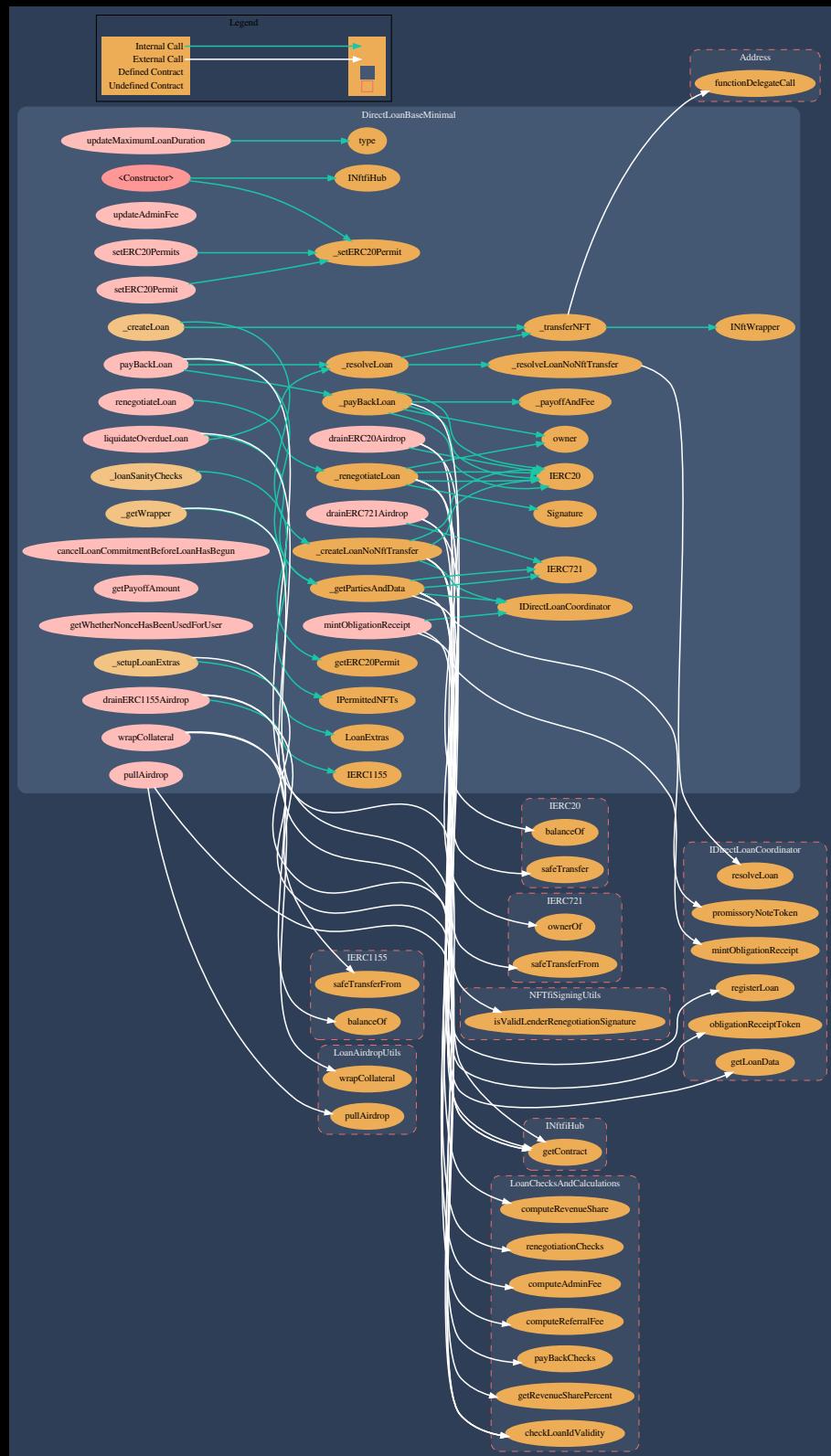


Figure 3: loans/direct/loanTypes/DirectLoanBaseMinimal.sol

CALL GRAPH AND INHERITANCE

DirectLoanFixedOffer:

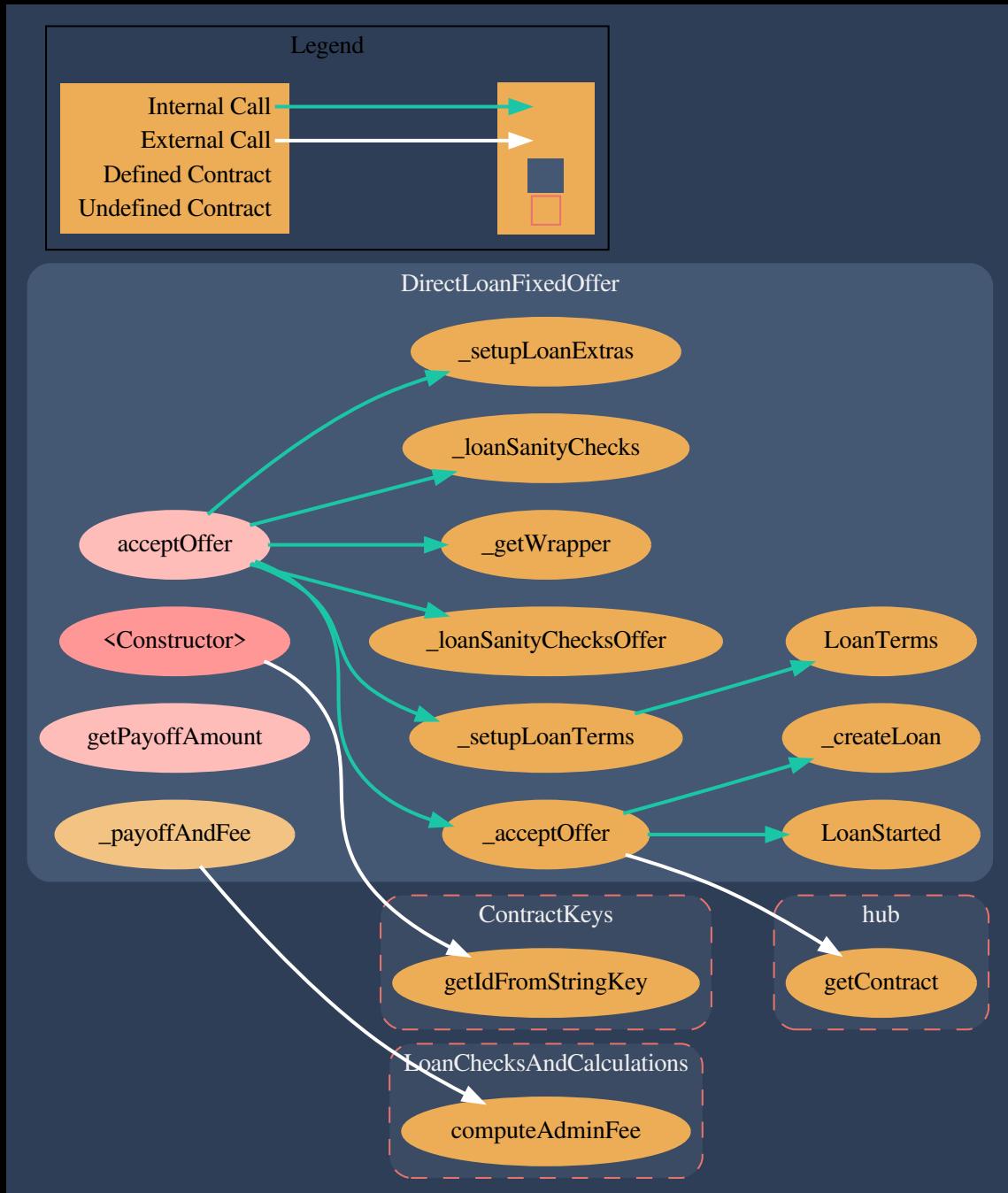


Figure 4: loans/direct/loanTypes/DirectLoanFixedOffer.sol

CALL GRAPH AND INHERITANCE

DirectLoanCoordinator:

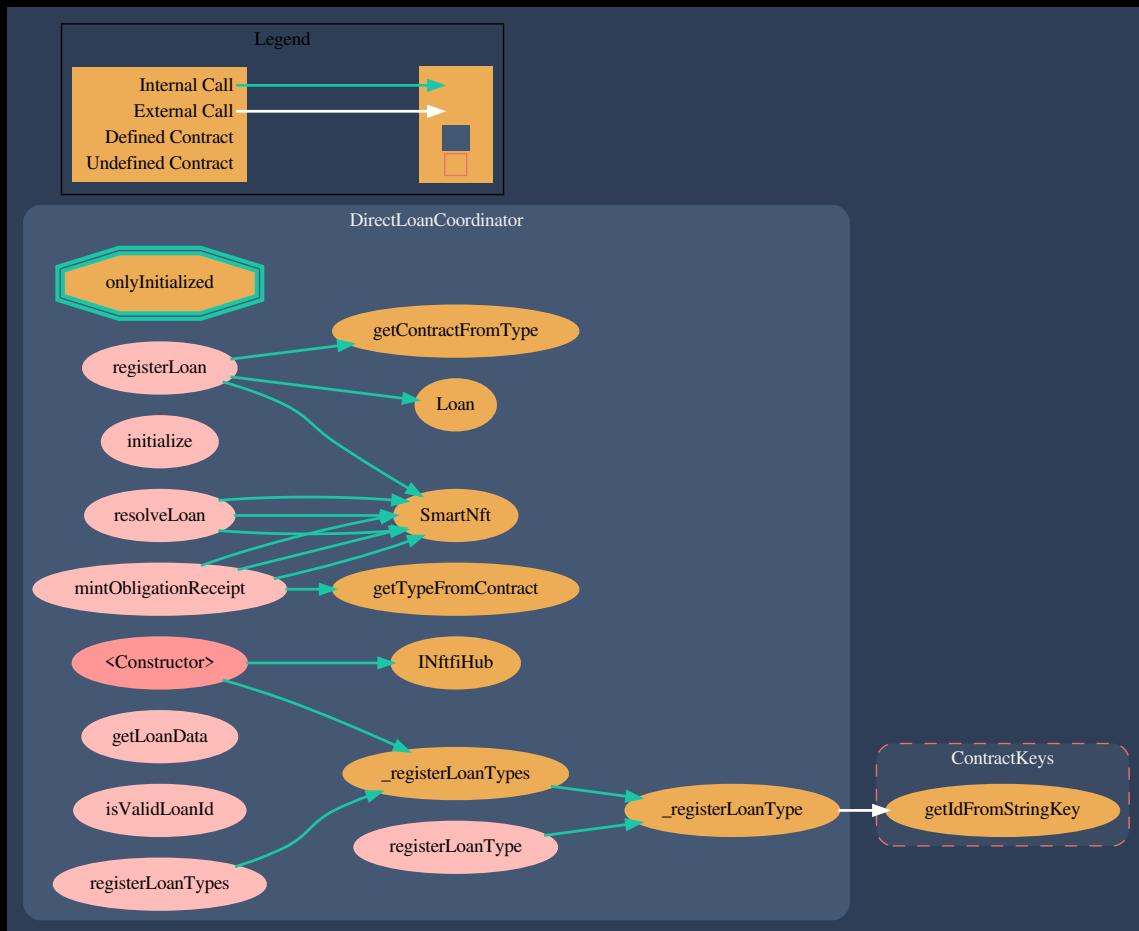


Figure 5: loans/direct/DirectLoanCoordinator.sol

CALL GRAPH AND INHERITANCE

SmartNFT:

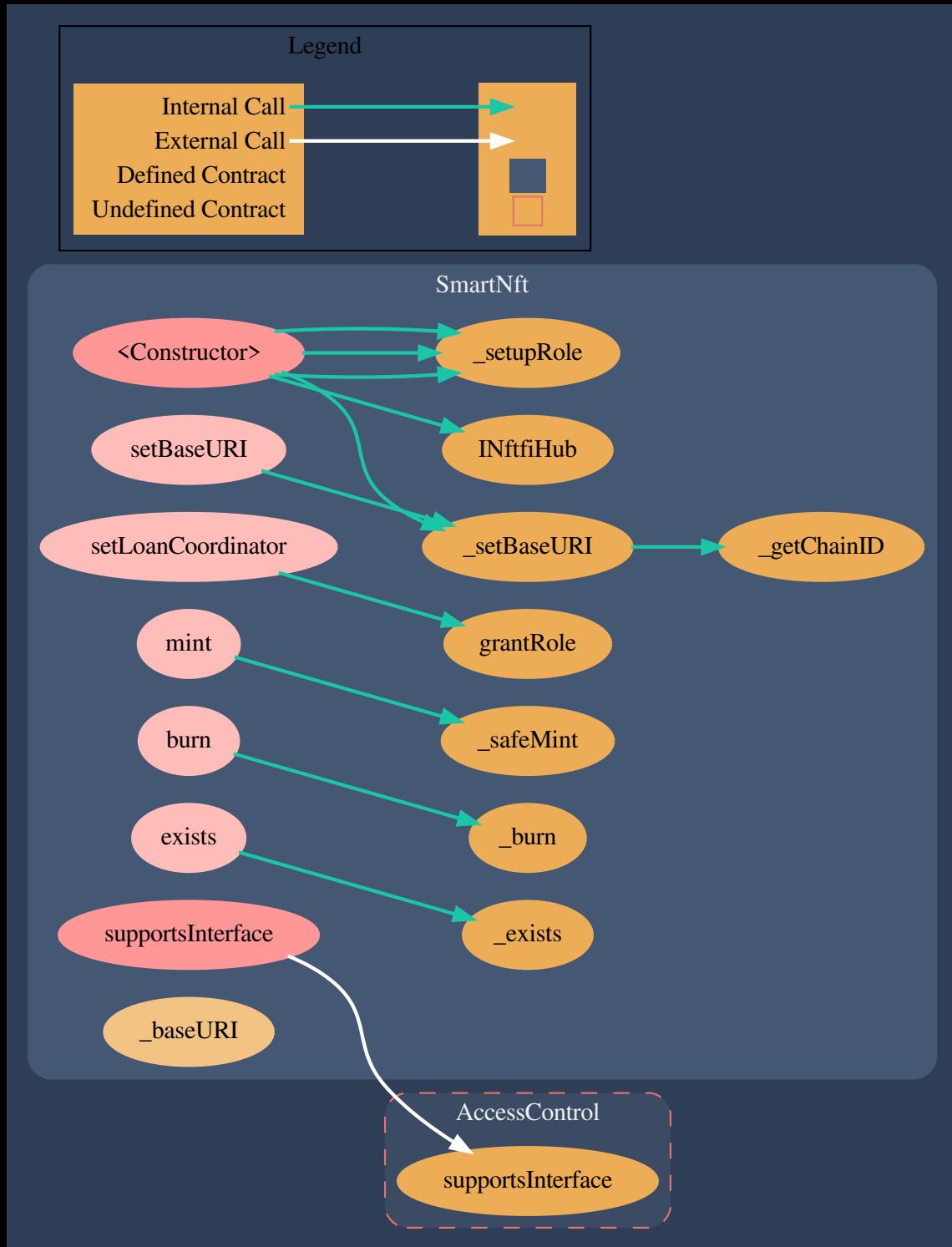


Figure 6: smartNFT/SmartNFT.sol

CALL GRAPH AND INHERITANCE

NFTfiBundler:

