Security Review Report
NM-0640 DCA.fun

NETHERMIND
SECURITY

(September 8, 2025)

Contents

1 Executive Summary

This document presents the results of a security review conducted by Nethermind Security for DCA.fun. **DCA.fun** is a decentralized infrastructure protocol designed to provide users with the tools to automate Dollar Cost Averaging (DCA) investment strategies on-chain. The project's primary goal is to offer a non-custodial and trust-minimized alternative to centralized platforms, allowing users to execute systematic investment plans while retaining full control over their assets.

The primary focus of this security review was a **differential audit** of the DCA.fun smart contract suite, examining the specific changes introduced since the previous assessment, **NM-0563**. The scope was strictly limited to the code changes between the final commit of the previous audit (7abd623) and the new codebase state at commit 72db329 for this audit, **NM-0640**. The key modifications involved refactoring the contract setup process by moving initial configuration logic from **constructors** into **initializer functions** and setters. This architectural shift is intended to facilitate a simpler and more flexible deployment process, particularly for deployments to **deterministic addresses**.

The audit comprises 1352 lines of Solidity code. **The audit was performed using** (a) manual analysis of the codebase, and (b) automated analysis tools.

Along this document, we report a single point of attention, classified as `Informational` severity. The issues are summarized in Fig. 1.

This document is organized as follows. Section 2 presents the files in the scope. Section 3 summarizes the issues. Section 4 discusses the risk rating methodology. Section 5 details the issues. Section 6 discusses the documentation provided by the client for this audit. Section 7 presents the test suite evaluation and automated tools used. Section 8 concludes the document.

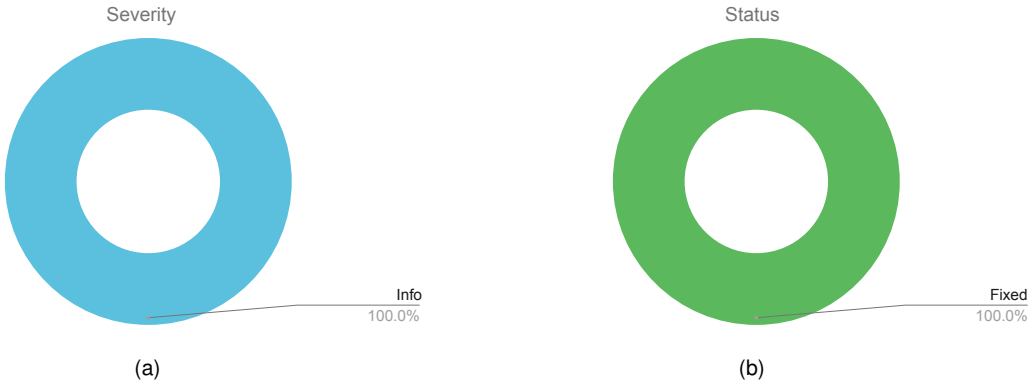

Fig. 1: Distribution of issues: Critical (0), **High** (0), **Medium** (0), **Low** (0), **Undetermined** (0), **Informational** (1), **Best Practices** (0). **Distribution of status: Fixed** (1), **Acknowledged** (0), **Mitigated** (0), **Unresolved** (0)

Summary of the Audit

Audit Type	Security Review
Final Report	September 8, 2025
Initial Commit	72db329418bbf72d5981fba82f16a13693391df1
Final Commit	3193689e8c94ce545ec2b30eb3558d0db36eb3e3
Documentation Assessment	High
Test Suite Assessment	High

