THE ARCADIA GROUP

Security Audit Report

FXDX

Revision history

Date	Commit
4/29/2023	#dcb52a72c3be66e9a6de23f9d3b56ddd33b302b7
5/6/2023	#6aa2b8b49c6cca0fd6a4d14f30b05f7689e758e2
5/6/2023	#43371598d93198b40dbef86dff6ee05bde94799f
5/11/2023	#e485d7168f7df0667852ba45e1573d95818743ee
5/11/2023	#bc65460fb90b79aacf1b78ed17a9f1ad1adf8039

PREPARED FOR:

fxdx EXCHANGE

FXDX Exchange

ARCADIA CONTACT INFO

Email: audits@arcadiamgroup.com

Telegram: https://t.me/thearcadiagroup

Table of Contents

Executive Summary

Introduction

FXDX Exchange engaged Arcadia to perform a security audit of their smart contracts within the fxdx-core repository within the FXDXDEX organization. Our review of their codebase occured on the commit hash #**dcb52a72c3be66e9a6de23f9d3b56ddd33b302b7**

Review Team

1. Tuan “Anhnt” Nguyen - Security Researcher and Engineer
2. Joel Farris - Project Manager

Project Background

FXDX is a decentralized spot and perpetuals exchange forked from GMX with modifications in relation to funding fee costs, removal of trading fees, and introduction of keeper contracts.

Coverage

For this audit, we performed research, test coverage, investigation, and review of FXDX followed by issue reporting, along with mitigation and remediation instructions as outlined in this report. The following code repositories are considered in scope for the review.

Files
contracts/core/BasePositionManager.sol
contracts/core/FlpManager.sol
contracts/core/OrderBook.sol
contracts/core/PositionRouter.sol
contracts/core/Router.sol
contracts/core/FeeUtilsV1.sol

contracts/core/FeeUtilsV2.sol
contracts/core/LiquidityRouter.sol
contracts/core/PositionManager.sol
contracts/core/SwapRouter.sol
contracts/core/Vault.sol
contracts/core/VaultUtils.sol
contracts/peripherals/FxdxTimelock.sol
contracts/peripherals/Reader.sol
contracts/peripherals/Timelock.sol
contracts/peripherals/VaultReader.sol
contracts/staking/RewardRouterV2.sol
contracts/oracle/FastPriceFeed.sol
contracts/oracle/VaultPriceFeed.sol
contracts/oracle/PriceFeedTimelock.sol

Methodology

Arcadia completed this security review using various methods, primarily consisting of dynamic and static analysis. This process included a line-by-line analysis of the in-scope contracts, optimization analysis, analysis of key functionalities and limiters, and reference against intended functionality.

The followings are the steps we have performed while auditing the smart contracts:

- Investigating the project and its technical architecture overview through its documentation
- Understanding the overview of the smart contracts, the functions of the contracts, the inheritance, and how the contracts interface with each others thanks to the graph created by Solidity Visual Developer
- Manual smart contract audit:
    - Review the code to find any issue that could be exploited by known attacks listed by Consensys
    - Identifying which existing projects the smart contracts are built upon and what are the known vulnerabilities and remediations to the existing projects

  - Line-by-line manual review of the code to find any algorithmic and arithmetic related vulnerabilities compared to what should be done based on the project's documentation
  - Find any potential code that could be refactored to save gas
  - Run through the unit-tests and test-coverage if exists
- Static Analysis:
  - Scanning for vulnerabilities in the smart contracts using Static Code Analysis Software
  - Making a static analysis of the smart contracts using Slither
- Fuzzing
  - Arcadia assisted in writing and ensuring full coverage of fuzzing implementations
- Additional review: a follow-up review is done when the smart contracts have any new update. The follow-up is done by reviewing all changes compared to the audited commit revision and its impact to the existing source code and found issues.

Summary

There were **14** issues found, **0** of which were deemed to be 'critical', and **0** of which were rated as 'high'. At the end of These issues were found throughout the review of a rapidly changing codebase and not a final static point in time.

Severity Rating	Number of Original Occurrences	Number of Remaining Occurrences
CRITICAL	0	
HIGH	0	
MEDIUM	2	1
LOW	6	3
INFORMATIONAL	6	3

Findings in Manual Audit

(FD-1) Function `handleRewards` allows swapping with zero minimum output.