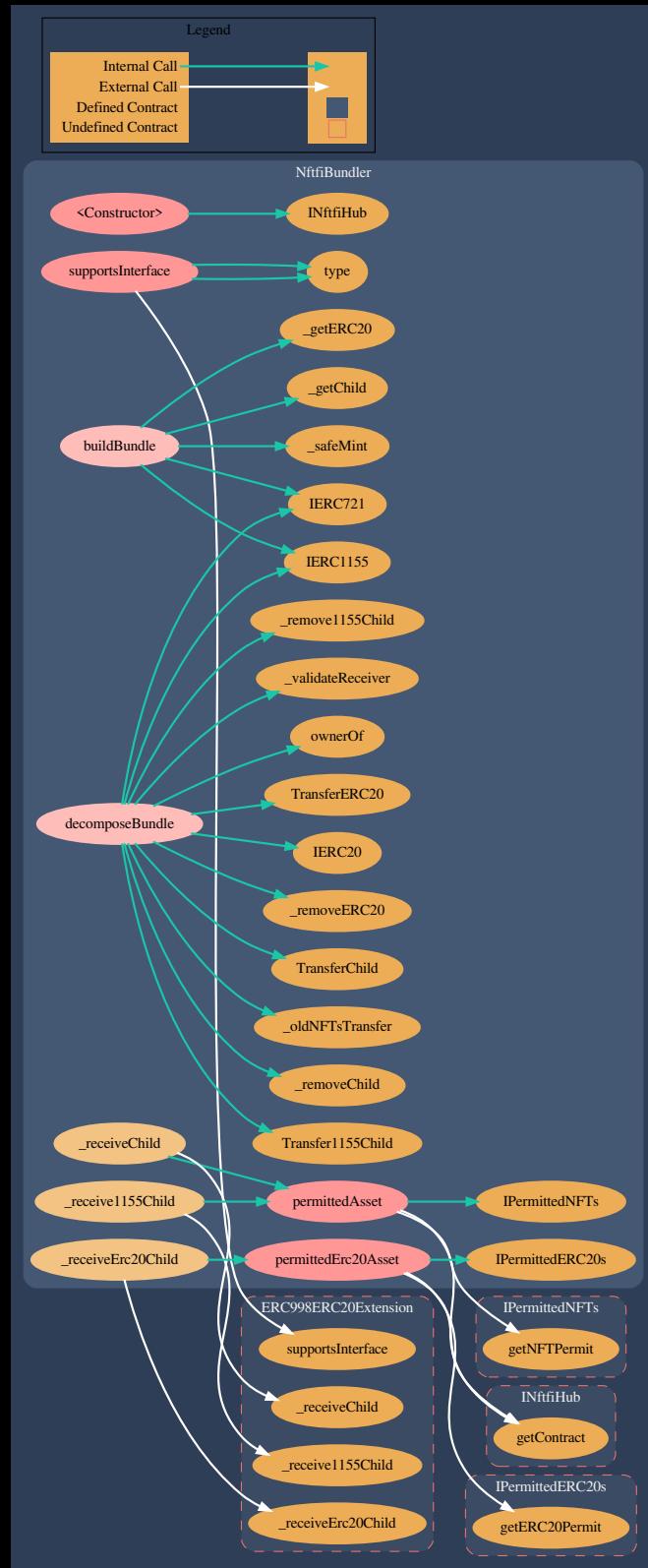


Figure 7: composable/NFTfiBundler.sol

AUTOMATED TESTING

6.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance coverage of certain areas of the scoped contract. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified all the contracts in the repository and was able to compile them correctly into their ABI and binary formats. This tool can statically verify mathematical relationships between Solidity variables to detect invalid or inconsistent usage of the contracts' APIs across the entire code-base.

Results:

```

PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) ignores return value by IERC721Receiver(to).onERC721Received(_msg)
lin-solidity/contracts/token/ERC721/ERC721.sol#389-399)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

SmartNft.constructor(address,address,string,string,string).name (contracts/smартnft/SmartNft.sol#59) shadows:
- ERC721.name (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#24) (state variable)
SmartNft.constructor(address,address,address,string,string,string).symbol (contracts/smартnft/SmartNft.sol#60) shadows:
- ERC721.symbol (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#27) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).revert' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#388) in ERC721._checkOnERC721Received(address,address,uint256,bytes) potentially used before declaration: revert(IERC721Receiver.onERC721Received.selector)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391) in ERC721._checkOnERC721Received(address,address,uint256,bytes) potentially used before declaration: reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391) in ERC721._checkOnERC721Received(address,address,uint256,bytes) potentially used before declaration: revert(uint256,uint256)(32 + reason,load(uint256))(reason) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in DirectLoanCoordinator.registerLoan(address,bytes32) (contracts/loans/direct/DirectLoanCoordinator.sol#127-158):
External calls:
- SmartNft(promissoryNoteToken).mint(_lender,smartNftId,abi.encode(totalNumLoans)) (contracts/loans/direct/DirectLoanCoordinator.sol#143)
State variables written after the call(s):
- loans[totalNumLoans] = newLoam (contracts/loans/direct/DirectLoanCoordinator.sol#145)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in DirectLoanCoordinator.registerLoan(address,bytes32) (contracts/loans/direct/DirectLoanCoordinator.sol#127-150):
External calls:
- SmartNft(promissoryNoteToken).mint(_lender,smartNftId,abi.encode(totalNumLoans)) (contracts/loans/direct/DirectLoanCoordinator.sol#143)
Event emitted after the call(s):
- UpdateStatus(totalNumLoans,smartNftId,loanContract,StatusType.NEW) (contracts/loans/direct/DirectLoanCoordinator.sol#147)
Reentrancy in DirectLoanCoordinator.resolveLoan(uint32) (contracts/loans/direct/DirectLoanCoordinator.sol#127-193):
External calls:
- SmartNft(promissoryNoteToken).burn(loan.smartNftId) (contracts/loans/direct/DirectLoanCoordinator.sol#187)
- SmartNft(oliginationReceiptNftId).burn(loan.smartNftId) (contracts/loans/direct/DirectLoanCoordinator.sol#189)
Event emitted after the call(s):
- UpdateStatus(_loanId,loan.smartNftId,msg.sender>StatusType.RESOLVED) (contracts/loans/direct/DirectLoanCoordinator.sol#192)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

SmartNft._getChainID() (contracts/smартнft/SmartNft.sol#155-162) uses assembly
- INLINE ASM (contracts/smартнft/SmartNft.sol#158-160)
ContractKeys.getIdFromStringKey(string) (contracts/utils/ContractKeys.sol#31-38) uses assembly
- INLINE ASM (contracts/utils/ContractKeys.sol#35-37)
ERC721._checkOnERC721Received(address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) uses assembly

```

Figure 8: contracts/loans/direct/contracts_loans_direct_DirectLoanCoordinator_slither

PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

Compilation warnings/errors on contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol:
Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value) in the compiler settings.
| -> contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol:57:1:
57 | contract DirectLoanFixedOffer is DirectLoanBaseMinimal {
| ^ (Relevant source part starts here and spans across multiple lines).

Reentrancy in DirectLoanBaseMinimal.payBackLoan(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#489-507):
External calls:
- _payBackLoan(_loanId, borrower, lender, loan) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#408)
 - returnData = address(token).functionCall(data, SafeERC20.sol#132)
 - (success, returnData) = target.call{value: value}(data) (openzeppelin-solidity/contracts/token/ERC20/utils/SafeERC20.sol#93)
 - IERC20(_loan, loanERC20Denomination).safeTransferFrom(msg.sender, lender, payoffAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#921)
 - IERC20(_loan, loanERC20Denomination).safeTransferFrom(msg.sender, owner, adminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#940)
- _resolveNFT(_loanId, _nftCollateralId, _recipient, _loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#500)
 - Address.functionCall(_loan, nftCollateralWrapper.abis[nftCollateralWrapperSelector].INftCollateralWrapper._transferNFT.selector, _sender, _recipient, _nftCollateralId, _NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#985-990)
 - loanCoordinator.resolveLoan(_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#999)
External calls (returnData):
- _payBackLoan(_loanId, borrower, lender, loan) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#408)
 - (success, returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)
State variables written after the calls:
- delete loanIdToLoan[_loanId] (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#505)
- delete loanIdToLoanExtras[_loanId] (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#506)
Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities>