2 Audited Files

	Contract	LoC	Comments	Ratio	Blank	Total
1	src/dcaDotFun/DcaDotFun.sol	353	79	22.4%	80	512
2	src/dcaDotFun/DcaVaultFactory.sol	64	25	39.1%	18	107
3	src/dcaDotFun/DcaVault.sol	165	48	29.1%	43	256
4	src/dcaDotFun/interfaces/IFillOrderCallback.sol	4	6	150.0%	1	11
5	src/dcaDotFun/interfaces/IDcaVault.sol	61	95	155.7%	26	182
6	src/dcaDotFun/interfaces/IDcaDotFun.sol	132	148	112.1%	28	308
7	src/dcaDotFun/interfaces/IDcaVaultFactory.sol	25	37	148.0%	11	73
8	src/verifierDotFun/VerifierDotFun.sol	78	38	48.7%	31	147
9	src/verifierDotFun/libraries/VerifierLib.sol	46	11	23.9%	10	67
10	src/verifierDotFun/interfaces/IVerifierDotFun.sol	56	59	105.4%	16	131
11	src/verifierDotFun/interfaces/IVerifierProxy.sol	14	15	107.1%	4	33
12	src/verifierDotFun/interfaces/IFeeManager.sol	10	11	110.0%	5	26
13	src/dotFun/DotFun.sol	229	105	45.9%	67	401
14	src/dotFun/libraries/AaveConstants.sol	25	2	8.0%	12	39
15	src/dotFun/interfaces/IDotFun.sol	83	103	124.1%	26	212
16	src/interfaces/IWETH.sol	7	15	214.3%	4	26
	Total	**1352**	**797**	**58.9%**	**382**	**2531**

*The scope of the audit was strictly limited to the code changes between the final commit of the previous audit (7abd623) and the new codebase state at commit 72db329 for this audit, NM-0640.

3 Summary of Issues

	Finding	Severity	Update
1	Key configuration variables should be set atomically upon deployment	Info	Fixed

4 Risk Rating Methodology

The risk rating methodology used by Nethermind Security follows the principles established by the OWASP Foundation. The severity of each finding is determined by two factors: **Likelihood** and **Impact**.

Likelihood measures how likely the finding is to be uncovered and exploited by an attacker. This factor will be one of the following values:

a) **High**: The issue is trivial to exploit and has no specific conditions that need to be met;

b) **Medium**: The issue is moderately complex and may have some conditions that need to be met;

c) **Low**: The issue is very complex and requires very specific conditions to be met.

When defining the likelihood of a finding, other factors are also considered. These can include but are not limited to motive, opportunity, exploit accessibility, ease of discovery, and ease of exploit.

Impact is a measure of the damage that may be caused if an attacker exploits the finding. This factor will be one of the following values:

a) **High**: The issue can cause significant damage, such as loss of funds or the protocol entering an unrecoverable state;

b) **Medium**: The issue can cause moderate damage, such as impacts that only affect a small group of users or only a particular part of the protocol;

c) **Low**: The issue can cause little to no damage, such as bugs that are easily recoverable or cause unexpected interactions that cause minor inconveniences.

When defining the impact of a finding, other factors are also considered. These can include but are not limited to Data/state integrity, loss of availability, financial loss, and reputation damage. After defining the likelihood and impact of an issue, the severity can be determined according to the table below.

		Severity Risk		
Impact	**High**	Medium	High	Critical
	Medium	Low	Medium	High
	Low	Info/Best Practices	Low	Medium
	Undetermined	Undetermined	Undetermined	Undetermined
		Low	**Medium**	**High**
		Likelihood		

To address issues that do not fit a High/Medium/Low severity, Nethermind Security also uses three more finding severities: **Informational**, **Best Practices**, and **Undetermined**.

a) **Informational** findings do not pose any risk to the application, but they carry some information that the audit team intends to pass to the client formally;

b) **Best Practice** findings are used when some piece of code does not conform with smart contract development best practices;

c) **Undetermined** findings are used when we cannot predict the impact or likelihood of the issue.

5 Issues

5.1 [Info] Key configuration variables should be set atomically upon deployment

File(s): src/dcaDotFun/DcaDotFun.sol

Description: Across multiple contracts in the scope of the audit, constructors have been simplified. Previously, they were responsible for initializing a comprehensive set of configuration parameters, ensuring that each contract was deployed in a fully configured and operational state.

