// HALBORN

NFTfi - eth.nftfi - Collection Offer

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **August 16th, 2022 - August 24th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	08/14/2022	Grzegorz Trawinski
0.2	Document Update	08/24/2022	Grzegorz Trawinski
0.3	Draft Review	08/24/2022	Gabi Urrutia
1.0	Remediation Plan	10/14/2022	Grzegorz Trawinski
1.1	Remediation Plan Review	10/14/2022	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
Grzegorz Trawinski	Halborn	Grzegorz.Trawinski@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

NFTfi engaged Halborn to conduct a security audit on their smart contracts beginning on August 16th, 2022 and ending on August 24th, 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 some security risks that were 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 bridge 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 (Brownie, Remix IDE, Visual Studio Code)

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.

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

1.4 SCOPE

IN-SCOPE:

The security assessment was scoped to the Collection Offer smart contract, NFTfi audit-collection-offer-28-07-2022 branch:

- DirectLoanFixedCollectionOffer.sol

Commit ID: 6c6fce28f47128d0410fb195142388bdf9e72763

The NFTfi's loan solution was already tested twice:

- in November 2021 (NFTfi develop-v2.1-audit branch contracts)
- in March 2022 (NFTfi v2.2-07-03-2022-audit)

OUT-OF-SCOPE:

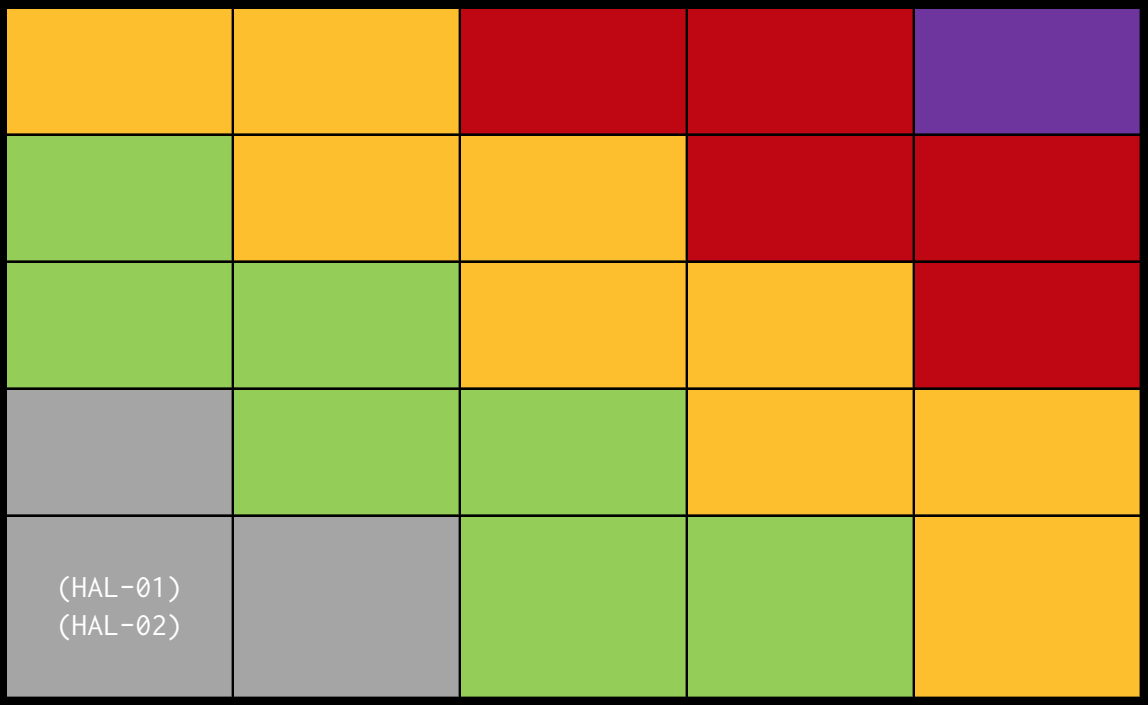
Other smart contracts in the repository, external libraries and economical attacks.

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	0	2

LIKELIHOOD

IMPACT



SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - GAS OVER-CONSUMPTION IN LOOPS	Informational	ACKNOWLEDGED
HAL-02 - SOLC 0.8.4 COMPILER VERSION CONTAINS MULTIPLE BUGS	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) GAS OVER-CONSUMPTION IN LOOPS - INFORMATIONAL

Description:

In all the loops, the counter variable is incremented using `i++`. It is known that, in loops, using `++i` costs less gas per iteration than `i++`.

Code Location:

DirectLoanBaseMinimal.sol
- Line 305: `for (uint256 i = 0; i < _permittedErc20s.length; i++){`
- Line 374: `for (uint256 i = 0; i < _erc20s.length; i++){`

Proof of Concept:

For example, based in the following test contract:

Listing 1: Test.sol

```
1 //SPDX-License-Identifier: MIT
2 pragma solidity 0.8.9;
3
4 contract test {
5     function postiincrement(uint256 iterations) public {
6         for (uint256 i = 0; i < iterations; i++) {
7         }
8     }
9     function preiincrement(uint256 iterations) public {
10        for (uint256 i = 0; i < iterations; ++i) {
11        }
12    }
13 }
```

We can see the difference in the gas costs:

```
>>> test_contract.postiincrement(1)
Transaction sent: 0xlecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 44
  test.postiincrement confirmed   Block: 13622335   Gas used: 21620 (0.32%)

<Transaction '0xlecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b'>
>>> test_contract.preiincrement(1)
Transaction sent: 0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 45
  test.preiincrement confirmed   Block: 13622336   Gas used: 21593 (0.32%)

<Transaction '0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a'>
>>> test_contract.postiincrement(10)
Transaction sent: 0x98c04430526a59balcf947c114b62666a4417165947d31bf300cd6ae68328033
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 46
  test.postiincrement confirmed   Block: 13622337   Gas used: 22673 (0.34%)

<Transaction '0x98c04430526a59balcf947c114b62666a4417165947d31bf300cd6ae68328033'>
>>> test_contract.preiincrement(10)
Transaction sent: 0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20elde05
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 47
  test.preiincrement confirmed   Block: 13622338   Gas used: 22601 (0.34%)

<Transaction '0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20elde05'>
```

Risk Level:

Likelihood - 1

Impact - 1

Recommendation:

It is recommended to use `++i` instead of `i++` to increment the value of an `uint` variable inside a loop to save some gas. This is not applicable outside of loops.

Remediation Plan:

ACKNOWLEDGED: The `\client team` acknowledged this issue.

3.2 (HAL-02) SOLC 0.8.4 COMPILER VERSION CONTAINS MULTIPLE BUGS - INFORMATIONAL

Description:

Presently, the smart contracts have configured the floating pragma set to ^0.8.0 or fixed pragma to 0.8.4 (e.g. DirectLoanFixedCollectionOffer.sol). The latest solidity compiler version 0.8.16 fixed important bugs in the compiler. The version 0.8.4 is missing all these fixes: 0.8.9, 0.8.13, 0.8.14, 0.8.15, 0.8.16.
The official Solidity's recommendations are: when deploying contracts, you should use the latest released version of Solidity. Apart from exceptional cases, only the latest version receives security fixes.

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to set the floating pragma at least to ^0.8.16 version.

Remediation Plan:

ACKNOWLEDGED: The \client team acknowledged this issue.