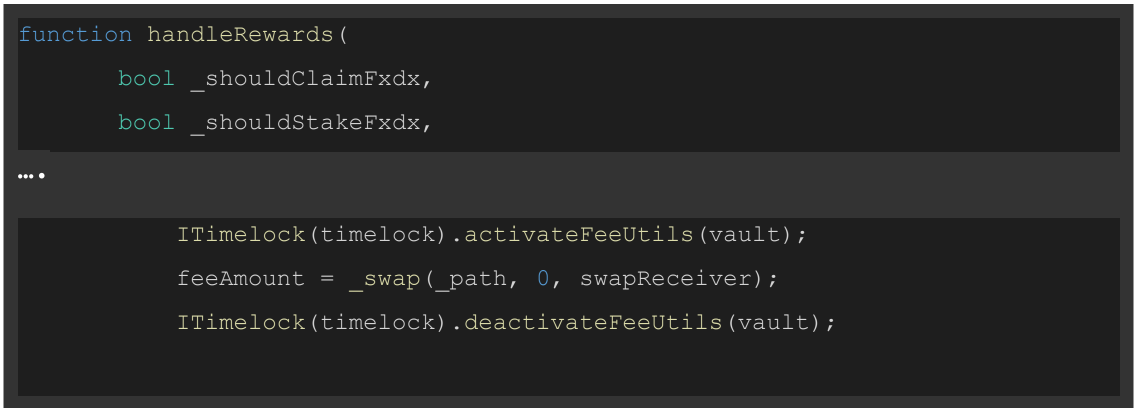
Status

Resolved

Risk Level

Severity: Medium, Likelihood: Medium

Code Segment



```
function handleRewards(
        bool _shouldClaimFxdx,
        bool _shouldStakeFxdx,
….
             ITimelock(timelock).activateFeeUtils(vault);
             feeAmount = _swap(_path, 0, swapReceiver);
             ITimelock(timelock).deactivateFeeUtils(vault);
```

Description

The _*swap* is called with a '_minOut' value of zero, it can potentially be abused by MEV searchers.

Code Location

```
contracts/staking/RewardRouterV2.sol
```

Proof of Concept

–

Recommendation

Estimate the _minOut before doing swap.

(FD-2) `MAX_PRICE_DURATION` set to 24 hours.

Status

Pending

Risk Level

Severity: Medium, Likelihood: Medium

Code Segment

```
uint256 public constant MAX_PRICE_DURATION = 24 hours;
```

Description

The price can be updated at most once every 24 hours. This would result in a slower update rate for the price, which could lead to the price used in the contract becoming outdated and inaccurate for longer periods of time. This could potentially lead to issues such as incorrect liquidations or inaccurate trading decisions.

Code Location

```
contracts/oracle/FastPriceFeed.sol
```

Proof of Concept

–

Recommendation

Set the time in a manner similar to GMX, for example setting it to 30 minutes.

(FD-3) `isEthOut` can be removed for saving gas.

Status

Resolved

Risk Level

Severity: Low

Code Segment

```
function executeRemoveLiquidity(bytes32 _key, address payable
_executionFeeReceiver) public nonReentrant returns (bool) {
............
       uint256 amountOut =
IRewardRouter(rewardRouter).unstakeAndRedeemFlpForAccount(
           request.account,
           request.tokenOut,
           request.flpAmount,
           request.minOut,
           address(this)
       );
       ITimelock(timelock).deactivateFeeUtils(vault);

       if (request.isETHOut) {
          _transferOutETHWithGasLimit(amountOut, payable(request.receiver));
       } else {
          IERC20(request.tokenOut).safeTransfer(request.receiver, amountOut);
       }
```

Description

The use of `isETHOut` can be replaced by `request.account == weth.`

Code Location

```
contracts/core/LiquidityRouter.sol
```

Proof of Concept

-

Recommendation

Use `request.account == weth` instead of `isETHOut`.

(FD-4) Renaming from funding-rate to rollover rate.

Status

Pending

Risk Level

Severity: Low

Code Segment

-

Description

The FeeUtilsV1 and FeeUtilsV2 were refactored from Vault/VaultUtils and the variable and function names were changed from 'funding-rate' to 'rollover rate,' but no changes were made to the logic or formula used to calculate the rollover rate.

Code Location

```
contracts/core/FeeUtilsV1.sol
contracts/core/FeeUtilsV2.sol
```

Proof of Concept

–

Recommendation

Change the method for calculating the rollover rate.

(FD-5) Emitting more events.

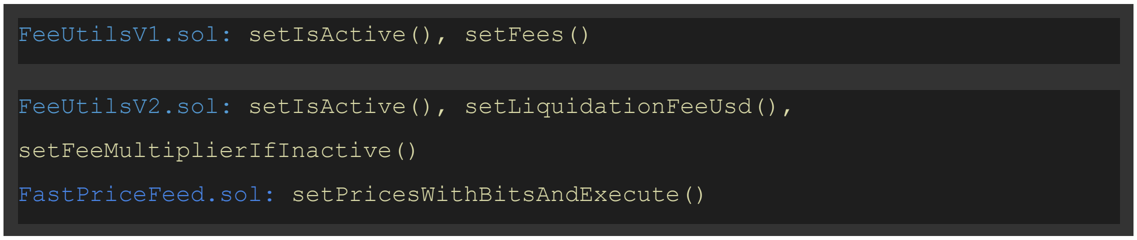
Status

Pending

Risk Level

Severity: Low

Code Segment



```
FeeUtilsV1.sol: setIsActive(), setFees()

FeeUtilsV2.sol: setIsActive(), setLiquidationFeeUsd(),
setFeeMultiplierIfInactive()
FastPriceFeed.sol: setPricesWithBitsAndExecute()
```

Description

To validate the proper deployment and initialization of the contracts, it’s a good practice to emit events, also any important state transaction can be logged, which is beneficial for monitoring the contract and tracking eventual bugs.

Code Location

```
contracts/core/FeeUtilsV1.sol
contracts/core/FeeUtilsV2.sol
contracts/oracle/setPricesWithBitsAndExecute.sol
```

Proof of Concept

–

Recommendation

Emit event for listed functions.