For example, the constructor in the `DcaDotFun` contract used to set critical addresses, slippage parameters, and fee splits. The new implementation now only handles minimal setup, such as setting the `PERMIT2` address and granting administrative roles.

```
constructor(address owner_, address permit2_) {
    _PERMIT2 = permit2_;
    _grantRole(DEFAULT_ADMIN_ROLE, owner_);
    _grantRole(PAUSER_ROLE, owner_);
}
```

This pattern of moving initialization logic from the constructor to individual setter functions is present in other contracts as well. It shifts the responsibility of configuration to the deployer, who must now execute a series of separate transactions after deployment. This creates a time window where contracts are live on-chain but not fully configured. If a user interacts with a contract during this period, it may operate with default zero-values for critical parameters, leading to unintended behavior.

Recommendation(s): Consider the risks associated with the multi-transaction initialization process. This approach is susceptible to human error and can leave contracts in an inconsistent state if the deployment and configuration script is interrupted or improperly executed. It is advised to carefully evaluate this strategy and ensure that robust, atomic deployment procedures are in place to prevent any period where contracts are live but misconfigured.

Status: Fixed

Update from the client: We acknowledge this finding and have implemented a safeguard by setting `isCreateOrderPaused` to true in the constructor, preventing any user interactions until the contract is fully configured. This approach allows us to maintain our gas optimization strategy (1 million optimizer runs) while ensuring contracts cannot be used in an incomplete state. Once all configuration parameters are properly set and verified, we will unpause the contract to enable normal operations.

6 Documentation Evaluation

Software documentation refers to the written or visual information that describes the functionality, architecture, design, and implementation of software. It provides a comprehensive overview of the software system and helps users, developers, and stakeholders understand how the software works, how to use it, and how to maintain it. Software documentation can take different forms, such as user manuals, system manuals, technical specifications, requirements documents, design documents, and code comments. Software documentation is critical in software development, enabling effective communication between developers, testers, users, and other stakeholders. It helps to ensure that everyone involved in the development process has a shared understanding of the software system and its functionality. Moreover, software documentation can improve software maintenance by providing a clear and complete understanding of the software system, making it easier for developers to maintain, modify, and update the software over time. Smart contracts can use various types of software documentation. Some of the most common types include:

- Technical whitepaper: A technical whitepaper is a comprehensive document describing the smart contract's design and technical details. It includes information about the purpose of the contract, its architecture, its components, and how they interact with each other;
- User manual: A user manual is a document that provides information about how to use the smart contract. It includes step-by-step instructions on how to perform various tasks and explains the different features and functionalities of the contract;
- Code documentation: Code documentation is a document that provides details about the code of the smart contract. It includes information about the functions, variables, and classes used in the code, as well as explanations of how they work;
- API documentation: API documentation is a document that provides information about the API (Application Programming Interface) of the smart contract. It includes details about the methods, parameters, and responses that can be used to interact with the contract;
- Testing documentation: Testing documentation is a document that provides information about how the smart contract was tested. It includes details about the test cases that were used, the results of the tests, and any issues that were identified during testing;
- Audit documentation: Audit documentation includes reports, notes, and other materials related to the security audit of the smart contract. This type of documentation is critical in ensuring that the smart contract is secure and free from vulnerabilities.

These types of documentation are essential for smart contract development and maintenance. They help ensure that the contract is properly designed, implemented, and tested, and they provide a reference for developers who need to modify or maintain the contract in the future.

Remarks about DCA.fun's documentation

The **DCA.fun** team provided comprehensive documentation for the protocol. The codebase is well-documented with detailed NatSpec comments, and the project's `README.md` file offers a clear overview of the system's architecture, core components, and user flows. Furthermore, the team was highly responsive and available for internal sync calls, promptly addressing all questions and discussion points raised by the Nethermind Security team. This collaborative approach provided valuable insights and ensured a thorough understanding of the protocol's technical aspects throughout the engagement.