MANUAL TESTING

Halborn performed several manual tests in the DirectLoanFixedCollectionOffer .sol contract:

```
Calling -> testRealsies.approve(directLoanFixedCollectionOffer, 4_000_000_000_000_000_000_000, {'from': lender})
Transaction sent: 0xe10821f210ce5cdb2e8faea4b380640ba89d95e450e6cf0212f9574e42bdf012
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  TestRealsies.approve confirmed   Block: 40   Gas used: 44213 (0.37%)

Calling -> testRealsies.approve(directLoanFixedCollectionOffer, 4_000_000_000_000_000_000_000, {'from': borrower})
Transaction sent: 0x6e3081afaf722026cb71f883e30dd71623fc369fdc1eb28cc1473c5ffe6c6cdb
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  TestRealsies.approve confirmed   Block: 41   Gas used: 44213 (0.37%)

Before acceptOffer
testRealsies.balance(borrower) 2,000,000,000,000,000,000
testRealsies.balance(lender) 2,000,000,000,000,000,000
testRealsies.balance(directLoanFixedCollectionOffer) 0
testCryptoKitties.balance(borrower) 2
testCryptoKitties.balance(lender) 1
testCryptoKitties.balance(directLoanFixedCollectionOffer) 0
borrower 0x33A4622B82D4c04a53e170c638B944ce27cffce3
lender 0xdeED260B7CcF1546d79baeA5Fd845663865B8873
directLoanFixedCollectionOffer 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d
testCryptoKitties.ownerOf(0) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testCryptoKitties.ownerOf(1) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testCryptoKitties.ownerOf(2) 0xdeED260B7CcF1546d79baeA5Fd845663865B8873
directLoanFixedCollectionOffer.getPayoffAmount(0) 0
Calling -> directLoanFixedCollectionOffer.acceptOffer(offer, signature, borrowerSettings, {'from': borrower})
Transaction sent: 0x7ce6372e1c373ef1b4fe8d4ff085079b87e51fd01d41fd8896e0e485cd560091
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  DirectLoanFixedCollectionOffer.acceptOffer confirmed   Block: 42   Gas used: 465815 (3.88%)

After acceptOffer
testRealsies.balance(borrower) 3,000,000,000,000,000,000
testRealsies.balance(lender) 1,000,000,000,000,000,000
testRealsies.balance(directLoanFixedCollectionOffer) 0
testCryptoKitties.balance(borrower) 1
testCryptoKitties.balance(lender) 1
testCryptoKitties.balance(directLoanFixedCollectionOffer) 1
borrower 0x33A4622B82D4c04a53e170c638B944ce27cffce3
lender 0xdeED260B7CcF1546d79baeA5Fd845663865B8873
directLoanFixedCollectionOffer 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d
testCryptoKitties.ownerOf(0) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testCryptoKitties.ownerOf(1) 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d
testCryptoKitties.ownerOf(2) 0xdeED260B7CcF1546d79baeA5Fd845663865B8873
directLoanFixedCollectionOffer.getPayoffAmount(0) 1,500,000,000,000,000,000
directLoanFixedCollectionOffer.loanIdToLoan(0) (1000000000000000000, 1500000000000000000, 1, '0x42E8D004c84E6B5B
19A04A4A', 1662448472, '0xb6286fAFd0451320ad6A8143089b216C2152c025', '0x33A4622B82D4c04a53e170c638B944ce27cffce3
Calling -> wrapCollateralTX = directLoanFixedCollectionOffer.wrapCollateral(1, {'from': borrower})
Transaction sent: 0x214fe9a3291b02635f71b0ba0fbc7c2e463728cfa5537b18fa6570bfd9e767cd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  DirectLoanFixedCollectionOffer.wrapCollateral confirmed   Block: 43   Gas used: 484299 (4.04%)

/Users/gt/.local/pipx/venvs/eth-brownie/lib/python3.8/site-packages/brownie/network/event.py:243: UserWarning: 0
opics for the given ABI - this is usually because an event argument is not marked as indexed
  warnings.warn(f"{address}: {exc}")
```