Reentrancy in DirectLoanBaseMinimal._renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#728-795):
External calls:
- require(bool, string) (NFT/isSigningUtil.sol#50) isValueLenderRenegotiationSignature(_loanId,_newLoanDuration,_newMaximumRepaymentAmount,_renegotiationFee,Signature(lender,_lenderNonce,_expiry is invalid) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#750-759)
- IERC20(_loan, loanERC20Denomination).safeTransferFrom(borrower, lender, renegotiationFee - renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#774-778)
- IERC20(_loan, loanERC20Denomination).safeTransferFrom(borrower, owner, renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#780)
State variables written after the calls:
- loan._loanDuration = _newLoanDuration (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#783)
- loan._maximumRepaymentAmount = _newMaximumRepaymentAmount (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#784)
Reentrancy in DirectLoanBaseMinimal.liquidateOverdueLoan(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#523-562):
External calls:
- _resolveLoan(_loanId, lender, loan, loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#543)

Figure 9: contracts/loans/direct/loanTypes/contracts_loans_direct_loanTypes_DirectLoanFixedOffer_slither

Figure 10: contracts/loans/direct/loanTypes/contracts_loans_direct_-loanTypes_LoanAirdropUtils_slither

```

PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

Reentrancy in DirectLoanBaseMinimal.payBackLoan(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#489-507):
External calls:
- _payBackLoan(_loanId,borrower,lender,loan) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#490)
  - returnData = target.call.value: value(data) (openzeppelin-solidity/contracts/utils/Address.sol#132)
  - (success,returnData) = target.call.value: value(data) (openzeppelin-solidity/contracts/utils/Address.sol#132)
  - IERC20(loan).loanERC20denomination().safeTransferFrom(msg.sender, lender, payoffAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#921)
  - IERC20(loan).loanERC20denomination().safeTransferFrom(msg.sender,loanExtras.revenueSharePartner,revenueShare) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#933)
  - IERC20(loan).loanERC20denomination().safeTransferFrom(msg.sender,owner(),adminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#940)
- _resolveLoan(_loanId,borrower,loan,loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#500)
  - Address.functionDelegatecall(_loanTerms.nftCollateralWrapper.selector,_sender,_recipient,_loanCoordinator.resolveLoan,_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#899)
  - (success,returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)

nftCollateralId._NFTId successfully transferred (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#505)
  - loanCoordinator.resolveLoan(_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#899)

External calls sending eth:
- _payBackLoan(_loanId,borrower,lender,loan) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#498)
  - (success,returnData) = target.functionCall(_loanId,_newLoanDuration,_newMaximumRepaymentAmount,_renegotiationFee,Signature(lender,_lenderNonce,_expiry)) (openzeppelin-solidity/contracts/utils/Address.sol#132)
  - State variables written after the calls:
    - delete loanIdToLoan[_loanId] (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#505)
    - delete loanIdToLoanExtras[_loanId] (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#506)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/reentrancies

Reentrancy in DirectLoanBaseMinimal._renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#728-795):
External calls:
  - require(bool,string)(NFTId.isSigningHolds,isValidLenderRenegotiationSignature,_loanId,_newLoanDuration,newMaximumRepaymentAmount,_renegotiationFee,Signature(lender,_lenderNonce,_expiry)) (openzeppelin-solidity/contracts/math/Math.sol#750-759)
  - IERC20(loan).loanERC20denomination().safeTransferFrom(borrower,lender,_renegotiationFee - _renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#774-778)
  - IERC20(loan).loanERC20denomination().safeTransferFrom(borrower,owner(),_renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#780)

State variables written after the calls:
  - loan.maximumRepaymentAmount = newMaximumRepaymentAmount (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#784)

Reentrancy in DirectLoanBaseMinimal._liquidateOversubLoan(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#523-562):
External calls:
  - _resolveLoan(_loanId,lender,loan,loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#543)
  - Address.functionDelegatecall(_loanTerms.nftCollateralWrapper.selector,_sender,_recipient,_loanCoordinator.resolveLoan,_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#895)
  - (success,returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)

State variables written after the calls:
  - delete loanIdToLoan[_loanId] (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#560)

Reentrancy in DirectLoanBaseMinimal._mintOBligationReceipt(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#416-424):
External calls:
  - loanCoordinator._mintOBligationReceipt(_loanId,borrower) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#421)

```

Figure 11: contracts/loans/direct/loanTypes/contracts_loans_direct_loanTypes_DirectLoanBaseMinimal_slither

```

PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) ignores return value by IERC721Receiver(to).onERC721Received(_msg)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

SmartNft.constructor(address,address,string,string).name (contracts/smartNft/SmartNft.sol#59) shadows:
  - ERC721._checkOnERC721Received(address,address,uint256,bytes).name (state variable)
SmartNft.constructor(address,address,string,string).symbol (contracts/smartNft/SmartNft.sol#60) shadows:
  - ERC721.symbol (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#27) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#488) in ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#392)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391) in ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: revert(uint256,uint256)(reason) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#392)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-use-of-local-variables

SmartNft._getChainID() (contracts/smartNft/SmartNft.sol#155-162) uses assembly
  - INLINE ASM (contracts/smartNft/SmartNft.sol#158-160)
ContractKeys.getIdFromStringKey(string) (contracts/ContractKeys.sol#31-38) uses assembly
  - INLINE ASM (contracts/ContractKeys.sol#35-37)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) uses assembly
  - Address.isContract(address) (openzeppelin-solidity/contracts/utils/Address.sol#33-35)
Address.verifyCallResult(bool,bytes,string) (openzeppelin-solidity/contracts/utils/Address.sol#196-216) uses assembly
  - INLINE ASM (openzeppelin-solidity/contracts/utils/Address.sol#208-211)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
  - Version used: ['0.8.4', '0.8.0']
  - 0.8.4 (contracts/interfaces/IMtfiHub.sol#3)
  - 0.8.4 (contracts/smartNft/SmartNft.sol#3)
  - 0.8.4 (contracts/utils/ContractKeys.sol#3)
  - ~0.8.0 (openzeppelin-solidity/contracts/access/AccessControl.sol#4)
  - ~0.8.0 (openzeppelin-solidity/contracts/access/Role.sol#4)
  - ~0.8.0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#4)
  - ~0.8.0 (openzeppelin-solidity/contracts/token/ERC721/IERC721.sol#4)
  - ~0.8.0 (openzeppelin-solidity/contracts/token/ERC721/IERC721Receiver.sol#4)
  - ~0.8.0 (openzeppelin-solidity/contracts/token/ERC721/extensions/IERC721Metadata.sol#4)
  - ~0.8.0 (openzeppelin-solidity/contracts/utils/Address.sol#4)
  - ~0.8.0 (openzeppelin-solidity/contracts/utils/Context.sol#4)