7 Test Suite Evaluation

7.1 Tests Output

```
$ npm run testAll
> dca-v2@1.0.0 testAll
> clear && forge clean && npx ts-node test/helpers/src/chainlink/index.ts bulk
 ↪  0x0003dc85e8b01946bf9dfd8b0db860129181eb6105a8c8981d9f28e00b6f60d9,0x000359843a543ee2fe414dc14c7e7920ef10f4372990b
79d6361cdc0dd1ba782,0x00036fe43f87884450b4c7e093cd5ed99cac6640d8c2000e6afc02c8838d0265 && forge test

 Cleared existing data.json
 Data written to data.json
Compiling 14 files with Solc 0.8.19
Compiling 133 files with Solc 0.8.25
Solc 0.8.19 finished in 361.80ms
Solc 0.8.25 finished in 48.93s
Compiler run successful!

Ran 1 test for test/protocol/dcaVaultFactory/dcaVaultFactory_eventEmission.t.sol:DcaVaultFactoryEventEmission
[PASS] test_dcaVaultFactory_createVault_event_emission() (gas: 357260)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 587.54µs (80.38µs CPU time)

Ran 5 tests for test/protocol/dcaVault/dcaVault_accessControl.t.sol:DcaVaultAccessControl
[PASS] test_dcaVault_cancelOrder_access_control() (gas: 21793)
[PASS] test_dcaVault_fillOrder_access_control() (gas: 18106)
[PASS] test_dcaVault_updateEscrow_access_control() (gas: 15911)
[PASS] test_dcaVault_withdrawErc20_access_control() (gas: 18270)
[PASS] test_dcaVault_withdrawNative_access_control() (gas: 15971)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 607.38µs (90.13µs CPU time)

Ran 3 tests for test/protocol/dcaVault/dcaVault_cancelOrder.t.sol:DcaVaultCancelOrder
[PASS] test_dcaVault_cancelOrder_already_cancelled() (gas: 421985)
[PASS] test_dcaVault_cancelOrder_no_yield() (gas: 429211)
[PASS] test_dcaVault_cancelOrder_with_escrow() (gas: 567381)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 1.10ms (495.46µs CPU time)

Ran 1 test for test/protocol/dcaVaultFactory/dcaVaultFactory_aavePoolManagement.t.sol:DcaVaultFactoryAavePoolManagement
[PASS] test_dcaVaultFactory_setAavePool_success() (gas: 728738)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 1.28ms (96.13µs CPU time)

Ran 1 test for test/protocol/dcaVault/dcaVault_initialization.t.sol:DcaVaultInitialization
[PASS] test_dcaVault_initialize_already_initialized() (gas: 377564)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 497.83µs (44.79µs CPU time)

Ran 4 tests for test/protocol/dcaVaultFactory/dcaVaultFactory_accessControl.t.sol:DcaVaultFactoryAccessControl
[PASS] test_dcaVaultFactory_createVault_access_control() (gas: 363436)
[PASS] test_dcaVaultFactory_initialize_access_control() (gas: 13264)
[PASS] test_dcaVaultFactory_ownership_transfer() (gas: 369900)
[PASS] test_dcaVaultFactory_setAavePool_access_control() (gas: 27362)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 687.92µs (220.50µs CPU time)

Ran 2 tests for
 ↪  test/protocol/dcaVaultFactory/dcaVaultFactory_constructorAndInitialization.t.sol:DcaVaultFactoryConstructorAndInit
ialization
[PASS] test_dcaVaultFactory_constructor_initialization() (gas: 21149)
[PASS] test_dcaVaultFactory_initialize_AlreadyInitialized() (gas: 15231)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 481.75µs (32.83µs CPU time)