MANUAL TESTING

```
testCryptoKitties.balance(wrapperAddress) 1
testCryptoKitties.ownerOf(1) 0xD8D5D5e6C6fBdDB81E6656084d4F5759497CE41c
testRealsies.balance(borrower) 3,000,000,000,000,000,000
testRealsies.balance(lender) 1,000,000,000,000,000,000
testRealsies.balance(directLoanFixedCollectionOffer) 0
testCryptoKitties.balance(borrower) 1
testCryptoKitties.balance(lender) 1
testCryptoKitties.balance(directLoanFixedCollectionOffer) 0
borrower 0x33A4622B82D4c04a53e170c638B944ce27cffce3
lender 0xdeED260B7CcF1546d79baeA5Fd845663865B8873
directLoanFixedCollectionOffer 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d
testCryptoKitties.ownerOf(0) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testCryptoKitties.ownerOf(1) 0xD8D5D5e6C6fBdDB81E6656084d4F5759497CE41c
testCryptoKitties.ownerOf(2) 0xdeED260B7CcF1546d79baeA5Fd845663865B8873
directLoanFixedCollectionOffer.getPayoffAmount(0) 1,500,000,000,000,000,000
directLoanFixedCollectionOffer.loanIdToLoan(0) (1000000000000000000, 1500000000000000000, 52016163199644109405817936449472455737654517453105
559D3b5CE7947AADb9E0bc', 604800, 0, 500, '0x420b1099B9eF5baba6D92029594eF45E19A04A4A', 1662448472, '0xD8D5D5e6C6fBdDB81E6656084d4F5759497CE
testCryptoKitties.balance(wrapperAddress) 1
testCryptoKitties.ownerOf(1) 0xD8D5D5e6C6fBdDB81E6656084d4F5759497CE41c
Calling -> directLoanFixedCollectionOffer.payBackLoan(1, {'from': borrower})
Calling -> chain.sleep(day * 10)
Calling -> chain.mine(1)
Calling -> directLoanFixedCollectionOffer.liquidateOverdueLoan(1, {'from': lender})
Transaction sent: 0x04674e3c7f395261c2f27b0d6ec1b292b1498b44717b611022496fe0dd68a740
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  DirectLoanFixedCollectionOffer.liquidateOverdueLoan confirmed (NFT not successfully transferred)   Block: 45   Gas used: 143377 (1.19%)

testRealsies.balance(borrower) 3,000,000,000,000,000,000
testRealsies.balance(lender) 1,000,000,000,000,000,000
testRealsies.balance(directLoanFixedCollectionOffer) 0
testCryptoKitties.balance(borrower) 1
testCryptoKitties.balance(lender) 1
testCryptoKitties.balance(directLoanFixedCollectionOffer) 0
borrower 0x33A4622B82D4c04a53e170c638B944ce27cffce3
lender 0xdeED260B7CcF1546d79baeA5Fd845663865B8873
directLoanFixedCollectionOffer 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d
testCryptoKitties.ownerOf(0) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testCryptoKitties.ownerOf(1) 0xD8D5D5e6C6fBdDB81E6656084d4F5759497CE41c
testCryptoKitties.ownerOf(2) 0xdeED260B7CcF1546d79baeA5Fd845663865B8873
directLoanFixedCollectionOffer.getPayoffAmount(0) 1,500,000,000,000,000,000
directLoanFixedCollectionOffer.loanIdToLoan(0) (1000000000000000000, 1500000000000000000, 52016163199644109405817936449472455737654517453105
559D3b5CE7947AADb9E0bc', 604800, 0, 500, '0x420b1099B9eF5baba6D92029594eF45E19A04A4A', 1662448472, '0xD8D5D5e6C6fBdDB81E6656084d4F5759497CE
testCryptoKitties.balance(wrapperAddress) 1
testCryptoKitties.ownerOf(1) 0xD8D5D5e6C6fBdDB81E6656084d4F5759497CE41c
```

```
Calling -> directLoanFixedCollectionOffer.acceptOffer(offer, signature, borrowerSettings, {'from': borrower})
Transaction sent: 0x36f68b890347cb7d11f69d3d57269ede48ce05e8ecb42dec1b59f11d9d061e48
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  DirectLoanFixedCollectionOffer.acceptOffer confirmed   Block: 87   Gas used: 480721 (4.01%)

After acceptOffer
testRealsies.balance(borrower) 3,000,000,000,000,000,000
testRealsies.balance(lender) 1,000,000,000,000,000,000
testRealsies.balance(directLoanFixedCollectionOffer) 0
testGaspMasks.balance(borrower) 1
testGaspMasks.balance(lender) 1
testGaspMasks.balance(directLoanFixedCollectionOffer) 1
borrower 0x33A4622B82D4c04a53e170c638B944ce27cffce3
lender 0x1e4d9879D4211953bCe0e94B4E9941a67d4508b6
directLoanFixedCollectionOffer 0x8b1B440724DCe2EE9779B58af841Ec59F545838B
testGaspMasks.ownerOf(0) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testGaspMasks.ownerOf(1) 0x8b1B440724DCe2EE9779B58af841Ec59F545838B
testGaspMasks.ownerOf(2) 0x1e4d9879D4211953bCe0e94B4E9941a67d4508b6
directLoanFixedCollectionOffer.getPayoffAmount(0) 1,500,000,000,000,000,000
directLoanFixedCollectionOffer.loanIdToLoan(0) (100000000000000000, 1500000000000000000, 1, '0xFB87C3600960385516
C1F63130', 1663312618, '0x0e19E87b363DD7a3af5c45C95ab6885367251B94', '0x33A4622B82D4c04a53e170c638B944ce27cffce3')
Calling -> wrapCollateralTX = directLoanFixedCollectionOffer.wrapCollateral(1, {'from': borrower})
Transaction sent: 0xdac90ab49e80a4dbdb81b9283ae52006ca5eb499cdbda132c74de8324fd54faf
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  DirectLoanFixedCollectionOffer.wrapCollateral confirmed   Block: 88   Gas used: 493842 (4.12%)

/Users/gt/.local/pipx/venvs/eth-brownie/lib/python3.8/site-packages/brownie/network/event.py:243: UserWarning: 0x3
opics for the given ABI - this is usually because an event argument is not marked as indexed
  warnings.warn(f"{address}: {exc}")
After wrapCollateral
testRealsies.balance(borrower) 3,000,000,000,000,000,000
testRealsies.balance(lender) 1,000,000,000,000,000,000
testRealsies.balance(directLoanFixedCollectionOffer) 0
testGaspMasks.balance(borrower) 1
testGaspMasks.balance(lender) 1
testGaspMasks.balance(directLoanFixedCollectionOffer) 0
borrower 0x33A4622B82D4c04a53e170c638B944ce27cffce3
lender 0x1e4d9879D4211953bCe0e94B4E9941a67d4508b6
directLoanFixedCollectionOffer 0x8b1B440724DCe2EE9779B58af841Ec59F545838B
testGaspMasks.ownerOf(0) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testGaspMasks.ownerOf(1) 0x304A08948f5b640Fa5CD31E6860d7cbde9b376a4
testGaspMasks.ownerOf(2) 0x1e4d9879D4211953bCe0e94B4E9941a67d4508b6
directLoanFixedCollectionOffer.getPayoffAmount(0) 1,500,000,000,000,000,000
directLoanFixedCollectionOffer.loanIdToLoan(0) (100000000000000000, 1500000000000000000, 4261739625453897835414
3ABe3BbF512D3F58fd3819', 604800, 0, 500, '0x0AC45e945A008D3fc19da8f591be8601C1F63130', 1663312618, '0x304A08948f5b
wrapperId: 426173962545389783541451024351773278454401481728765534505870463047655391736 91
wrapperAddress: 0x304A08948f5b640Fa5CD31E6860d7cbde9b376a4
testGaspMasks.balance(wrapperAddress) 1
testGaspMasks.ownerOf(1) 0x304A08948f5b640Fa5CD31E6860d7cbde9b376a4
Calling -> airdropReceiver.pullAirdrop(testAirdrop, encodedFunctionData, {'from': borrower})
Transaction sent: 0xfb156b70ea767ccfb7c4e035b3456e816daad41210de0acf575e85a11a3419a8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  AirdropReceiver.pullAirdrop confirmed   Block: 89   Gas used: 88646 (0.74%)
```

```
testRealsies.balance(borrower) 3,000,000,000,000,000,000
testRealsies.balance(lender) 1,000,000,000,000,000,000
testRealsies.balance(directLoanFixedCollectionOffer) 0
testGaspMasks.balance(borrower) 1
testGaspMasks.balance(lender) 1
testGaspMasks.balance(directLoanFixedCollectionOffer) 0
borrower 0x33A4622B82D4c04a53e170c638B944ce27cffce3
lender 0x1e4d9879D4211953bCe0e94B4E9941a67d4508b6
directLoanFixedCollectionOffer 0x8b1B440724DCe2EE9779B58af841Ec59F545838B
testGaspMasks.ownerOf(0) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testGaspMasks.ownerOf(1) 0x304A08948f5b640Fa5CD31E6860d7cbde9b376a4
testGaspMasks.ownerOf(2) 0x1e4d9879D4211953bCe0e94B4E9941a67d4508b6
directLoanFixedCollectionOffer.getPayoffAmount(0) 1,500,000,000,000,000,000
directLoanFixedCollectionOffer.loanIdToLoan(0) (100000000000000000, 150000000000000000, 4261739625453897835414
3ABe3BbF512D3F58fd3819', 604800, 0, 500, '0x0AC45e945A008D3fc19da8f591be8601C1F63130', 1663312618, '0x304A08948f
testGaspMasks.balance(wrapperAddress) 1
testGaspMasks.ownerOf(1) 0x304A08948f5b640Fa5CD31E6860d7cbde9b376a4
Calling -> directLoanFixedCollectionOffer.payBackLoan(1, {'from': borrower})
Calling -> chain.sleep(day * 10)
Calling -> chain.mine(1)
Calling -> directLoanFixedCollectionOffer.liquidateOverdueLoan(1, {'from': lender})
Transaction sent: 0x05f71eff67ac28e083585b4f4371be8889565950dc725e58d8ff2daf35ef1a37
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  DirectLoanFixedCollectionOffer.liquidateOverdueLoan confirmed   Block: 91   Gas used: 132861 (1.11%)

testRealsies.balance(borrower) 3,000,000,000,000,000,000
testRealsies.balance(lender) 1,000,000,000,000,000,000
testRealsies.balance(directLoanFixedCollectionOffer) 0
testGaspMasks.balance(borrower) 1
testGaspMasks.balance(lender) 1
testGaspMasks.balance(directLoanFixedCollectionOffer) 0
borrower 0x33A4622B82D4c04a53e170c638B944ce27cffce3
lender 0x1e4d9879D4211953bCe0e94B4E9941a67d4508b6
directLoanFixedCollectionOffer 0x8b1B440724DCe2EE9779B58af841Ec59F545838B
testGaspMasks.ownerOf(0) 0x33A4622B82D4c04a53e170c638B944ce27cffce3
testGaspMasks.ownerOf(1) 0x304A08948f5b640Fa5CD31E6860d7cbde9b376a4
testGaspMasks.ownerOf(2) 0x1e4d9879D4211953bCe0e94B4E9941a67d4508b6
directLoanFixedCollectionOffer.getPayoffAmount(0) 0
directLoanFixedCollectionOffer.loanIdToLoan(0) (0, 0, 0, '0x0000000000000000000000000000000000000000', 0, 0, 0,
00000000000000', '0x0000000000000000000000000000000000000000')
testGaspMasks.balance(wrapperAddress) 1
testGaspMasks.ownerOf(1) 0x304A08948f5b640Fa5CD31E6860d7cbde9b376a4
```

The manual tests were focused on testing the main functions of this contract:

- acceptOffer()
- getPayoffAmount()
- updateMaximumLoanDuration()
- updateAdminFee()
- drainERC20Airdrop()
- setERC20Permit()
- setERC20Permits()
- drainERC721Airdrop()
- drainERC1155Airdrop()
- mintObligationReceipt()
- renegotiateLoan()
- payBackLoan()
- liquidateOverdueLoan()
- pullAirdrop()
- wrapCollateral()

MANUAL TESTING

- `cancelLoanCommitmentBeforeLoanHasBegun()`
- `getWhetherNonceHasBeenUsedForUser()`
- `getERC20Permit()`

No significant issues were found during the manual tests.

AUTOMATED TESTING

5.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the scoped contracts. 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, Slither was run on the all-scoped contracts. 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.