(FD-6) _onlyGov should be a modifier to improve the code clarity.

Status

Pending

Risk Level

Severity: Low

Code Segment

```
_onlyGov();
```

Description

The function _onlyGov() is duplicated multiple times within each contract, resulting in code duplication even there is onlyGov written in Governable.sol

Code Location

```
contracts/core/FeeUtilsV1.sol
contracts/core/FeeUtilsV2.sol
contracts/core/Vault.sol
```

Proof of Concept

–

Recommendation

Should use "access/Governable.sol" modifier instead.

(FD-7) `action` parameter is redundant.

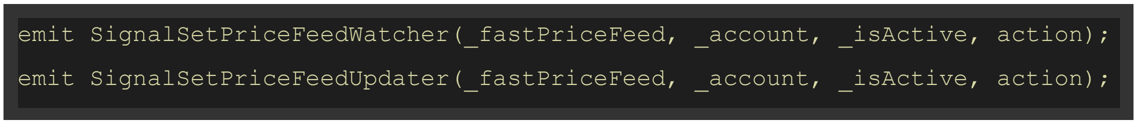
Status

Pending

Risk Level

Severity: Low

Code Segment



```
emit SignalSetPriceFeedWatcher(_fastPriceFeed, _account, _isActive, action);
emit SignalSetPriceFeedUpdater(_fastPriceFeed, _account, _isActive, action);
```

Description

`action` parameter is the hash of `fastPriceFeed, _account, _isActive`, the events are distinguished by name so this is redundant.

Code Location

```
contracts/peripherals/PriceFeedTimelock.sol
```

Proof of Concept

–

Recommendation

Remove the action parameter, saving gas.

(FD-8) `initialize` declared but all other functions don't check `isInitialized`.

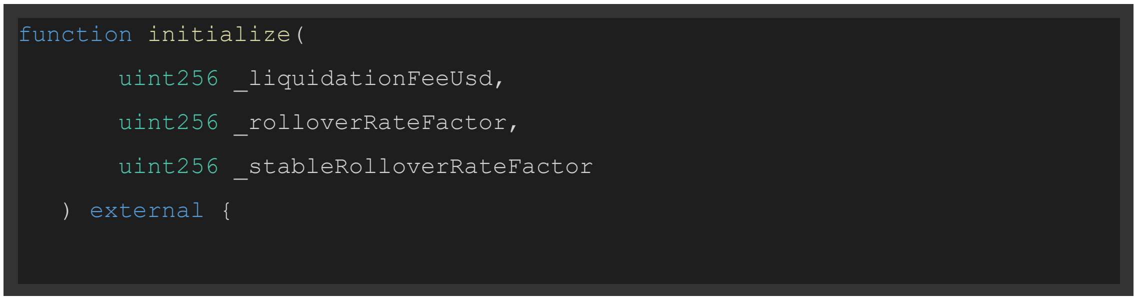
Status

Resolved

Risk Level

Severity: Low

Code Segment



```
function initialize(
       uint256 _liquidationFeeUsd,
       uint256 _rolloverRateFactor,
       uint256 _stableRolloverRateFactor
   ) external {
```

Description

The code used the `initialize` pattern but doesn't check the `isInitialized` variable when trying to access the get/set function

Code Location

```
contracts/core/FeeUtilsV1.sol
contracts/core/FeeUtilsV2.sol
```

Proof of Concept

–

Recommendation

Add `isInitialized` modifier for all get/set functions.

(FD-9) Duplication code between `initialize` and `setFees`

Status

Resolved

Risk Level

Severity: Informational

Code Segment

Description

Duplication code between `initialize` and `setFees`.

Code Location

```
contracts/core/FeeUtilsV1.sol
contracts/core/FeeUtilsV2.sol
```

Proof of Concept

–

Recommendation

Should use `setFees` inside `initialize`.

(FD-10) `getIncreasePositionFee` and `getDecreasePositionFee` have some parameters that are unused.

Risk Level

Severity: Informational

Code Segment

Description

There are some parameters that have no use since the developer refactored the code.

Code Location

```
contracts/core/FeeUtilsV1.sol
```

```
contracts/core/FeeUtilsV2.sol
```

Proof of Concept

–

Recommendation

Consider removing unused parameters to make the code easier to understand.

(FD-11) Unable to Remove a Token From Whitelist.

Status

Resolved

Risk Level

Severity: Low

Code Segment

Description

In the current implementation of `Vault.sol` and `Timelock.sol`, it is not possible to remove a token from `whitelistedTokens`. Once a token is added in `Vault.sol` through `setTokeConfig(...)` function , the mapping `whitelistedTokens` for the token is set to `true`. The token can only be removed from the whitelist if `Vault.sol:clearTokenConfig(...)` function is called, which is only accessible to the `gov` address. Since the `Timelock` contract (the `gov` of the `Vault.sol`) is missing an access function to use `clearTokenConfig(...)` function, it is not possible to remove the token from the whitelist.