Ran 2 tests for test/protocol/dcaVaultFactory/dcaVaultFactory_createVault.t.sol:DcaVaultFactoryEventEmission
[PASS] test_dcaVaultFactory_createVault_event_emission() (gas: 357239)
[PASS] test_dcaVaultFactory_createVault_success() (gas: 373075)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 576.54µs (132.33µs CPU time)

Ran 3 tests for
 ↪  test/protocol/dcaVaultFactory/dcaVaultFactory_deterministicAddress.t.sol:DcaVaultFactoryDeterministicAddress
[PASS] test_dcaVaultFactory_predictVaultAddress_accuracy() (gas: 351749)
[PASS] test_dcaVaultFactory_vault_address_determinism() (gas: 12035)
[PASS] test_dcaVaultFactory_vault_address_uniqueness() (gas: 12048)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 607.63µs (72.04µs CPU time)
```

```
Ran 1 test for
↪  test/protocol/dcaVaultFactory/dcaVaultFactory_edgeCasesAndFailures.t.sol:DcaVaultFactoryEdgeCasesAndFailures
[PASS] test_dcaVaultFactory_vault_double_create() (gas: 290689638)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 481.71µs (49.58µs CPU time)

Ran 1 test for test/protocol/verifierDotFun/verifierDotFun_initializer.sol:VerifierDotFunAccessControlTest
[PASS] test_verifierDotFun_initialize_AlreadyInitialized() (gas: 13216)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 9.27s (285.00µs CPU time)

Ran 2 tests for test/protocol/ValidateSetup.t.sol:CreateOrder
[PASS] test_validate_Setup() (gas: 3340)
[PASS] test_validate_token_props() (gas: 46746)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 9.28s (583.17µs CPU time)

Ran 1 test for test/protocol/verifierDotFun/verifierDotFun_setFunContract.t.sol:VerifierDotFunSetFunContractTest
[PASS] test_verifierDotFun_setFunContract_onlyOwner() (gas: 35271)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 9.42s (157.42ms CPU time)

Ran 1 test for
↪  test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder_notStaked_notStaked.t.sol:CreateOrderNotStakedTokenInNot
StakedTokenOut
[PASS] test_dcaDotFun_createOrder_notStaked_tokenIn_notStaked_tokenOut() (gas: 906305)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 11.68s (2.40s CPU time)

Ran 3 tests for test/protocol/dcaVault/dcaVault_eventEmission.t.sol:DcaVaultEventEmission
[PASS] test_dcaVault_cancelOrder_event() (gas: 852539)
[PASS] test_dcaVault_withdrawErc20_event() (gas: 980640)
[PASS] test_dcaVault_withdrawNative_event() (gas: 869916)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 11.82s (2.56s CPU time)

Ran 5 tests for test/protocol/dcaDotFun/dcaDotFun_accessControl.t.sol:DcaDotFunAccessControl
[PASS] test_dcaDotFun_access_control_events() (gas: 138880)
[PASS] test_dcaDotFun_admin_functions_onlyOwner() (gas: 64002)
[PASS] test_dcaDotFun_ownership_transfer() (gas: 102699)
[PASS] test_dcaDotFun_pause_blocks_operations() (gas: 1354704)
[PASS] test_dcaDotFun_role_based_access() (gas: 128735)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 11.92s (2.64s CPU time)