Slither results:

DirectLoanFixedCollectionOffer.sol

```
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#498)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#93)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
                - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,_lender,payoffAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#919)
                - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,loanExtras.revenueSharePartner,revenueShare) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#931
-935)
                - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,owner(),adminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#938)
        - _resolveLoan(_loanId,borrower,loan,loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#500)
                - Address.functionDelegateCall(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_lo
anTerms.nftCollateralContract,_loanTerms.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
                - _loanCoordinator.resolveLoan(_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#997)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
        External calls sending eth:
        - _payBackLoan(_loanId,borrower,lender,loan) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#498)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
        State variables written after the call(s):
        - 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#727-794):
        External calls:
        - IERC20(loan.loanERC20Denomination).safeTransferFrom(borrower,lender,_renegotiationFee - renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#773-777)
        - IERC20(loan.loanERC20Denomination).safeTransferFrom(borrower,owner(),renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#779)
        State variables written after the call(s):
        - loan.loanDuration = _newLoanDuration (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#782)
        - loan.maximumRepaymentAmount = _newMaximumRepaymentAmount (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#783)
Reentrancy in DirectLoanBaseMinimal.liquidateOverdueLoan(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#523-561):
        External calls:
        - _resolveLoan(_loanId,lender,loan,loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#542)
                - Address.functionDelegateCall(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_lo
anTerms.nftCollateralContract,_loanTerms.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
                - _loanCoordinator.resolveLoan(_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#997)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
        State variables written after the call(s):
        - delete loanIdToLoan[_loanId] (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#559)
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)
        State variables written after the call(s):
        - delete loanIdToLoan[_loanId].borrower (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#423)
Reentrancy in DirectLoanBaseMinimal.wrapCollateral(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#608-617):
        External calls:
        - (instance,receiverId) = LoanAirdropUtils.wrapCollateral(_loanId,loan,hub) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#615)
        State variables written after the call(s):
        - _escrowTokens[instance][receiverId] += 1 (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#616)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

DirectLoanBaseMinimal._renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes).renegotiationAdminFee (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#760) is a local
variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

DirectLoanBaseMinimal._transferNFT(LoanData.LoanTerms,address,address) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#878-894) ignores return value by Address.functionDelegateCa
ll(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loanTerms.nftCollateralContract,_loanTerms.nft
CollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
LoanAirdropUtils._transferNFT(LoanData.LoanTerms,address,address) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#183-199) ignores return value by Address.functionDelegateCall(_loan.n
ftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loan.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loan.nftCollateralContract,_loan.nftCollateralId),NFT not successf
ully transferred) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#188-198)
LoanAirdropUtils._transferNFTtoAirdropReceiver(LoanData.LoanTerms,address,address) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#209-225) ignores return value by Address.functionDel
egateCall(_loan.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loan.nftCollateralWrapper).wrapAirdropReceiver.selector,_airdropReceiverInstance,_loan.nftCollateralContract,_loan.nf
tCollateralId,_airdropBeneficiary),NFT was not successfully migrated) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#214-224)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
```

```
Variable 'ECDSA.tryRecover(bytes32,bytes).r (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#62)' in ECDSA.tryRecover(bytes32,bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#57-86) potentially used before declaration: r = mload(uint256)(signature + 0x20) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#79)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in DirectLoanBaseMinimal._createLoan(bytes32,LoanData.LoanTerms,LoanData.LoanExtras,address,address,address) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#807-820):
        External calls:
        - _transferNFT(_loanTerms,_borrower,address(this)) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#817)
                - Address.functionDelegateCall(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loanTerms.nftCollateralContract,_loanTerms.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
        - _createLoanNoNftTransfer(_loanType,_loanTerms,_loanExtras,_borrower,_lender,_referrer) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#819)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#93)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
                - IERC20(_loanTerms.loanERC20Denomination).safeTransferFrom(_lender,_referrer,referralfee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#851)
                - IERC20(_loanTerms.loanERC20Denomination).safeTransferFrom(_lender,_borrower,principalAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#854)
                - loanId = loanCoordinator.registerLoan(_lender,_loanType) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#861)
        External calls sending eth:
        - _createLoanNoNftTransfer(_loanType,_loanTerms,_loanExtras,_borrower,_lender,_referrer) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#819)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
        State variables written after the call(s):
        - _createLoanNoNftTransfer(_loanType,_loanTerms,_loanExtras,_borrower,_lender,_referrer) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#819)
                - loanIdToLoan[loanId] = _loanTerms (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#865)
        - _createLoanNoNftTransfer(_loanType,_loanTerms,_loanExtras,_borrower,_lender,_referrer) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#819)
                - loanIdToLoanExtras[loanId] = _loanExtras (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#866)
Reentrancy in DirectLoanBaseMinimal._createLoanNoNftTransfer(bytes32,LoanData.LoanTerms,LoanData.LoanExtras,address,address,address) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#833-869):
        External calls:
        - IERC20(_loanTerms.loanERC20Denomination).safeTransferFrom(_lender,_referrer,referralfee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#851)
        - IERC20(_loanTerms.loanERC20Denomination).safeTransferFrom(_lender,_borrower,principalAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#854)
        - loanId = loanCoordinator.registerLoan(_lender,_loanType) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#861)
        State variables written after the call(s):
        - loanIdToLoan[loanId] = _loanTerms (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#865)
        - loanIdToLoanExtras[loanId] = _loanExtras (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#866)
Reentrancy in DirectLoanBaseMinimal.liquidateOverdueLoan(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#523-561):
        External calls:
        - _resolveLoan(_loanId,lender,loan,loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#542)
                - Address.functionDelegateCall(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loanTerms.nftCollateralContract,_loanTerms.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
                - _loanCoordinator.resolveLoan(_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#997)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
        State variables written after the call(s):
        - delete loanIdToLoanExtras[_loanId] (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#560)
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#498)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#93)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
                - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,_lender,payoffAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#919)
                - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,loanExtras.revenueSharePartner,revenueShare) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#931-935)
                - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,owner(),adminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#938)
        - _resolveLoan(_loanId,borrower,loan,loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#500)
                - Address.functionDelegateCall(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loanTerms.nftCollateralContract,_loanTerms.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
                - _loanCoordinator.resolveLoan(_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#997)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
        External calls sending eth:
        - _payBackLoan(_loanId,borrower,lender,loan) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#498)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
        State variables written after the call(s):
        - _resolveLoan(_loanId,borrower,loan,loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#500)
                - loanRepaidOrLiquidated[_loanId] = true (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#992)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in DirectLoanFixedCollectionOffer._acceptOffer(LoanData.LoanTerms,LoanData.LoanExtras,LoanData.Offer,LoanData.Signature) (contracts/loans/direct/loanTypes/DirectLoanFixedCollectionOffer.sol#93-124):
        External calls:
        - loanId = _createLoan(LOAN_TYPE(),_loanTerms,_loanExtras,msg.sender,_signature.signer,_offer.referrer) (contracts/loans/direct/loanTypes/DirectLoanFixedCollectionOffer.sol#113-120)
                - Address.functionDelegateCall(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loanTerms.nftCollateralContract,_loanTerms.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#93)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
                - IERC20(_loanTerms.loanERC20Denomination).safeTransferFrom(_lender,_referrer,referralfee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#851)
                - IERC20(_loanTerms.loanERC20Denomination).safeTransferFrom(_lender,_borrower,principalAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#854)
                - loanId = loanCoordinator.registerLoan(_lender,_loanType) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#861)
        External calls sending eth:
        - loanId = _createLoan(LOAN_TYPE(),_loanTerms,_loanExtras,msg.sender,_signature.signer,_offer.referrer) (contracts/loans/direct/loanTypes/DirectLoanFixedCollectionOffer.sol#113-120)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
        Event emitted after the call(s):
        - LoanStarted(loanId,msg.sender,_signature.signer,_loanTerms,_loanExtras) (contracts/loans/direct/loanTypes/DirectLoanFixedCollectionOffer.sol#123)
Reentrancy in DirectLoanFixedOffer._acceptOffer(LoanData.LoanTerms,LoanData.LoanExtras,LoanData.Offer,LoanData.Signature) (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#150-182):
        External calls:
        - loanId = _createLoan(LOAN_TYPE(),_loanTerms,_loanExtras,msg.sender,_signature.signer,_offer.referrer) (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#171-178)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#93)
                - Address.functionDelegateCall(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loanTerms.nftCollateralContract,_loanTerms.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
                - IERC20(_loanTerms.loanERC20Denomination).safeTransferFrom(_lender,_referrer,referralfee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#851)
                - IERC20(_loanTerms.loanERC20Denomination).safeTransferFrom(_lender,_borrower,principalAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#854)
                - loanId = loanCoordinator.registerLoan(_lender,_loanType) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#861)
        External calls sending eth:
        - loanId = _createLoan(LOAN_TYPE(),_loanTerms,_loanExtras,msg.sender,_signature.signer,_offer.referrer) (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#171-178)
                - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