```

Figure 12: contracts/smartNft/contracts_smartNft_SmartNft_slither

PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

```
Compiling warnings/errors on contracts/composable/NftfiBundler.sol:
Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value).
--> contracts/composable/NftfiBundler.sol:23:1
23 | contract NftfiBundler is IBundleBuilder, ERC9981155Extension, ERC998ERC20Extension {
| ^ (Relevant source part starts here and spans across multiple lines).

ERC9981155Extension._receive1155Child(uint256,address,uint256,uint256) (contracts/composable/ERC9981155Extension.sol#188-199) ignores return value by childContracts[_tokenId].add(_childContract)
ERC9981155Extension._receive1155Child(uint256,address,uint256,uint256) (contracts/composable/ERC9981155Extension.sol#186-199) ignores return value by childTokens[_tokenId][_childContract].add(_childContract)
ERC9981155Extension._remove1155Child(uint256,address,uint256,uint256) (contracts/composable/ERC9981155Extension.sol#216-237) ignores return value by childTokens[_tokenId][_childContract].remove(_childContract)
ERC9981155Extension._remove1155Child(uint256,address,uint256,uint256) (contracts/composable/ERC9981155Extension.sol#216-237) ignores return value by childContracts[_tokenId].remove(_childContract)
ERC998ERC20Extension._receiveErc20Child(address,uint256,address,uint256) (contracts/composable/ERC998ERC20Extension.sol#170-183) ignores return value by erc20ChildContracts[_tokenId].add(_erc20Contract)
ERC998ERC20Extension._removeErc20(uint256,address,uint256) (contracts/composable/ERC998ERC20Extension.sol#191-203) ignores return value by erc20ChildContracts[_tokenId].remove(_erc20Contract)
ERC998TopDown._removeRootOfChild(address,uint256) (contracts/composable/ERC998TopDown.sol#153-188) ignores return value by IERC998ERC721TopDown(rootOwnerAddress).rootOwnerOfChild(address(this),998TopDown.sol#156-177)
ERC998TopDown._removeChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#405-418) ignores return value by childTokens[_tokenId][_childContract].remove(_childTokenId) (contracts/composable/ERC998TopDown._removeChild(uint256,address,uint256).remove(_childContract))
ERC998TopDown._receiveChild(address,uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#446-460) ignores return value by childContracts[_tokenId].add(_childContract) (contracts/composable/ERC998TopDown._receiveChild(address,uint256,address,uint256).add(_childContract))
ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) ignores return value by IERC721Receiver(to).onERC721Received(_msg)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

NftfiBundler.constructor(address,string,_name) (contracts/composable/NftfiBundler.sol#41) shadows:
- ERC721._name (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#24) (state variable)
NftfiBundler.constructor(address,string,_symbol) (contracts/composable/NftfiBundler.sol#42) shadows:
- ERC721._symbol (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#47) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

NftfiBundler.permittedAsset(address) (contracts/composable/NftfiBundler.sol#68-71) has external calls inside a loop: permittedNFTs = IPermittedNFTs(hub.getContract(ContractKey.PERMITTED_NFTS).ls#69)
NftfiBundler.permittedAsset(address) (contracts/composable/NftfiBundler.sol#68-71) has external calls inside a loop: permittedNFTs.getNFTPermit_asset() > 0 (contracts/composable/NftfiBundler.NftfiBundler.buildElements(IBundleBuilder.BundleElements,address,address)) (contracts/composable/NftfiBundler.sol#91-132) has external calls inside a loop: IERC721I_bundleElements.erc721s[i].tokens(this),_bundleElements.erc721s[i].id,abi.encodePacked(bundleId)) (contracts/composable/NftfiBundler.sol#105-110)
ERC998TopDown._get Child(address,uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#313-321) has external calls inside a loop: IERC721I_childContract.transferFrom(_from,address)
NftfiBundler.permittedERC20Asset(address) (contracts/composable/NftfiBundler.sol#78-81) has external calls inside a loop: permittedERC20s = IPermittedERC20s(hub.getContract(ContractKey.PERMITED_NFTS).ls#79)
NftfiBundler.permittedERC20Asset(address) (contracts/composable/NftfiBundler.sol#78-81) has external calls inside a loop: permittedERC20s.getERC20Permit_erc20Contract() (contracts/composable/Address.FunctionCallWithValue(address,bytes,uint256,bytes)) (openzeppelin-solidity/contracts/utils/Address.sol#129-134) has external calls inside a loop: (success,returnData) = target.callViaContract(address,utils/Address.sol#132)
NftfiBundler.buildBundle(IBundleBuilder.BundleElements,address,address) (contracts/composable/NftfiBundler.sol#91-132) has external calls inside a loop: IERC1155I_bundleElements.erc1155s[i].scope,ids,_bundleElements.erc1155s[i].amounts,abi.encodePacked(bundleId)) (contracts/composable/NftfiBundler.sol#121-127)
NftfiBundler.decomposeBundle(uint256,address) (contracts/composable/NftfiBundler.sol#141-181) has external calls inside a loop: IERC1155I_childContract.safeTransferFrom(address(this),_receive
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composable/ERC998TopDown.sol#153-188) ignores return value by IERC998ERC721TopDown(rootOwnerAddress).rootOwnerOfChild(address(this),998TopDown.sol#156-177)
ERC998TopDown._removeChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#405-418) ignores return value by childTokens[_tokenId][_childContract].remove(_childTokenId) (contracts/composable/ERC998TopDown._removeChild(uint256,address,uint256).remove(_childContract))
ERC998TopDown._receiveChild(address,uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#446-460) ignores return value by childContracts[_tokenId].add(_childContract) (contracts/composable/ERC998TopDown._receiveChild(address,uint256,address,uint256).add(_childContract))
ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) ignores return value by IERC721Receiver(to).onERC721Received(_msg)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'ERC998TopDown.rootOwnerOfChild(address,uint256).returnedRootOwner' (contracts/composable/ERC998TopDown.sol#169)' in ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composable/ERC998TopDown.rootOwnerOfChild(address,uint256).returnedRootOwner) used before declaration: returnedRootOwner (contracts/composable/ERC998TopDown.sol#172)
Variable 'ERC998TopDown.rootOwnerOfChild(address,uint256).returnedRootOwner' (contracts/composable/ERC998TopDown.rootOwnerOfChild(address,uint256)) (contracts/composable/ERC998TopDown.rootOwnerOfChild(address,uint256).returnedRootOwner) used before declaration: returnedRootOwner (contracts/composable/ERC998TopDown.rootOwnerOfChild(address,uint256).returnedRootOwner.sol#173)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retVal' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#389) potentially used before declaration: retVal = IERC721Receiver.onERC721Received(address,to).selector (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#390)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#389) potentially used before declaration: reason (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#392)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#393)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-use-of-local-variables

Reentrancy in ERC998TopDown.safeTransferChild(uint256,address,address,uint256) (contracts/composable/ERC998TopDown.sol#223-232):
- External calls:
  - IERC721I_childContract.safeTransferFrom(Address(this),_to,_childTokenId) (contracts/composable/ERC998TopDown.sol#230)
  Event emitted after the call(s):
  - TransferChild_(fromTokenId,_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#231)
Reentrancy in ERC998TopDown.safeTransferChild(uint256,address,address,uint256) (contracts/composable/ERC998TopDown.sol#242-256):
External calls:
- IERC721I_childContract.safeTransferFrom(Address(this),_to,_childTokenId,_data) (contracts/composable/ERC998TopDown.sol#253)
Event emitted after the call(s):
- TransferChild_(fromTokenId,_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#254)
Reentrancy in ERC998TopDown.transferChild(uint256,address,address,uint256) (contracts/composable/ERC998TopDown.sol#265-274):
External calls:
- _oldNFTsTransfer_(to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#277)
  - IERC721I_childContract.approve(Address(this),_childTokenId) (contracts/composable/ERC998TopDown.sol#506-518)
  - IERC721I_childContract.transferFrom(Address(this),_to,_childTokenId) (contracts/composable/ERC998TopDown.sol#512)
  Event emitted after the call(s):
  - TransferChild_(fromTokenId,_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#273)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ERC998TopDown._parse tokenId(bytes) (contracts/composable/ERC998TopDown.sol#116-130) uses assembly
  - INLINE ASM (contracts/composable/ERC998TopDown.sol#117-129)
ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composable/ERC998TopDown.sol#153-188) uses assembly
  - INLINE ASM (contracts/composable/ERC998TopDown.sol#184-186)
ERC998TopDown._parse tokenId(bytes) (contracts/composable/ERC998TopDown.sol#485-490) uses assembly
  -
```