Ran 4 tests for test/protocol/dcaDotFun/cancelOrder/dcaDotFun_cancelOrder.t.sol:CancelOrder
[PASS] test_dcaDotFun_cancelOrders_NotOrderCreator() (gas: 863843)
[PASS] test_dcaDotFun_cancelOrders_OrderCancelled() (gas: 855521)
[PASS] test_dcaDotFun_cancelOrders_repeats_set_to_0() (gas: 855190)
[PASS] test_dcaDotFun_cancelOrders_with_fill_escrow_set_to_0() (gas: 1266710)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 11.92s (2.64s CPU time)

Ran 14 tests for test/protocol/dcaDotFun/fillOrder/dcaDotFun_quote_revert.t.sol:QuoteRevert
[PASS] test_dcaDotFun_quote_FeedIdMismatch() (gas: 1066203)
[PASS] test_dcaDotFun_quote_InvalidReportLength() (gas: 1244936)
[PASS] test_dcaDotFun_quote_InvalidTokenOutAmount() (gas: 994045)
[PASS] test_dcaDotFun_quote_OrderDoesNotExist() (gas: 149116)
[PASS] test_dcaDotFun_quote_OrderNotFillable() (gas: 973231)
[PASS] test_dcaDotFun_quote_PriceIsZero() (gas: 1026083)
[PASS] test_dcaDotFun_quote_PriorToOrderExecution() (gas: 971990)
[PASS] test_dcaDotFun_quote_TimeStampMismatch() (gas: 993683)
[PASS] test_dcaDotFun_quote_TokenInNotActive() (gas: 983329)
[PASS] test_dcaDotFun_quote_TokenOutNotActive() (gas: 983553)
[PASS] test_dcaDotFun_quote_cancel_order_OrderNotActive() (gas: 964013)
[PASS] test_dcaDotFun_quote_no_repeats_remaining_OrderNotActive() (gas: 1278343)
[PASS] test_dcaDotFun_quote_tokenIn_ExpiredReport() (gas: 1274922)
[PASS] test_dcaDotFun_quote_tokenOut_ExpiredReport() (gas: 1279678)
Suite result: ok. 14 passed; 0 failed; 0 skipped; finished in 11.93s (2.66s CPU time)

Ran 1 test for test/protocol/dcaVault/dcaVault_withdrawNative.t.sol:DcaVaultWithdrawNative
[PASS] test_dcaVault_withdrawNative_success() (gas: 400573)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 1.29ms (147.67µs CPU time)

Ran 4 tests for
↪  test/protocol/dcaDotFun/cancelOrder/dcaDotFun_cancelOrder_staked_notStaked.t.sol:CancelOrderStakedTokenInNotStaked
TokenOut
[PASS] test_dcaDotFun_cancelOrder_staked_tokenIn_notStaked_tokenOut_no_fills_no_yield() (gas: 1215436)
[PASS] test_dcaDotFun_cancelOrder_staked_tokenIn_notStaked_tokenOut_no_fills_with_yield() (gas: 1315300)
[PASS] test_dcaDotFun_cancelOrder_staked_tokenIn_notStaked_tokenOut_with_fill_no_yield() (gas: 186)
```

```
[PASS] test_dcaDotFun_cancelOrder_staked_tokenIn_notStaked_tokenOut_with_fill_with_yield() (gas: 1847083)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 15.30s (6.03s CPU time)

Ran 1 test for test/protocol/dcaDotFun/flashRouter/dcaDotFun_flashRouter_fillOrder.t.sol:FlashRouterFillOrder
[PASS] test_dcaDotFun_flashRouter_fillOrder() (gas: 3605908)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 16.37s (7.09s CPU time)
Ran 3 tests for test/protocol/dcaDotFun/dcaDotFun_constructor.t.sol:DcaDotFunConstructor
[PASS] test_dcaDotFun_invalidYieldSplit() (gas: 5112888)
[PASS] test_dcaDotFun_max_gt_10000_InvalidSlippage() (gas: 5113062)
[PASS] test_dcaDotFun_min_gt_max_InvalidSlippage() (gas: 5112972)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 1.10ms (742.42µs CPU time)