        Event emitted after the call(s):
        - LoanStarted(loanId,msg.sender,_signature.signer,_loanTerms,_loanExtras) (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#181)
Reentrancy in DirectLoanBaseMinimal._payBackLoan(uint32,address,address,LoanData.LoanTerms) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#907-954):
        External calls:
        - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,_lender,payoffAmount) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#919)
        - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,loanExtras.revenueSharePartner,revenueShare) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#931-935)
        - IERC20(_loan.loanERC20Denomination).safeTransferFrom(msg.sender,owner(),adminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#938)
        Event emitted after the call(s):
        - LoanRepaid(_loanId,_borrower,_lender,_loan.loanPrincipalAmount,_loan.nftCollateralId,payoffAmount,adminFee,revenueShare,loanExtras.revenueSharePartner,_loan.nftCollateralContract,_loan.loanERC20Denomination) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#941-951)
Reentrancy in DirectLoanBaseMinimal._renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#727-794):
        External calls:
        - IERC20(loan.loanERC20Denomination).safeTransferFrom(borrower,lender,_renegotiationFee - renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#773-777)
        - IERC20(loan.loanERC20Denomination).safeTransferFrom(borrower,owner(),renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#779)
        Event emitted after the call(s):
        - LoanRenegotiated(_loanId,borrower,lender,_newLoanDuration,_newMaximumRepaymentAmount,_renegotiationFee,renegotiationAdminFee) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#785-793)
Reentrancy in DirectLoanBaseMinimal.liquidateOverdueLoan(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#523-561):
        External calls:
        - _resolveLoan(_loanId,lender,loan,loanCoordinator) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#542)
                - Address.functionDelegateCall(_loanTerms.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loanTerms.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loanTerms.nftCollateralContract,_loanTerms.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#883-893)
                - _loanCoordinator.resolveLoan(_loanId) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#997)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
        Event emitted after the call(s):
        - LoanLiquidated(_loanId,borrower,lender,loan.loanPrincipalAmount,loan.nftCollateralId,loanMaturityDate,block.timestamp,loan.nftCollateralContract) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#545-554)
Reentrancy in LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#65-132):
        External calls:
        - _transferNFT(_loan,address(this),address(airdropFlashLoan)) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#98)
                - Address.functionDelegateCall(_loan.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loan.nftCollateralWrapper).transferNFT.selector,_sender,_recipient,_loan.nftCollateralContract,_loan.nftCollateralId),NFT not successfully transferred) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#188-198)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
        - airdropFlashLoan.pullAirdrop(_loan.nftCollateralContract,_loan.nftCollateralId,_loan.nftCollateralWrapper,_target,_data,_nftAirdrop,_nftAirdropId,_is1155,_nftAirdropAmount,borrower) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#100-111)
        Event emitted after the call(s):
        - AirdropPulledFlashloan(_loanId,borrower,_loan.nftCollateralId,_loan.nftCollateralContract,_target,_data) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#124-131)
Reentrancy in LoanAirdropUtils.wrapCollateral(uint32,LoanData.LoanTerms,INftfiHub) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#134-174):
        External calls:
        - (instance,receiverId) = factory.createAirdropReceiver(address(this)) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#157)
        - _transferNFTToAirdropReceiver(_loan,instance,borrower) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#160)
                - Address.functionDelegateCall(_loan.nftCollateralWrapper,abi.encodeWithSelector(INftWrapper(_loan.nftCollateralWrapper).wrapAirdropReceiver.selector,_airdropReceiverInstance,_loan.nftCollateralContract,_loan.nftCollateralId,_airdropBeneficiary),NFT was not successfully migrated) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#214-224)
                - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
        Event emitted after the call(s):
        - CollateralWrapped(_loanId,borrower,_loan.nftCollateralId,_loan.nftCollateralContract,receiverId,instance) (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#162-169)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
DirectLoanBaseMinimal.liquidateOverdueLoan(uint32) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#523-561) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp > loanMaturityDate,Loan is not overdue yet) (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#538)
DirectLoanFixedCollectionOffer._acceptOffer(LoanData.LoanTerms,LoanData.LoanExtras,LoanData.Offer,LoanData.Signature) (contracts/loans/direct/loanTypes/DirectLoanFixedCollectionOffer.sol#93-124) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_loanTerms.nftCollateralContract != bundle,Collateral cannot be bundle) (contracts/loans/direct/loanTypes/DirectLoanFixedCollectionOffer.sol#111)
DirectLoanFixedOffer._acceptOffer(LoanData.LoanTerms,LoanData.LoanExtras,LoanData.Offer,LoanData.Signature) (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#150-182) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_loanTerms.nftCollateralContract != bundle,Collateral cannot be bundle) (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#169)
LoanChecksAndCalculations.payBackChecks(uint32,INftfiHub) (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#28-42) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= (uint256(loanStartTime) + uint256(loanDuration)),Loan is expired) (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#41)
LoanChecksAndCalculations.renegotiationChecks(LoanData.LoanTerms,uint32,uint32,uint256,uint256,INftfiHub) (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#96-139) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= (uint256(_loan.loanStartTime) + _newLoanDuration),New duration already expired) (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#119)
NFTfiSigningUtils.isValidBorrowerSignature(LoanData.ListingTerms,LoanData.Signature,address) (contracts/utils/NFTfiSigningUtils.sol#102-128) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= _signature.expiry,Borrower Signature has expired) (contracts/utils/NFTfiSigningUtils.sol#107)
NFTfiSigningUtils.isValidBorrowerSignatureBundle(LoanData.ListingTerms,IBundleBuilder.BundleElements,LoanData.Signature,address) (contracts/utils/NFTfiSigningUtils.sol#282-230) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= _signature.expiry,Borrower Signature has expired) (contracts/utils/NFTfiSigningUtils.sol#208)
NFTfiSigningUtils.isValidLenderSignature(LoanData.Offer,LoanData.Signature,address) (contracts/utils/NFTfiSigningUtils.sol#299-321) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= _signature.expiry,Lender Signature has expired) (contracts/utils/NFTfiSigningUtils.sol#304)
NFTfiSigningUtils.isValidLenderSignatureBundle(LoanData.Offer,IBundleBuilder.BundleElements,LoanData.Signature,address) (contracts/utils/NFTfiSigningUtils.sol#392-420) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= _signature.expiry,Lender Signature has expired) (contracts/utils/NFTfiSigningUtils.sol#398)
NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature,address) (contracts/utils/NFTfiSigningUtils.sol#479-511) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= _signature.expiry,Renegotiation Signature has expired) (contracts/utils/NFTfiSigningUtils.sol#487)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

ContractKeys.getIdFromStringKey(string) (contracts/utils/ContractKeys.sol#31-38) uses assembly
        - INLINE ASM (contracts/utils/ContractKeys.sol#35-37)
NFTfiSigningUtils.getChainID() (contracts/utils/NFTfiSigningUtils.sol#23-30) uses assembly
        - INLINE ASM (contracts/utils/NFTfiSigningUtils.sol#26-28)
Address.verifyCallResult(bool,bytes,string) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#213-216)
ECDSA.tryRecover(bytes32,bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#57-86) uses assembly
        - INLINE ASM (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#67-71)
        - INLINE ASM (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#78-81)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
        - Version used: ['0.8.4', '^0.8.0', '^0.8.1']
        - 0.8.4 (contracts/airdrop/IAirdropReceiverFactory.sol#3)
        - 0.8.4 (contracts/interfaces/IAirdropFlashLoan.sol#3)
        - 0.8.4 (contracts/interfaces/IBundleBuilder.sol#3)
        - 0.8.4 (contracts/interfaces/IDirectLoanCoordinator.sol#3)
        - 0.8.4 (contracts/interfaces/INftWrapper.sol#3)
        - 0.8.4 (contracts/interfaces/INftfiHub.sol#3)
        - 0.8.4 (contracts/interfaces/IPermittedERC20s.sol#3)
        - 0.8.4 (contracts/interfaces/IPermittedNFTs.sol#3)
        - 0.8.4 (contracts/interfaces/IPermittedPartners.sol#3)
        - 0.8.4 (contracts/loans/BaseLoan.sol#3)
        - 0.8.4 (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#3)
        - 0.8.4 (contracts/loans/direct/loanTypes/DirectLoanFixedCollectionOffer.sol#3)
        - 0.8.4 (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#3)
        - 0.8.4 (contracts/loans/direct/loanTypes/IDirectLoanBase.sol#5)
        - 0.8.4 (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#3)
        - 0.8.4 (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#3)
        - 0.8.4 (contracts/loans/direct/loanTypes/LoanData.sol#3)
        - 0.8.4 (contracts/utils/ContractKeys.sol#3)
        - 0.8.4 (contracts/utils/NFTfiSigningUtils.sol#3)
        - 0.8.4 (contracts/utils/NftReceiver.sol#2)
        - 0.8.4 (contracts/utils/Ownable.sol#3)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/interfaces/IERC1271.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/security/Pausable.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/security/ReentrancyGuard.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC1155/IERC1155Receiver.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC721/IERC721.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC721/IERC721Receiver.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC721/utils/ERC721Holder.sol#4)
        - ^0.8.1 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Context.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Strings.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/SignatureChecker.sol#4)
        - ^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/introspection/IERC165.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionDelegateCall(address,bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionStaticCall(address,bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#60-65) is never used and should be removed
Context._msgData() (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Context.sol#21-23) is never used and should be removed
DirectLoanFixedOffer._acceptOffer(LoanData.LoanTerms,LoanData.LoanExtras,LoanData.Offer,LoanData.Signature) (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#150-182) is never used and should be removed
ECDSA._throwError(ECDSA.RecoverError) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#23-35) is never used and should be removed
ECDSA.recover(bytes32,bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#102-106) is never used and should be removed
ECDSA.recover(bytes32,bytes32,bytes32) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#130-138) is never used and should be removed
ECDSA.recover(bytes32,uint8,bytes32,bytes32) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#181-190) is never used and should be removed
ECDSA.toEthSignedMessageHash(bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#214-216) is never used and should be removed
ECDSA.toTypedDataHash(bytes32,bytes32) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#227-229) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#45-58) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#69-80) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#60-67) is never used and should be removed
Strings.toHexString(uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Strings.sol#40-51) is never used and should be removed
Strings.toHexString(uint256,uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Strings.sol#56-66) is never used and should be removed
Strings.toString(uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Strings.sol#15-35) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/interfaces/IERC1271.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/security/ReentrancyGuard.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC1155/IERC1155Receiver.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC721/IERC721.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC721/IERC721Receiver.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC721/utils/ERC721Holder.sol#4) allows old versions
Pragma version^0.8.1 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Strings.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/SignatureChecker.sol#4) allows old versions
Pragma version^0.8.0 (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/introspection/IERC165.sol#4) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