Figure 13: contracts/composable/contracts_composable_NftfiBundler_slither

Figure 16: contracts/composable/contracts_composable_ERC998ERC20Extension_slither

```
PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

ERC998ERC20Extension._receiveErc20Child(address,uint256,address,uint256) (contracts/composable/ERC998ERC20Extension.sol#170-183) ignores return value by erc20ChildContracts[_tokenId].add(_erc
ERC20Contract).sol#179)
ERC998ERC20Extension._removeERC20(uint256,address,uint256) (contracts/composable/ERC998ERC20Extension.sol#191-203) ignores return value by erc20ChildContracts[_tokenId].remove(_erc20Contract)
ion.sol#201)
ERC998TopDown._removeChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#186-199) ignores return value by childTokens[_tokenId][_childContract].add(_childContract).add
(_childContract).sol#195)
ERC998TopDown._receiveErc20Child(uint256,address,uint256,uint256) (contracts/composable/ERC998TopDown.sol#186-199) ignores return value by childTokens[_tokenId][_childContract].remo
ERC998TopDown._removeErc20Child(uint256,address,uint256,uint256) (contracts/composable/ERC998TopDown.sol#216-237) ignores return value by childTokens[_tokenId][_childContract].remo
ERC998TopDown._removeErc20Child(uint256,address,uint256,uint256) (contracts/composable/ERC998TopDown.sol#216-237) ignores return value by childContracts[_tokenId].remove(_childCont
tention.sol#234)
ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composable/ERC998TopDown.rootOwnerOfChild(address,uint256) #153-188) ignores return value by IERC998ERC721TopDown(rootOwnerAddress).rootOwnerOfChild(address>this),
g9898TopDown.sol#153-188)
ERC998TopDown._removeChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#485-418) ignores return value by childTokens[_tokenId][_childContract].remove(_childTokenId) (contr
ERC998TopDown._removeChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#485-418) ignores return value by childContracts[_tokenId].remove(_childContract) (contracts/composa
ERC998TopDown._receiveChild(address,uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#446-460) ignores return value by childContracts[_tokenId].add(_childContract) (contracts/c
ERC998TopDown._receiveChild(address,uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#446-460) ignores return value by childTokens[_tokenId].childContract.add(_childTokenId)
457)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) ignores return value by IERC721Receiver(to).onERC721Received(_m
Lin-solidity/contracts/token/ERC721/ERC721.sol#389-399)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'ERC998TopDown.rootOwnerOfChild(address,uint256).returnedRootOwner (contracts/composable/ERC998TopDown.sol#169)' in ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composa
ly used before declaration returnedRootOwner & ERC998_MAGIC_MASK == ERC998_MAGIC_VALUE (contracts/composable/ERC998TopDown.sol#172)
Variable 'ERC998TopDown.rootOwnerOfChild(address,uint256).onERC721Received(address,uint256.selector(OpenZeppelin-Solidity/contracts/token/ERC721/ERC721.sol#390))' in ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composa
ly used before declaration onERC721Received.selector(OpenZeppelin-Solidity/contracts/token/ERC721/ERC721.sol#390)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#389)' in ERC721._checkOnERC721Received(address,address,u
tracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: retval = IERC721Receiver.onERC721Received.selector(OpenZeppelin-Solidity/contracts/token/ERC721/ERC721.sol#390)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,u
tracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,u
tracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-use-of-local-variables

Reentrancy in ERC9981155Extension.safeBatchTransferChild(uint256,address,address,uint256)[],uint256[],bytes) (contracts/composable/ERC9981155Extension.sol#90-116):
External calls:
- IERC1155._childContract.safecallTransferFrom(address(this),_to,_childTokenIds,_amounts,_data) (contracts/composable/ERC9981155Extension.sol#113)
Event emitted after the calls):
- Transfer1155BatchChild[_tokenId],_to,_childContract,_childTokenIds,_amounts) (contracts/composable/ERC9981155Extension.sol#114)
Reentrancy in ERC9981155Extension.safeTransferChild(uint256,address,address,uint256,uint256,bytes) (contracts/composable/ERC9981155Extension.sol#62-79):
External calls:
- IERC1155._childContract.safetransferFrom(address(this),_to,_childTokenId,_amount,_data) (contracts/composable/ERC9981155Extension.sol#76)
Event emitted after the calls):
- Transfer1155Child[_tokenId],_to,_childContract,_childTokenId,_amount) (contracts/composable/ERC9981155Extension.sol#77)
Reentrancy in ERC998TopDown.safeTransferChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#23-232):
External calls:
- IERC721._childContract.safetransferFrom(address(this),_to,_childTokenId) (contracts/composable/ERC998TopDown.sol#230)
Event emitted after the calls):
- TransferChild[_fromTokenId],_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#231)
Reentrancy in ERC998TopDown.safeTransferChild(uint256,address,address,uint256,bytes) (contracts/composable/ERC998TopDown.sol#242-256):
External calls:
- IERC721._childContract.safetransferFrom(address(this),_to,_childTokenId,_data) (contracts/composable/ERC998TopDown.sol#253)
Event emitted after the calls):
- TransferChild[_fromTokenId],_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#254)
```

Figure 15: contracts/composable/contracts_composable_ERC9981155Extension_slither

```
PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

ERC998ERC20Extension._receiveErc20Child(address,uint256,address,uint256) (contracts/composable/ERC998ERC20Extension.sol#170-183) ignores return value by erc20ChildContracts[_tokenId].add(_erc
ERC20Contract).sol#179)
ERC998ERC20Extension._removeERC20(uint256,address,uint256) (contracts/composable/ERC998ERC20Extension.sol#191-203) ignores return value by erc20ChildContracts[_tokenId].remove(_erc20Contract)
ion.sol#201)
ERC998TopDown._removeChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#186-199) ignores return value by childTokens[_tokenId][_childContract].add(_childContract).add
(_childContract).sol#195)
ERC998TopDown._receiveErc20Child(uint256,address,uint256,uint256) (contracts/composable/ERC998TopDown.sol#186-199) ignores return value by childTokens[_tokenId][_childContract].remo
ERC998TopDown._removeErc20Child(uint256,address,uint256,uint256) (contracts/composable/ERC998TopDown.sol#216-237) ignores return value by childTokens[_tokenId][_childContract].remo
ERC998TopDown._removeErc20Child(uint256,address,uint256,uint256) (contracts/composable/ERC998TopDown.sol#216-237) ignores return value by childContracts[_tokenId].remove(_childCont
tention.sol#234)
ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composable/ERC998TopDown.rootOwnerOfChild(address,uint256) #153-188) ignores return value by IERC998ERC721TopDown(rootOwnerAddress).rootOwnerOfChild(address>this),
g9898TopDown.sol#153-188)
ERC998TopDown._removeChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#485-418) ignores return value by childTokens[_tokenId][_childContract].remove(_childTokenId) (contr
ERC998TopDown._removeChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#485-418) ignores return value by childContracts[_tokenId].remove(_childContract) (contracts/composa
ERC998TopDown._receiveChild(address,uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#446-460) ignores return value by childContracts[_tokenId].add(_childContract) (contracts/c
ERC998TopDown._receiveChild(address,uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#446-460) ignores return value by childTokens[_tokenId].childContract.add(_childTokenId)
457)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) ignores return value by IERC721Receiver(to).onERC721Received(_m
Lin-solidity/contracts/token/ERC721/ERC721.sol#389-399)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'ERC998TopDown.rootOwnerOfChild(address,uint256).returnedRootOwner (contracts/composable/ERC998TopDown.sol#169)' in ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composa
ly used before declaration returnedRootOwner & ERC998_MAGIC_MASK == ERC998_MAGIC_VALUE (contracts/composable/ERC998TopDown.sol#172)
Variable 'ERC998TopDown.rootOwnerOfChild(address,uint256).onERC721Received(address,uint256.selector(OpenZeppelin-Solidity/contracts/token/ERC721/ERC721.sol#390))' in ERC998TopDown.rootOwnerOfChild(address,uint256) (contracts/composa
ly used before declaration onERC721Received.selector(OpenZeppelin-Solidity/contracts/token/ERC721/ERC721.sol#390)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#389)' in ERC721._checkOnERC721Received(address,address,u
tracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: retval = IERC721Receiver.onERC721Received.selector(OpenZeppelin-Solidity/contracts/token/ERC721/ERC721.sol#390)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,u
tracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,u
tracts/token/ERC721/ERC721.sol#382-403) potentially used before declaration: reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-use-of-local-variables

Reentrancy in ERC998TopDown.safeTransferChild(uint256,address,uint256) (contracts/composable/ERC998TopDown.sol#223-232):
External calls:
- IERC721._childContract.safetransferFrom(address(this),_to,_childTokenId) (contracts/composable/ERC998TopDown.sol#230)
Event emitted after the calls):
- TransferChild[_fromTokenId],_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#231)
Reentrancy in ERC998TopDown.safeTransferChild(uint256,address,uint256,bytes) (contracts/composable/ERC998TopDown.sol#242-256):
External calls:
- IERC721._childContract.safetransferFrom(address(this),_to,_childTokenId,_data) (contracts/composable/ERC998TopDown.sol#253)
Event emitted after the calls):
- TransferChild[_fromTokenId],_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#254)
Reentrancy in ERC998TopDown.transferChild(uint256,address,address,uint256) (contracts/composable/ERC998TopDown.sol#265-274):
External calls:
- _dntfTransfer(_fromTokenId,_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#272)
- IERC721._childContract.approve(address(this),_childTokenId) (contracts/composable/ERC998TopDown.sol#506-510)
- IERC721._childContract.transferFrom(address(this),_to,_childTokenId) (contracts/composable/ERC998TopDown.sol#512)
Event emitted after the calls):
- TransferChild[_fromTokenId],_to,_childContract,_childTokenId) (contracts/composable/ERC998TopDown.sol#273)
Reentrancy in ERC998ERC20Extension.transferERC20(uint256,address,address,uint256) (contracts/composable/ERC998ERC20Extension.sol#74-87):
External calls:
- IERC20(_erc20Contract).safeTransfer(_to,_value) (contracts/composable/ERC998ERC20Extension.sol#85)
Event emitted after the calls):
- TransferERC20[_tokenId],_to,_erc20Contract,_value) (contracts/composable/ERC998ERC20Extension.sol#86)
```

PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

```
Compilation warnings/errors on contracts/nftTypeRegistry/nftTypes/CryptoKittiesWrapper.sol:
Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value).
--> contracts/airdrop/AirdropReceiver.sol#28:1
28 | contract AirdropReceiver is ERC721Enumerable, ERC721Holder, ERC1155Holder, Initializable, ReentrancyGuard {
| ^ (Relevant source part starts here and spans across multiple lines).

AirdropReceiver (contracts/airdrop/AirdropReceiver.sol#28-318) is an upgradeable contract that does not protect its initialize functions: AirdropReceiver.initialize(address) (contracts/airdrop/com/crytic/slither/wiki/Detector#unprotected-upgradeable-contract)

Reentrancy in AirdropReceiver.unwrap(address) (contracts/airdrop/AirdropReceiver.sol#119-130):
External calls:
- _transferNFTInTransferWrapper(address(this),_receiver,wrappedNft,wrappedNftId) (contracts/airdrop/AirdropReceiver.sol#122)
- _nftTransferWrapper.functionDelegateCallabi.encodeWithSelector(INTWrapper._nftTransferWrapper).transferNFT.selector,_sender,_recipient,_nftCollateralContract,_nftCollateralId
d) (contracts/airdrop/AirdropReceiver.sol#212-221)
    - (success,returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)
    State variables written after the calls:
    - nftTransferWrapper = address(0) (contracts/airdrop/AirdropReceiver.sol#129)
    - wrappedNft = address(0) (contracts/airdrop/AirdropReceiver.sol#127)
    - wrappedNftId = 0 (contracts/airdrop/AirdropReceiver.sol#128)
Reentrancy in AirdropReceiver.unwrap(address, address,address,uint256) (contracts/airdrop/AirdropReceiver.sol#91-117):
External calls:
- _transferNFTInTransferWrapper(_from,address(this),_nftCollateralContract,_nftCollateralId) (contracts/airdrop/AirdropReceiver.sol#108)
- _nftTransferWrapper.functionDelegateCallabi.encodeWithSelector(INTWrapper._nftTransferWrapper).transferNFT.selector,_sender,_recipient,_nftCollateralContract,_nftCollateralId
d) (contracts/airdrop/AirdropReceiver.sol#212-221)
    - (success,returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)
    State variables written after the calls:
    - wrappedNft = _nftCollateralContract (contracts/airdrop/AirdropReceiver.sol#111)
    - wrappedNftId = _nftCollateralId (contracts/airdrop/AirdropReceiver.sol#111)
Reference: https://github.com/crytic/slither/wiki/Detector#Documentation#reentrancy-vulnerabilities-1

AirdropReceiver.pullAirdrop(address,bytes) (contracts/airdrop/AirdropReceiver.sol#132-141) ignores return value by _target.functionCall_(data) (contracts/airdrop/AirdropReceiver.sol#140)
AirdropReceiver.transferNFT(address,address,address,uint256) (contracts/airdrop/AirdropReceiver.sol#205-222) ignores return value by _nftTransferWrapper.functionDelegateCallabi.encodeWithSelector(INTWrapper._nftTransferWrapper).transferNFT.selector,_sender,_recipient,_nftCollateralContract,_nftCollateralId,NFT was not successfully transferred (contracts/airdrop/AirdropReceiver.sol#212-221)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) ignores return value by ERC721Receiver(to).onERC721Received(_msgValue,address,uint256,bytes)
Reference: https://github.com/crytic/slither/wiki/Detector#Documentation#reentrancy-vulnerabilities-1

Reentrancy in AirdropReceiver.unwrap(address) (contracts/airdrop/AirdropReceiver.sol#119-130):
|
```

Figure 17: contracts/nftTypeRegistry/nftTypes/contracts_nftTypeRegistry_nftTypes_CryptoKittiesWrapper_slither

PROBLEMS 44 OUTPUT TERMINAL GITLENS DEBUG CONSOLE

```
Compilation warnings/errors on contracts/airdrop/AirdropReceiverFactory.sol:
Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value).
--> contracts/airdrop/AirdropReceiver.sol#28:1
28 | contract AirdropReceiver is ERC721Enumerable, ERC721Holder, ERC1155Holder, Initializable, ReentrancyGuard {
| ^ (Relevant source part starts here and spans across multiple lines).

AirdropReceiver (contracts/airdrop/AirdropReceiver.sol#28-318) is an upgradeable contract that does not protect its initialize functions: AirdropReceiver.initialize(address) (contracts/airdrop/com/crytic/slither/wiki/Detector#unprotected-upgradeable-contract)

Reentrancy in AirdropReceiver.unwrap(address) (contracts/airdrop/AirdropReceiver.sol#119-130):
External calls:
- _transferNFTInTransferWrapper(address(this),_receiver,wrappedNft,wrappedNftId) (contracts/airdrop/AirdropReceiver.sol#122)
- _nftTransferWrapper.functionDelegateCallabi.encodeWithSelector(INTWrapper._nftTransferWrapper).transferNFT.selector,_sender,_recipient,_nftCollateralContract,_nftCollateralId
d) (contracts/airdrop/AirdropReceiver.sol#212-221)
    - (success,returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)
    State variables written after the calls:
    - nftTransferWrapper = address(0) (contracts/airdrop/AirdropReceiver.sol#129)
    - wrappedNft = address(0) (contracts/airdrop/AirdropReceiver.sol#127)
    - wrappedNftId = 0 (contracts/airdrop/AirdropReceiver.sol#128)
Reentrancy in AirdropReceiver.unwrap(address, address,address,uint256) (contracts/airdrop/AirdropReceiver.sol#91-117):
External calls:
- _transferNFTInTransferWrapper(_from,address(this),_nftCollateralContract,_nftCollateralId) (contracts/airdrop/AirdropReceiver.sol#108)
- _nftTransferWrapper.functionDelegateCallabi.encodeWithSelector(INTWrapper._nftTransferWrapper).transferNFT.selector,_sender,_recipient,_nftCollateralContract,_nftCollateralId
d) (contracts/airdrop/AirdropReceiver.sol#212-221)
    - (success,returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)
    State variables written after the calls:
    - wrappedNft = _nftCollateralContract (contracts/airdrop/AirdropReceiver.sol#111)
    - wrappedNftId = _nftCollateralId (contracts/airdrop/AirdropReceiver.sol#111)
Reference: https://github.com/crytic/slither/wiki/Detector#Documentation#reentrancy-vulnerabilities-1

AirdropReceiver.pullAirdrop(address,bytes) (contracts/airdrop/AirdropReceiver.sol#132-141) ignores return value by _target.functionCall_(data) (contracts/airdrop/AirdropReceiver.sol#140)
AirdropReceiver.transferNFT(address,address,address,uint256) (contracts/airdrop/AirdropReceiver.sol#205-222) ignores return value by _nftTransferWrapper.functionDelegateCallabi.encodeWithSelector(INTWrapper._nftTransferWrapper).transferNFT.selector,_sender,_recipient,_nftCollateralContract,_nftCollateralId,NFT was not successfully transferred (contracts/airdrop/AirdropReceiver.sol#212-221)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#382-403) ignores return value by ERC721Receiver(to).onERC721Received(_msgValue,address,uint256,bytes)
Reference: https://github.com/crytic/slither/wiki/Detector#Documentation#reentrancy-vulnerabilities-1

Reentrancy in AirdropReceiver.unwrap(address) (contracts/airdrop/AirdropReceiver.sol#119-130):
|
```