Ran 21 tests for test/protocol/dcaVault/dcaVault_withdrawErc20.t.sol:DcaVaultWithdrawErc20
[PASS] test_dcaVault_withdrawErc20_aTokenIn_cancelled_add_post_cancel() (gas: 1331605)
[PASS] test_dcaVault_withdrawErc20_aTokenIn_cancelled_add_prior_to_cancel() (gas: 1357369)
[PASS] test_dcaVault_withdrawErc20_aTokenIn_not_cancelled() (gas: 1079605)
[PASS] test_dcaVault_withdrawErc20_aTokenOut() (gas: 1593198)
[PASS] test_dcaVault_withdrawErc20_aTokenOut_cancel_withdrawErc20_to_bypass_yield_split() (gas: 1705625)
[PASS] test_dcaVault_withdrawErc20_aTokenOut_cancelled_add_post_cancel() (gas: 1227190)
[PASS] test_dcaVault_withdrawErc20_aTokenOut_not_cancelled_balance_eq_escrow() (gas: 1593161)
[PASS] test_dcaVault_withdrawErc20_aTokenOut_not_cancelled_balance_gt_escrow() (gas: 1708589)
[PASS] test_dcaVault_withdrawErc20_cancelled() (gas: 922099)
[PASS] test_dcaVault_withdrawErc20_non_order_token() (gas: 1695433)
[PASS] test_dcaVault_withdrawErc20_not_cancelled() (gas: 930049)
[PASS] test_dcaVault_withdrawErc20_tokenIn_cancelled_add_post_cancel() (gas: 881156)
[PASS] test_dcaVault_withdrawErc20_tokenIn_cancelled_add_prior_to_cancel() (gas: 872804)
[PASS] test_dcaVault_withdrawErc20_tokenIn_not_cancelled() (gas: 864944)
[PASS] test_dcaVault_withdrawErc20_tokenOut_cancelled_add_post_cancel() (gas: 1286810)
[PASS] test_dcaVault_withdrawErc20_tokenOut_cancelled_add_prior_to_cancel() (gas: 1273538)
[PASS] test_dcaVault_withdrawErc20_tokenOut_not_cancelled_balance_eq_escrow() (gas: 1296131)
[PASS] test_dcaVault_withdrawErc20_tokenOut_not_cancelled_balance_gt_escrow() (gas: 1425748)
[PASS] test_dcaVault_withdrawErc20_tokenOut_with_escrow() (gas: 1445937)
[PASS] test_dcaVault_withdrawErc20_zero_address() (gas: 864564)
[PASS] test_dcaVault_withdrawErc20_zero_balance() (gas: 1638471)
Suite result: ok. 21 passed; 0 failed; 0 skipped; finished in 17.26s (7.99s CPU time)

Ran 1 test for test/protocol/verifierDotFun/verifierDotFun_withdrawToken.t.sol:VerifierDotFunWithdrawTokenTest
[PASS] test_verifierDotFun_withdrawToken_onlyOwner() (gas: 203203)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 9.61s (238.10ms CPU time)

Ran 5 tests for test/protocol/verifierDotFun/verifierDotFun_accessControl.t.sol:VerifierDotFunAccessControlTest
[PASS] test_verifierDotFun_setFunContract_NotOwner() (gas: 15656)
[PASS] test_verifierDotFun_setManagersFeeTokenAndApprove_NotOwner() (gas: 13304)
[PASS] test_verifierDotFun_transferOwnership_NotOwner() (gas: 23084)
[PASS] test_verifierDotFun_verifyReportBulk_NotFun() (gas: 15772)
[PASS] test_verifierDotFun_withdrawToken_NotOwner() (gas: 13702)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 8.14s (115.35ms CPU time)

Ran 1 test for
↪  test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder_notStaked_staked.t.sol:CreateOrderNotStakedTokenInStaked
TokenOut
[PASS] test_dcaDotFun_createOrder_notStaked_tokenIn_staked_tokenOut() (gas: 955673)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 11.47s (2.42s CPU time)