This is problematic because `whitelistedTokens` is used to validate a transaction for the following functions in : `Vault.sol`

1. `buyUSDF(...)` function

2. `sellUSDF(...)` function

3. `swap(...)` function

4. `increasePosition(...)` function

Without the ability to remove whitelisted tokens, the FLP will not be able to react flexibly with rapid changes on a token situation.

Code Location

```
contracts/core/Vault.sol
contracts/peripherals/Timelock.sol
```

Proof of Concept

If a token is hacked or having regulation issues, the team might want to de-list the token. However, the team will not be able to do so unless swapping the of the .Exploit Scenario: `gov` of the `Vault`.

Recommendation

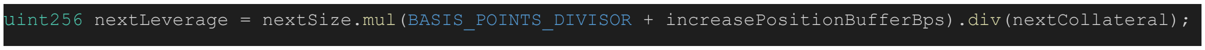
- Decouple the usage of `whiteListedToken` in `FlpManager.getAum`
- Add a function access to the function `clearTokenConfig` from the `Timelock.sol` contract.

(FD-12) `increasePositionBufferBps` Overflowable, Leading to Fees Being Always Collected.

Risk Level

Severity: Low

Code Segment



```
uint256 nextLeverage = nextSize.mul(BASIS_POINTS_DIVISOR + increasePositionBufferBps).div(nextCollateral);
```

Description

Function `BasePositionManager.setIncreasePositionBufferBps()` does not constrain the input `_increasePositionBufferBps` in either direction. In combination with the use of the unsafe addition in function `_shouldDeductFee()`. This leads to a multiplier smaller than `BASIS_POINTS_DIVISOR = 10000` and potentially always returning true for (`return nextLeverage < prevLeverage`) and lead to fees always being collected, regardless of leverage change.

Similarly in `PositionManager.sol` contract , if state `shouldValidateIncreaseOrder` variable is set to `true`, an overflowed `increasePositionBufferBps` value could lead to function `_validateIncreaseOrder()` always reverting, preventing the execution of order increasing function `executeIncreaseOrder()`. This is due to the same shared code between `BasePositionManager._shouldDeductFee()` and `PositionManager._validateIncreaseOrder()`, both of which contain the same overflow issue.

Code Location

```
contracts/core/BasePositionManager.sol
contracts/core/PositionManager.sol
```

Proof of Concept

–

Recommendation

Consider having a sane upper and lower bound for `increasePositionBufferBps`, communicate it via public-facing documentation, and accordingly check against it in function `setIncreasePositionBufferBps()`. And/or consider replacing the unsafe addition operation in `_shouldDeductFee()` with its safe counterpart (`.add()`).

(FD-13) Risk of Inaccurate FLP Token Minting

Status

Resolved

Risk Level

Severity: Low

Code Segment

Description

The `FlpManager.sol:getAum` function loops against all tokens in the `Vault.sol:allWhitelistedTokens` array. First, there is a concern of a DOS (denial of service) factor that if the governance adds enough tokens by `Vault.sol:setTokenConfig`, the loop can run out of gas. Secondly, the `Vault.sol:clearTokenConfig` function does not remove the token from the `allWhitelistedTokens` array. Thus, if the same token is cleared and set again, the token will duplicate in `allWhitelistedTokens`. The `FlpManager.sol:getAum` function will calculate the same token twice and add it to the final aum result. The problem will eventually impact the `FlpManager.sol:_addLiquidity` function and mint less FLP tokens (`aumInUsdf` will increase so that the `mintAmount` will decrease).

Code Location

```
contracts/core/Vault.sol
contracts/core/FlpManager.sol
```

Proof of Concept

–

Recommendation

Ensure the token is removed from `allWhitelistedTokens` in the `Vault.sol:clearTokenConfig` function.

(FD-14) Concerning Unexecuted Time-Locked Actions

Status

Pending

Risk Level

Severity: Informational

Description

There is no deadline for action, if action is "signaled" but never executed, it can cause unexpected risks to the users.

Code Location

```
contracts/peripherals/TimeLock.sol
contracts/peripherals/FxDxTimelock.sol
```

Recommendation

We recommend adding a deadline to the signaled actions. So once the deadline passes, the admin can no longer execute the action.