```
Low level call in Address.sendValue(address,uint256) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Address.sol#191)
Low level call in SignatureChecker.isValidSignatureNow(address,bytes32,bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/SignatureChecker.sol#25-39):
        - (success,result) = signer.staticcall(abi.encodeWithSelector(IERC1271.isValidSignature.selector,hash,signature)) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/SignatureChecker.sol#35-37)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Parameter DirectLoanBaseMinimal.updateMaximumLoanDuration(uint256)._newMaximumLoanDuration (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#321) is not in mixedCase
Parameter DirectLoanBaseMinimal.updateAdminFee(uint16)._newAdminFeeInBasisPoints (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#334) is not in mixedCase
Parameter DirectLoanBaseMinimal.drainERC20Airdrop(address,address)._tokenAddress (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#346) is not in mixedCase
Parameter DirectLoanBaseMinimal.drainERC20Airdrop(address,address)._receiver (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#346) is not in mixedCase
Parameter DirectLoanBaseMinimal.setERC20Permit(address,bool)._erc20 (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#360) is not in mixedCase
Parameter DirectLoanBaseMinimal.setERC20Permit(address,bool)._permit (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#360) is not in mixedCase
Parameter DirectLoanBaseMinimal.setERC20Permits(address[],bool[])._erc20s (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#371) is not in mixedCase
Parameter DirectLoanBaseMinimal.setERC20Permits(address[],bool[])._permits (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#371) is not in mixedCase
Parameter DirectLoanBaseMinimal.drainERC721Airdrop(address,uint256,address)._tokenAddress (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#387) is not in mixedCase
Parameter DirectLoanBaseMinimal.drainERC721Airdrop(address,uint256,address)._tokenId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#388) is not in mixedCase
Parameter DirectLoanBaseMinimal.drainERC721Airdrop(address,uint256,address)._receiver (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#389) is not in mixedCase
Parameter DirectLoanBaseMinimal.drainERC1155Airdrop(address,uint256,address)._tokenAddress (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#405) is not in mixedCase
Parameter DirectLoanBaseMinimal.drainERC1155Airdrop(address,uint256,address)._tokenId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#406) is not in mixedCase
Parameter DirectLoanBaseMinimal.drainERC1155Airdrop(address,uint256,address)._receiver (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#407) is not in mixedCase
Parameter DirectLoanBaseMinimal.mintObligationReceipt(uint32)._loanId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#416) is not in mixedCase
Parameter DirectLoanBaseMinimal.renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes)._loanId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#459) is not in mixedC
ase
Parameter DirectLoanBaseMinimal.renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes)._newLoanDuration (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#460) is not
in mixedCase
Parameter DirectLoanBaseMinimal.renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes)._newMaximumRepaymentAmount (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#46
1) is not in mixedCase
Parameter DirectLoanBaseMinimal.renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes)._renegotiationFee (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#462) is not
 in mixedCase
Parameter DirectLoanBaseMinimal.renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes)._lenderNonce (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#463) is not in m
ixedCase
Parameter DirectLoanBaseMinimal.renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes)._expiry (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#464) is not in mixedC
ase
Parameter DirectLoanBaseMinimal.renegotiateLoan(uint32,uint32,uint256,uint256,uint256,uint256,bytes)._lenderSignature (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#465) is not
in mixedCase
```

```
Parameter DirectLoanBaseMinimal.payBackLoan(uint32)._loanId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#489) is not in mixedCase
Parameter DirectLoanBaseMinimal.liquidateOverdueLoan(uint32)._loanId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#523) is not in mixedCase
Parameter DirectLoanBaseMinimal.pullAirdrop(uint32,address,bytes,address,uint256,bool,uint256)._loanId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#576) is not in mixedCase
Parameter DirectLoanBaseMinimal.pullAirdrop(uint32,address,bytes,address,uint256,bool,uint256)._target (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#577) is not in mixedCase
Parameter DirectLoanBaseMinimal.pullAirdrop(uint32,address,bytes,address,uint256,bool,uint256)._data (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#578) is not in mixedCase
Parameter DirectLoanBaseMinimal.pullAirdrop(uint32,address,bytes,address,uint256,bool,uint256)._nftAirdrop (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#579) is not in mixedCas
e
Parameter DirectLoanBaseMinimal.pullAirdrop(uint32,address,bytes,address,uint256,bool,uint256)._nftAirdropId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#580) is not in mixedC
ase
Parameter DirectLoanBaseMinimal.pullAirdrop(uint32,address,bytes,address,uint256,bool,uint256)._is1155 (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#581) is not in mixedCase
Parameter DirectLoanBaseMinimal.pullAirdrop(uint32,address,bytes,address,uint256,bool,uint256)._nftAirdropAmount (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#582) is not in mi
xedCase
Parameter DirectLoanBaseMinimal.wrapCollateral(uint32)._loanId (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#608) is not in mixedCase
Parameter DirectLoanBaseMinimal.cancelLoanCommitmentBeforeLoanHasBegun(uint256)._nonce (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#635) is not in mixedCase
Parameter DirectLoanBaseMinimal.getWhetherNonceHasBeenUsedForUser(address,uint256)._user (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#672) is not in mixedCase
Parameter DirectLoanBaseMinimal.getWhetherNonceHasBeenUsedForUser(address,uint256)._nonce (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#672) is not in mixedCase
Parameter DirectLoanBaseMinimal.getERC20Permit(address)._erc20 (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#683) is not in mixedCase
Variable DirectLoanBaseMinimal.LOAN_COORDINATOR (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#89) is not in mixedCase
Variable DirectLoanBaseMinimal._nonceHasBeenUsedForUser (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#136) is not in mixedCase
Function DirectLoanFixedCollectionOffer.LOAN_TYPE() (contracts/loans/direct/loanTypes/DirectLoanFixedCollectionOffer.sol#77-79) is not in mixedCase
Parameter DirectLoanFixedOffer.acceptOffer(LoanData.Offer,LoanData.Signature,LoanData.BorrowerSettings)._offer (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#96) is not in mixedC
ase
```