Figure 18: contracts/airdrop/contracts_airdrop_AirdropReceiverFactory_slither

AirdropFlashLoan.pullAirdrop(address,uint256,address,address,bytes,address,uint256,bool,uint256,address) (contracts/airdrop/AirdropFlashLoan.sol#35-75) ignores return value by _target.functionDelegateCall(abi.encodeWithSelector(_nftWrapper.functionDelegateCall.selector), abi.encodeWithSelector(_nftCollateralContract._nftCollateralId.selector), address)

AirdropFlashLoan.transferNFT(address,address,address,address,uint256) (contracts/airdrop/AirdropFlashLoan.sol#84-101) ignores return value by _nftWrapper.functionDelegateCall(abi.encodeWithSelector(_nftCollateralContract._nftCollateralId.selector), address)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AirdropFlashLoan.getSelector(bytes) (contracts/airdrop/AirdropFlashLoan.sol#103-108) uses assembly

- INLINE ASM (contracts/airdrop/AirdropFlashLoan.sol#105-107)

ContractKeys.getIdFromStringKey(string) (contracts/utils/ContractKeys.sol#31-38) uses assembly

Address.isContract(address) (openzeppelin-solidity/contracts/utils/Address.sol#27-37) uses assembly

- INLINE ASM (openzeppelin-solidity/contracts/utils/Address.sol#33-35)

Address.verifyCallResult(bool,bytes,string) (openzeppelin-solidity/contracts/utils/Address.sol#196-216) uses assembly

- INLINE ASM (openzeppelin-solidity/contracts/utils/Address.sol#208-211)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:

- Version used: "[0.8.4-'0.8.0']"
- 0.8.4 (contracts/airdrop/AirdropFlashLoan.sol#3)
- 0.8.4 (contracts/interfaces/INftWrapper.sol#3)
- 0.8.4 (contracts/interfaces/INftHub.sol#3)
- 0.8.4 (contracts/interfaces/INftSubmittedAirdrops.sol#3)
- 0.8.4 (openzeppelin-solidity/contracts/security/ReentrancyGuard.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/token/ERC1155/IERC1155.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/token/ERC1155/IERC1155Receiver.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/token/ERC1155/IERC1155Holder.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/token/ERC1155/IERC1155Receiver.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/token/ERC20/utils/SafeERC20.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/token/ERC721/IERC721.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/token/ERC721/IERC721Receiver.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/token/ERC721/IERC721Holder.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/utils/Address.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/utils/introspection/IERC165.sol#4)
- ~0.8.0 (openzeppelin-solidity/contracts/utils/introspection/IERC165.sol#4)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCallWithValue(address,bytes,uint256) (openzeppelin-solidity/contracts/utils/Address.sol#109-115) is never used and should be removed

Address.functionDelegateCall(address,bytes) (openzeppelin-solidity/contracts/utils/Address.sol#169-171) is never used and should be removed

Address.functionStaticCall(address,bytes) (openzeppelin-solidity/contracts/utils/Address.sol#142-144) is never used and should be removed

Figure 19: contracts/airdrop/contracts_AirdropFlashLoan_slither

Compilation warnings/errors on contracts/airdrop/AirdropReceiver.sol:

Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value) or libraries.

>>> contracts/airdrop/AirdropReceiver.sol:28:1

28 contract AirdropReceiver is ERC721Enumerable, ERC721Holder, ERC1155Holder, Initializable, ReentrancyGuard {
| ^ (Relevant source part starts here and spans across multiple lines).

AirdropReceiver (contracts/airdrop/AirdropReceiver.sol#28-314) is an upgradable contract that does not protect its initialize functions: AirdropReceiver.initialize(address) (contracts/airdrop/AirdropReceiver.sol#91-117) AirdropReceiver.unwrap(address) (contracts/airdrop/AirdropReceiver.sol#119-130)

delete the contract with: AirdropReceiver.wrap(address,address,address,uint256) (contracts/airdrop/AirdropReceiver.sol#91-117) AirdropReceiver.unwrap(address) (contracts/airdrop/AirdropReceiver.sol#119-130)

Reentrancy in AirdropReceiver.unwrap(address): (contracts/airdrop/AirdropReceiver.sol#119-130):

External calls:

- _transferNFTTransferWrapper(address(this),_receiver,wrappedNft,wrappedNftId) (contracts/airdrop/AirdropReceiver.sol#122)
- _nftTransferWrapper.functionDelegateCall(abi.encodeWithSelector(INftWrapper._nftTransferWrapper.selector),_sender,_recipient,_nftCollateralContract,_nftCollateralId)

d) (contracts/airdrop/AirdropReceiver.sol#122-221)

- (success,returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)

State variables written after the calls:

- nftTransferWrapper = address(0) (contracts/airdrop/AirdropReceiver.sol#129)
- wrappedNftId = 0 (contracts/airdrop/AirdropReceiver.sol#127)
- wrappedNftId = 0 (contracts/airdrop/AirdropReceiver.sol#128)

Reentrancy in AirdropReceiver.wrap(address,address,uint256) (contracts/airdrop/AirdropReceiver.sol#91-117):

External calls:

- _transferNFTTransferWrapper(_from,address(this),_nftCollateralContract._nftCollateralId) (contracts/airdrop/AirdropReceiver.sol#108)
- _nftTransferWrapper.functionDelegateCall(abi.encodeWithSelector(INftWrapper._nftTransferWrapper.selector),_sender,_recipient,_nftCollateralContract,_nftCollateralId)

d) (contracts/airdrop/AirdropReceiver.sol#108-111)

- (success,returnData) = target.delegatecall(data) (openzeppelin-solidity/contracts/utils/Address.sol#186)

State variables written after the calls:

- wrappedNft = nftCollateralContract (contracts/airdrop/AirdropReceiver.sol#111)

Reentrancy in AirdropReceiver.unwrap(address) (contracts/airdrop/AirdropReceiver.sol#119-130):

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#389)' in ERC721._checkOnERC721Received(address,address,uint256,bytes).retval = IERC721Receiver.onERC721Received.selector('openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#390)' Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,uint256,bytes).reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#392)' Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,uint256,bytes).reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#392)' Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,uint256,bytes).reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#392)' Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#389)' in ERC721._checkOnERC721Received(address,address,uint256,bytes).retval = IERC721Receiver.onERC721Received.selector('openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#390)' Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,uint256,bytes).reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#392)' Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason' (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#391)' in ERC721._checkOnERC721Received(address,address,uint256,bytes).reason.length == 0 (openzeppelin-solidity/contracts/token/ERC721/ERC721.sol#392)' Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-use-of-local-variables

Reentrancy in AirdropReceiver.unwrap(address) (contracts/airdrop/AirdropReceiver.sol#119-130):

Figure 20: contracts/airdrop/contracts_AirdropReceiver_slither

According to the test results, the findings found by these tools were considered false positives. All relevant findings were reviewed by the auditors, and relevant findings were addressed in the report as security

AUTOMATED TESTING

concerns.

THANK YOU FOR CHOOSING
 HALBORN