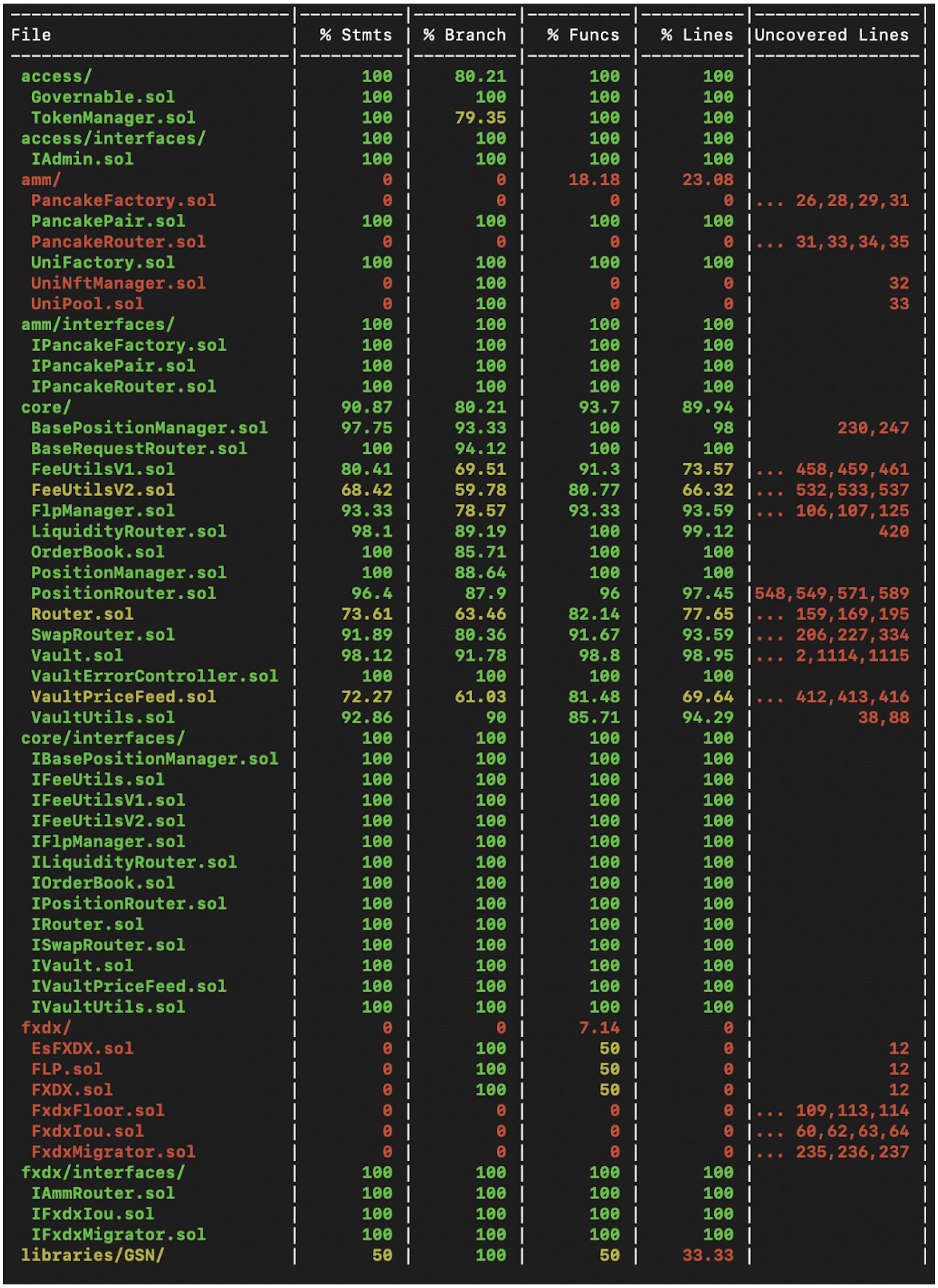
Automated Tests and Tooling

Static Analysis with Slither

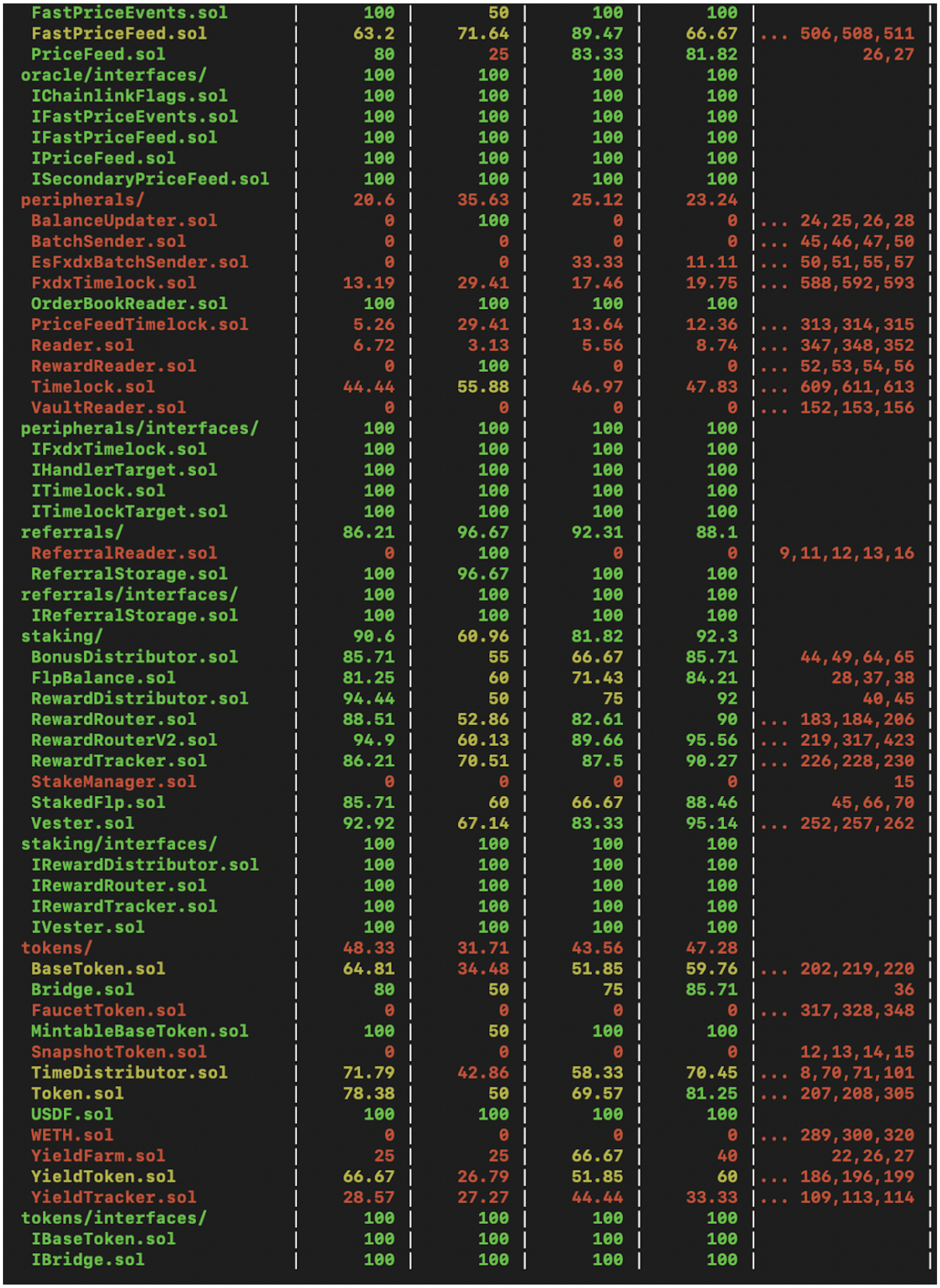
As a part of our engagement with FXDX, we ran a static analysis against the source code using Slither, which is a Solidity static analysis framework written in Python. Slither runs a suite of vulnerability detectors and prints visual information about contract details. Slither enables developers to find vulnerabilities, enhance their code comprehension, and quickly prototype custom analyses.