```
Parameter DirectLoanFixedOffer.acceptOffer(LoanData.Offer,LoanData.Signature,LoanData.BorrowerSettings)._signature (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#97) is not in mi
xedCase
Parameter DirectLoanFixedOffer.acceptOffer(LoanData.Offer,LoanData.Signature,LoanData.BorrowerSettings)._borrowerSettings (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#98) is no
t in mixedCase
Function DirectLoanFixedOffer.LOAN_TYPE() (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#120-122) is not in mixedCase
Parameter DirectLoanFixedOffer.getPayoffAmount(uint32)._loanId (contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol#133) is not in mixedCase
Function IDirectLoanBase.LOAN_COORDINATOR() (contracts/loans/direct/loanTypes/IDirectLoanBase.sol#13) is not in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._loanId (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#66) is n
ot in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._loan (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#67) is not
 in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._target (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#68) is n
ot in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._data (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#69) is not
 in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._nftAirdrop (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#70)
is not in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._nftAirdropId (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#71
) is not in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._is1155 (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#72) is n
ot in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._nftAirdropAmount (contracts/loans/direct/loanTypes/LoanAirdropUtils.so
l#73) is not in mixedCase
Parameter LoanAirdropUtils.pullAirdrop(uint32,LoanData.LoanTerms,address,bytes,address,uint256,bool,uint256,INftfiHub)._hub (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#74) is not
in mixedCase
Parameter LoanAirdropUtils.wrapCollateral(uint32,LoanData.LoanTerms,INftfiHub)._loanId (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#135) is not in mixedCase
Parameter LoanAirdropUtils.wrapCollateral(uint32,LoanData.LoanTerms,INftfiHub)._loan (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#136) is not in mixedCase
Parameter LoanAirdropUtils.wrapCollateral(uint32,LoanData.LoanTerms,INftfiHub)._hub (contracts/loans/direct/loanTypes/LoanAirdropUtils.sol#137) is not in mixedCase
Parameter LoanChecksAndCalculations.payBackChecks(uint32,INftfiHub)._loanId (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#28) is not in mixedCase
Parameter LoanChecksAndCalculations.payBackChecks(uint32,INftfiHub)._hub (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#28) is not in mixedCase
Parameter LoanChecksAndCalculations.checkLoanIdValidity(uint32,INftfiHub)._loanId (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#44) is not in mixedCase
Parameter LoanChecksAndCalculations.checkLoanIdValidity(uint32,INftfiHub)._hub (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#44) is not in mixedCase
Parameter LoanChecksAndCalculations.getRevenueSharePercent(address,INftfiHub)._revenueSharePartner (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#61) is not in mixedCase
Parameter LoanChecksAndCalculations.getRevenueSharePercent(address,INftfiHub)._hub (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#61) is not in mixedCase
Parameter LoanChecksAndCalculations.renegotiationChecks(LoanData.LoanTerms,uint32,uint32,uint256,uint256,INftfiHub)._loan (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#97)
is not in mixedCase
```

AUTOMATED TESTING