While Slither is not the primary element of Arcadia’s offering, in some cases, it can be useful. The following shows the results found by the static analysis by Slither. We reviewed the results, and all of the issues found by Slither were at that point in time false positives.

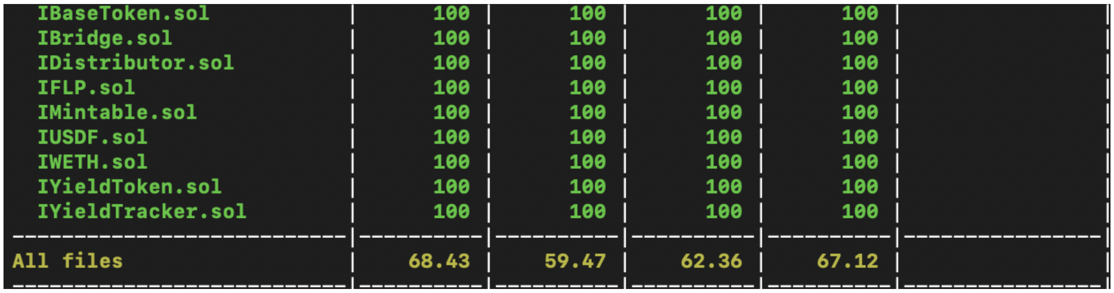
Unit Test Coverage



```
---------------------------|----------|----------|----------|----------|----------------|
File                       |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
---------------------------|----------|----------|----------|----------|----------------|
 access/                   |      100 |    80.21 |      100 |      100 |                |
  Governable.sol           |      100 |      100 |      100 |      100 |                |
  TokenManager.sol         |      100 |    79.35 |      100 |      100 |                |
 access/interfaces/        |      100 |      100 |      100 |      100 |                |
  IAdmin.sol               |      100 |      100 |      100 |      100 |                |
 amm/                      |        0 |        0 |    18.18 |    23.08 |                |
  PancakeFactory.sol       |        0 |        0 |        0 |        0 |... 26,28,29,31 |
  PancakePair.sol          |      100 |      100 |      100 |      100 |                |
  PancakeRouter.sol        |        0 |        0 |        0 |        0 |... 31,33,34,35 |
  UniFactory.sol           |      100 |      100 |      100 |      100 |                |
  UniNftManager.sol        |        0 |      100 |        0 |        0 |             32 |
  UniPool.sol              |        0 |      100 |        0 |        0 |             33 |
 amm/interfaces/           |      100 |      100 |      100 |      100 |                |
  IPancakeFactory.sol      |      100 |      100 |      100 |      100 |                |
  IPancakePair.sol         |      100 |      100 |      100 |      100 |                |
  IPancakeRouter.sol       |      100 |      100 |      100 |      100 |                |
 core/                     |    90.87 |    80.21 |     93.7 |    89.94 |                |
  BasePositionManager.sol  |    97.75 |    93.33 |      100 |       98 |        230,247 |
  BaseRequestRouter.sol    |      100 |    94.12 |      100 |      100 |                |
  FeeUtilsV1.sol           |    80.41 |    69.51 |     91.3 |    73.57 |... 458,459,461 |
  FeeUtilsV2.sol           |    68.42 |    59.78 |    80.77 |    66.32 |... 532,533,537 |
  FlpManager.sol           |    93.33 |    78.57 |    93.33 |    93.59 |... 106,107,125 |
  LiquidityRouter.sol      |     98.1 |    89.19 |      100 |    99.12 |            420 |
  OrderBook.sol            |      100 |    85.71 |      100 |      100 |                |
  PositionManager.sol      |      100 |    88.64 |      100 |      100 |                |
  PositionRouter.sol       |     96.4 |     87.9 |       96 |    97.45 |548,549,571,589 |
  Router.sol               |    73.61 |    63.46 |    82.14 |    77.65 |... 159,169,195 |
  SwapRouter.sol           |    91.89 |    80.36 |    91.67 |    93.59 |... 206,227,334 |
  Vault.sol                |    98.12 |    91.78 |     98.8 |    98.95 |... 2,1114,1115 |
  VaultErrorController.sol |      100 |      100 |      100 |      100 |                |
  VaultPriceFeed.sol       |    72.27 |    61.03 |    81.48 |    69.64 |... 412,413,416 |
  VaultUtils.sol           |    92.86 |       90 |    85.71 |    94.29 |          38,88 |
 core/interfaces/          |      100 |      100 |      100 |      100 |                |
  IBasePositionManager.sol |      100 |      100 |      100 |      100 |                |
  IFeeUtils.sol            |      100 |      100 |      100 |      100 |                |
  IFeeUtilsV1.sol          |      100 |      100 |      100 |      100 |                |
  IFeeUtilsV2.sol          |      100 |      100 |      100 |      100 |                |
  IFlpManager.sol          |      100 |      100 |      100 |      100 |                |
  ILiquidityRouter.sol     |      100 |      100 |      100 |      100 |                |
  IOrderBook.sol           |      100 |      100 |      100 |      100 |                |
  IPositionRouter.sol      |      100 |      100 |      100 |      100 |                |
  IRouter.sol              |      100 |      100 |      100 |      100 |                |
  ISwapRouter.sol          |      100 |      100 |      100 |      100 |                |
  IVault.sol               |      100 |      100 |      100 |      100 |                |
  IVaultPriceFeed.sol      |      100 |      100 |      100 |      100 |                |
  IVaultUtils.sol          |      100 |      100 |      100 |      100 |                |
 fxdx/                     |        0 |        0 |     7.14 |        0 |                |
  EsFXDX.sol               |        0 |      100 |       50 |        0 |             12 |
  FLP.sol                  |        0 |      100 |       50 |        0 |             12 |
  FXDX.sol                 |        0 |      100 |       50 |        0 |             12 |
  FxdxFloor.sol            |        0 |        0 |        0 |        0 |... 109,113,114 |
  FxdxIou.sol              |        0 |        0 |        0 |        0 |... 60,62,63,64 |
  FxdxMigrator.sol         |        0 |        0 |        0 |        0 |... 235,236,237 |
 fxdx/interfaces/          |      100 |      100 |      100 |      100 |                |
  IAmmRouter.sol           |      100 |      100 |      100 |      100 |                |
  IFxdxIou.sol             |      100 |      100 |      100 |      100 |                |
  IFxdxMigrator.sol        |      100 |      100 |      100 |      100 |                |
 libraries/GSN/            |       50 |      100 |       50 |    33.33 |                |
```



```
  FastPriceEvents.sol      |      100 |       50 |      100 |      100 |                |
  FastPriceFeed.sol        |     63.2 |    71.64 |    89.47 |    66.67 |... 506,508,511 |
  PriceFeed.sol            |       80 |       25 |    83.33 |    81.82 |          26,27 |
 oracle/interfaces/        |      100 |      100 |      100 |      100 |                |
  IChainlinkFlags.sol      |      100 |      100 |      100 |      100 |                |
  IFastPriceEvents.sol     |      100 |      100 |      100 |      100 |                |
  IFastPriceFeed.sol       |      100 |      100 |      100 |      100 |                |
  IPriceFeed.sol           |      100 |      100 |      100 |      100 |                |
  ISecondaryPriceFeed.sol  |      100 |      100 |      100 |      100 |                |
 peripherals/              |     20.6 |    35.63 |    25.12 |    23.24 |                |
  BalanceUpdater.sol       |        0 |      100 |        0 |        0 |... 24,25,26,28 |
  BatchSender.sol          |        0 |        0 |        0 |        0 |... 45,46,47,50 |
  EsFxdxBatchSender.sol    |        0 |        0 |    33.33 |    11.11 |... 50,51,55,57 |
  FxdxTimelock.sol         |    13.19 |    29.41 |    17.46 |    19.75 |... 588,592,593 |
  OrderBookReader.sol      |      100 |      100 |      100 |      100 |                |
  PriceFeedTimelock.sol    |     5.26 |    29.41 |    13.64 |    12.36 |... 313,314,315 |
  Reader.sol               |     6.72 |     3.13 |     5.56 |     8.74 |... 347,348,352 |
  RewardReader.sol         |        0 |      100 |        0 |        0 |... 52,53,54,56 |
  Timelock.sol             |    44.44 |    55.88 |    46.97 |    47.83 |... 609,611,613 |
  VaultReader.sol          |        0 |        0 |        0 |        0 |... 152,153,156 |
 peripherals/interfaces/   |      100 |      100 |      100 |      100 |                |
  IFxdxTimelock.sol        |      100 |      100 |      100 |      100 |                |
  IHandlerTarget.sol       |      100 |      100 |      100 |      100 |                |
  ITimelock.sol            |      100 |      100 |      100 |      100 |                |
  ITimelockTarget.sol      |      100 |      100 |      100 |      100 |                |
 referrals/                |    86.21 |    96.67 |    92.31 |     88.1 |                |
  ReferralReader.sol       |        0 |      100 |        0 |        0 |  9,11,12,13,16 |
  ReferralStorage.sol      |      100 |    96.67 |      100 |      100 |                |
 referrals/interfaces/     |      100 |      100 |      100 |      100 |                |
  IReferralStorage.sol     |      100 |      100 |      100 |      100 |                |
 staking/                  |     90.6 |    60.96 |    81.82 |     92.3 |                |
  BonusDistributor.sol     |    85.71 |       55 |    66.67 |    85.71 |    44,49,64,65 |
  FlpBalance.sol           |    81.25 |       60 |    71.43 |    84.21 |       28,37,38 |