```
Parameter LoanChecksAndCalculations.renegotiationChecks(LoanData.LoanTerms,uint32,uint32,uint256,uint256,INftfiHub)._loanId (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#98
) is not in mixedCase
Parameter LoanChecksAndCalculations.renegotiationChecks(LoanData.LoanTerms,uint32,uint32,uint256,uint256,INftfiHub)._newLoanDuration (contracts/loans/direct/loanTypes/LoanChecksAndCalculatio
ns.sol#99) is not in mixedCase
Parameter LoanChecksAndCalculations.renegotiationChecks(LoanData.LoanTerms,uint32,uint32,uint256,uint256,INftfiHub)._newMaximumRepaymentAmount (contracts/loans/direct/loanTypes/LoanChecksAnd
Calculations.sol#100) is not in mixedCase
Parameter LoanChecksAndCalculations.renegotiationChecks(LoanData.LoanTerms,uint32,uint32,uint256,uint256,INftfiHub)._lenderNonce (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.s
ol#101) is not in mixedCase
Parameter LoanChecksAndCalculations.renegotiationChecks(LoanData.LoanTerms,uint32,uint32,uint256,uint256,INftfiHub)._hub (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#102)
is not in mixedCase
Parameter LoanChecksAndCalculations.computeRevenueShare(uint256,uint256)._adminFee (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#153) is not in mixedCase
Parameter LoanChecksAndCalculations.computeRevenueShare(uint256,uint256)._revenueShareInBasisPoints (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#153) is not in mixedCase
Parameter LoanChecksAndCalculations.computeAdminFee(uint256,uint256)._interestDue (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#174) is not in mixedCase
Parameter LoanChecksAndCalculations.computeAdminFee(uint256,uint256)._adminFeeInBasisPoints (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#174) is not in mixedCase
Parameter LoanChecksAndCalculations.computeReferralFee(uint256,uint256,address)._loanPrincipalAmount (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#193) is not in mixedCase
Parameter LoanChecksAndCalculations.computeReferralFee(uint256,uint256,address)._referralFeeInBasisPoints (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#194) is not in mixed
Case
Parameter LoanChecksAndCalculations.computeReferralFee(uint256,uint256,address)._referrer (contracts/loans/direct/loanTypes/LoanChecksAndCalculations.sol#195) is not in mixedCase
Parameter ContractKeys.getIdFromStringKey(string)._key (contracts/utils/ContractKeys.sol#31) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignature(LoanData.ListingTerms,LoanData.Signature)._listingTerms (contracts/utils/NFTfiSigningUtils.sol#90) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignature(LoanData.ListingTerms,LoanData.Signature)._signature (contracts/utils/NFTfiSigningUtils.sol#90) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignature(LoanData.ListingTerms,LoanData.Signature,address)._listingTerms (contracts/utils/NFTfiSigningUtils.sol#103) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignature(LoanData.ListingTerms,LoanData.Signature,address)._signature (contracts/utils/NFTfiSigningUtils.sol#104) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignature(LoanData.ListingTerms,LoanData.Signature,address)._loanContract (contracts/utils/NFTfiSigningUtils.sol#105) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignatureBundle(LoanData.ListingTerms,IBundleBuilder.BundleElements,LoanData.Signature)._listingTerms (contracts/utils/NFTfiSigningUtils.sol#191) i
s not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignatureBundle(LoanData.ListingTerms,IBundleBuilder.BundleElements,LoanData.Signature)._bundleElements (contracts/utils/NFTfiSigningUtils.sol#192)
 is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignatureBundle(LoanData.ListingTerms,IBundleBuilder.BundleElements,LoanData.Signature)._signature (contracts/utils/NFTfiSigningUtils.sol#193) is n
ot in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignatureBundle(LoanData.ListingTerms,IBundleBuilder.BundleElements,LoanData.Signature,address)._listingTerms (contracts/utils/NFTfiSigningUtils.so
l#203) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignatureBundle(LoanData.ListingTerms,IBundleBuilder.BundleElements,LoanData.Signature,address)._bundleElements (contracts/utils/NFTfiSigningUtils.
sol#204) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignatureBundle(LoanData.ListingTerms,IBundleBuilder.BundleElements,LoanData.Signature,address)._signature (contracts/utils/NFTfiSigningUtils.sol#2
05) is not in mixedCase
Parameter NFTfiSigningUtils.isValidBorrowerSignatureBundle(LoanData.ListingTerms,IBundleBuilder.BundleElements,LoanData.Signature,address)._loanContract (contracts/utils/NFTfiSigningUtils.so
l#206) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignature(LoanData.Offer,LoanData.Signature)._offer (contracts/utils/NFTfiSigningUtils.sol#287) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignature(LoanData.Offer,LoanData.Signature)._signature (contracts/utils/NFTfiSigningUtils.sol#287) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignature(LoanData.Offer,LoanData.Signature,address)._offer (contracts/utils/NFTfiSigningUtils.sol#300) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignature(LoanData.Offer,LoanData.Signature,address)._signature (contracts/utils/NFTfiSigningUtils.sol#301) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignature(LoanData.Offer,LoanData.Signature,address)._loanContract (contracts/utils/NFTfiSigningUtils.sol#302) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignatureBundle(LoanData.Offer,IBundleBuilder.BundleElements,LoanData.Signature)._offer (contracts/utils/NFTfiSigningUtils.sol#381) is not in mixedCa
se
Parameter NFTfiSigningUtils.isValidLenderSignatureBundle(LoanData.Offer,IBundleBuilder.BundleElements,LoanData.Signature)._bundleElements (contracts/utils/NFTfiSigningUtils.sol#382) is not i
n mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignatureBundle(LoanData.Offer,IBundleBuilder.BundleElements,LoanData.Signature)._signature (contracts/utils/NFTfiSigningUtils.sol#383) is not in mix
edCase
Parameter NFTfiSigningUtils.isValidLenderSignatureBundle(LoanData.Offer,IBundleBuilder.BundleElements,LoanData.Signature,address)._offer (contracts/utils/NFTfiSigningUtils.sol#393) is not in
 mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignatureBundle(LoanData.Offer,IBundleBuilder.BundleElements,LoanData.Signature,address)._bundleElements (contracts/utils/NFTfiSigningUtils.sol#394)
is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignatureBundle(LoanData.Offer,IBundleBuilder.BundleElements,LoanData.Signature,address)._signature (contracts/utils/NFTfiSigningUtils.sol#395) is no
t in mixedCase
Parameter NFTfiSigningUtils.isValidLenderSignatureBundle(LoanData.Offer,IBundleBuilder.BundleElements,LoanData.Signature,address)._loanContract (contracts/utils/NFTfiSigningUtils.sol#396) is
 not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature)._loanId (contracts/utils/NFTfiSigningUtils.sol#458) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature)._newLoanDuration (contracts/utils/NFTfiSigningUtils.sol#459) is not in mixe
dCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature)._newMaximumRepaymentAmount (contracts/utils/NFTfiSigningUtils.sol#460) is n
ot in mixedCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature)._renegotiationFee (contracts/utils/NFTfiSigningUtils.sol#461) is not in mix
edCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature)._signature (contracts/utils/NFTfiSigningUtils.sol#462) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature,address)._loanId (contracts/utils/NFTfiSigningUtils.sol#480) is not in mixed
Case
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature,address)._newLoanDuration (contracts/utils/NFTfiSigningUtils.sol#481) is not
 in mixedCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature,address)._newMaximumRepaymentAmount (contracts/utils/NFTfiSigningUtils.sol#4
82) is not in mixedCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature,address)._renegotiationFee (contracts/utils/NFTfiSigningUtils.sol#483) is no
t in mixedCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature,address)._signature (contracts/utils/NFTfiSigningUtils.sol#484) is not in mi
xedCase
Parameter NFTfiSigningUtils.isValidLenderRenegotiationSignature(uint256,uint32,uint256,uint256,LoanData.Signature,address)._loanContract (contracts/utils/NFTfiSigningUtils.sol#485) is not in
 mixedCase
Parameter NFTfiSigningUtils.getEncodedListing(LoanData.ListingTerms)._listingTerms (contracts/utils/NFTfiSigningUtils.sol#516) is not in mixedCase
Parameter NFTfiSigningUtils.getEncodedOffer(LoanData.Offer)._offer (contracts/utils/NFTfiSigningUtils.sol#535) is not in mixedCase
Parameter NFTfiSigningUtils.getEncodedSignature(LoanData.Signature)._signature (contracts/utils/NFTfiSigningUtils.sol#552) is not in mixedCase
Parameter NftReceiver.supportsInterface(bytes4)._interfaceId (contracts/utils/NftReceiver.sol#48) is not in mixedCase
Parameter Ownable.transferOwnership(address)._newOwner (contracts/utils/Ownable.sol#46) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable DirectLoanBaseMinimal.LOAN_COORDINATOR (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#89) is too similar to DirectLoanBaseMinimal._resolveLoanNoNftTransfer(uint32,LoanD
ata.LoanTerms,IDirectLoanCoordinator)._loanCoordinator (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#988)
Variable DirectLoanBaseMinimal.LOAN_COORDINATOR (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#89) is too similar to DirectLoanBaseMinimal._resolveLoan(uint32,address,LoanData.L
oanTerms,IDirectLoanCoordinator)._loanCoordinator (contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol#969)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

supportsInterface(bytes4) should be declared external:
        - NftReceiver.supportsInterface(bytes4) (contracts/utils/NftReceiver.sol#48-53)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (contracts/utils/Ownable.sol#46-49)
onERC721Received(address,address,uint256,bytes) should be declared external:
        - ERC721Holder.onERC721Received(address,address,uint256,bytes) (OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC721/utils/ERC721Holder.sol#20-27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- The reentrancy issues identified are false positives.
- Some identified issues are related to OppenZepplin's libraries.
- The dangerous comparison instances are false positives.
- The uses inline assembly findings are false positives.
- Pragma usage with old version was reported in SOLC 0.8.4 COMPILER VERSION CONTAINS MULTIPLE BUGS.
- Several informational issues related to solidity naming convention were identified.
- No major issues were found by Slither.

5.2 AUTOMATED SECURITY SCAN

Description:

Halborn used automated security scanners to assist with detection of well-known security issues, and to identify low-hanging fruits on the targets for this engagement. Among the tools used was MythX, a security analysis service for Ethereum smart contracts. MythX performed a scan on all the contracts and sent the compiled results to the analyzers to locate any vulnerabilities.

MythX results:

DirectLoanFixedCollectionOffer.sol

Report for OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/token/ERC20/utils/SafeERC20.sol
https://dashboard.mythx.io/#/console/analyses/c14bdec2-35a3-47eb-af6b-eb212de65516

Line	SWC Title	Severity	Short Description
65	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
77	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

Report for OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/Strings.sol
https://dashboard.mythx.io/#/console/analyses/c14bdec2-35a3-47eb-af6b-eb212de65516

Line	SWC Title	Severity	Short Description
25	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
26	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/=" discovered
30	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
31	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
31	(SWC-110) Assert Violation	Unknown	Out of bounds array access
31	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
32	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/=" discovered
47	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
57	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
57	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
58	(SWC-110) Assert Violation	Unknown	Out of bounds array access
59	(SWC-110) Assert Violation	Unknown	Out of bounds array access
60	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
60	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
60	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "--" discovered
61	(SWC-110) Assert Violation	Unknown	Out of bounds array access

AUTOMATED TESTING

Report for OpenZeppelin/openzeppelin-contracts@4.6.0/contracts/utils/cryptography/ECDSA.sol
https://dashboard.mythx.io/#/console/analyses/c14bdec2-35a3-47eb-af6b-eb212de65516

Line	SWC Title	Severity	Short Description
121	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

Report for contracts/loans/direct/loanTypes/DirectLoanBaseMinimal.sol
https://dashboard.mythx.io/#/console/analyses/c14bdec2-35a3-47eb-af6b-eb212de65516

Line	SWC Title	Severity	Short Description
305	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
306	(SWC-110) Assert Violation	Unknown	Out of bounds array access
374	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
375	(SWC-110) Assert Violation	Unknown	Out of bounds array access
537	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
614	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
616	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
776	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
841	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
848	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
929	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
994	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered

Report for contracts/loans/direct/loanTypes/DirectLoanFixedOffer.sol
https://dashboard.mythx.io/#/console/analyses/c14bdec2-35a3-47eb-af6b-eb212de65516

Line	SWC Title	Severity	Short Description
218	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
223	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

- Majority of identified issues are related to OppenZepplin's libraries.
- The identified arithmetic operations or assert violations are false positives.
- No major issues were found by Mythx.

THANK YOU FOR CHOOSING

// HALBORN