  RewardDistributor.sol    |    94.44 |       50 |       75 |       92 |          40,45 |
  RewardRouter.sol         |    88.51 |    52.86 |    82.61 |       90 |... 183,184,206 |
  RewardRouterV2.sol       |     94.9 |    60.13 |    89.66 |    95.56 |... 219,317,423 |
  RewardTracker.sol        |    86.21 |    70.51 |     87.5 |    90.27 |... 226,228,230 |
  StakeManager.sol         |        0 |        0 |        0 |        0 |             15 |
  StakedFlp.sol            |    85.71 |       60 |    66.67 |    88.46 |       45,66,70 |
  Vester.sol               |    92.92 |    67.14 |    83.33 |    95.14 |... 252,257,262 |
 staking/interfaces/       |      100 |      100 |      100 |      100 |                |
  IRewardDistributor.sol   |      100 |      100 |      100 |      100 |                |
  IRewardRouter.sol        |      100 |      100 |      100 |      100 |                |
  IRewardTracker.sol       |      100 |      100 |      100 |      100 |                |
  IVester.sol              |      100 |      100 |      100 |      100 |                |
 tokens/                   |    48.33 |    31.71 |    43.56 |    47.28 |                |
  BaseToken.sol            |    64.81 |    34.48 |    51.85 |    59.76 |... 202,219,220 |
  Bridge.sol               |       80 |       50 |       75 |    85.71 |             36 |
  FaucetToken.sol          |        0 |        0 |        0 |        0 |... 317,328,348 |
  MintableBaseToken.sol    |      100 |       50 |      100 |      100 |                |
  SnapshotToken.sol        |        0 |        0 |        0 |        0 |    12,13,14,15 |
  TimeDistributor.sol      |    71.79 |    42.86 |    58.33 |    70.45 |... 8,70,71,101 |
  Token.sol                |    78.38 |       50 |    69.57 |    81.25 |... 207,208,305 |
  USDF.sol                 |      100 |      100 |      100 |      100 |                |
  WETH.sol                 |        0 |        0 |        0 |        0 |... 289,300,320 |
  YieldFarm.sol            |       25 |       25 |    66.67 |       40 |       22,26,27 |
  YieldToken.sol           |    66.67 |    26.79 |    51.85 |       60 |... 186,196,199 |
  YieldTracker.sol         |    28.57 |    27.27 |    44.44 |    33.33 |... 109,113,114 |
 tokens/interfaces/        |      100 |      100 |      100 |      100 |                |
  IBaseToken.sol           |      100 |      100 |      100 |      100 |                |
  IBridge.sol              |      100 |      100 |      100 |      100 |                |
```



```
   IBaseToken.sol             |      100 |      100 |      100 |      100 |                |
   IBridge.sol                |      100 |      100 |      100 |      100 |                |
   IDistributor.sol           |      100 |      100 |      100 |      100 |                |
   IFLP.sol                   |      100 |      100 |      100 |      100 |                |
   IMintable.sol              |      100 |      100 |      100 |      100 |                |
   IUSDF.sol                  |      100 |      100 |      100 |      100 |                |
   IWETH.sol                  |      100 |      100 |      100 |      100 |                |
   IYieldToken.sol            |      100 |      100 |      100 |      100 |                |
   IYieldTracker.sol          |      100 |      100 |      100 |      100 |                |
------------------------------|----------|----------|----------|----------|----------------|
All files                     |    68.43 |    59.47 |    62.36 |    67.12 |                |
------------------------------|----------|----------|----------|----------|----------------|
```

Conclusion

Arcadia identified issues that occurred at hash **#dcb52a72c3be66e9a6de23f9d3b56ddd33b302b7**.

Disclaimer

While best efforts and precautions have been taken in the preparation of this document, The Arcadia Group and the Authors assume no responsibility for errors, omissions, or damages resulting from the use of the provided information. Additionally, Arcadia would like to emphasize that the use of Arcadia's services does not guarantee the security of a smart contract or set of smart contracts and does not guarantee against attacks. One audit on its own is not enough for a project to be considered secure; that categorization can only be earned through extensive peer review and battle testing over an extended period.