Ran 4 tests for test/protocol/dcaDotFun/fillOrder/dcaDotFun_scalingSlippage.t.sol:ScalingSlippage
[PASS] test_dcaDotFun_scaling_slippage_at_half_of_scalingFactor() (gas: 1071705)
[PASS] test_dcaDotFun_scaling_slippage_freqInterval_max() (gas: 1071616)
[PASS] test_dcaDotFun_scaling_slippage_freqInterval_min() (gas: 1071182)
[PASS] test_dcaDotFun_scaling_slippage_gt_half_of_scalingFactor() (gas: 1071730)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 11.24s (2.49s CPU time)

Ran 26 tests for test/protocol/dcaDotFun/dcaDotFun_tokenManagement.t.sol:DcaDotFunTokenManagement
[PASS] test_dcaDotFun_getTokenProps() (gas: 58465)
[PASS] test_dcaDotFun_pauseCreateOrder() (gas: 37251)
[PASS] test_dcaDotFun_pauseFillOrder() (gas: 37232)
[PASS] test_dcaDotFun_setAavePool1() (gas: 28354)
[PASS] test_dcaDotFun_setFeeCollector() (gas: 29252)
[PASS] test_dcaDotFun_setMaxFeedAge_for_createOrder() (gas: 46146)
[PASS] test_dcaDotFun_setMaxFeedAge_for_fillOrder() (gas: 46123)
[PASS] test_dcaDotFun_setMaxScalingInterval() (gas: 46709)
[PASS] test_dcaDotFun_setMinExecutionValue() (gas: 287563)
```

```
[PASS] test_dcaDotFun_setMinMaxSlippage() (gas: 50996)
[PASS] test_dcaDotFun_setMinMaxSlippage_max_gt_1000InvalidSlippage() (gas: 13529)
[PASS] test_dcaDotFun_setMinMaxSlippage_min_gt_max_InvalidSlippage() (gas: 13496)
[PASS] test_dcaDotFun_setMinOrderFrequencyInterval() (gas: 46839)
[PASS] test_dcaDotFun_setNativeToken() (gas: 58228)
[PASS] test_dcaDotFun_setProtocolFee() (gas: 35594)
[PASS] test_dcaDotFun_setTimestampTolerance() (gas: 32126)
[PASS] test_dcaDotFun_setTokenProps_1() (gas: 219486)
[PASS] test_dcaDotFun_setTokenProps_AccessControlUnauthorizedAccount() (gas: 22543)
[PASS] test_dcaDotFun_setTokenProps_InvalidAaveAsset() (gas: 65532)
[PASS] test_dcaDotFun_setTokenProps_ZeroFeed() (gas: 21659)
[PASS] test_dcaDotFun_setTokenState() (gas: 70096)
[PASS] test_dcaDotFun_setTokenState_AccessControlUnauthorizedAccount() (gas: 20061)
[PASS] test_dcaDotFun_setVaultFactory() (gas: 46898)
[PASS] test_dcaDotFun_setVerifierDotFun() (gas: 31211)
[PASS] test_dcaDotFun_setYieldSplit() (gas: 46767)
[PASS] test_dcaDotFun_setYieldSplit_InvalidYieldSplit() (gas: 13421)
Suite result: ok. 26 passed; 0 failed; 0 skipped; finished in 11.67s (3.65s CPU time)

Ran 6 tests for test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder.t.sol:CreateOrderMissingTests
[PASS] test_dcaDotFun_createOrder_IncrementOrderId() (gas: 858212)
[PASS] test_dcaDotFun_createOrder_firstExecution_nonZero() (gas: 883219)
[PASS] test_dcaDotFun_createOrder_multipleOrders() (gas: 1522789)
[PASS] test_dcaDotFun_createOrder_protocolFee() (gas: 874446)
[PASS] test_dcaDotFun_createOrder_recipient() (gas: 870163)
[PASS] test_dcaDotFun_createOrder_yieldSplit() (gas: 1092594)
Suite result: ok. 6 passed; 0 failed; 0 skipped; finished in 15.48s (5.95s CPU time)

Ran 1 test for
↪  test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder_staked_staked.t.sol:CreateOrderStakedTokenInStakedToken
Out
[PASS] test_dcaDotFun_createOrder_staked_tokenIn_staked_tokenOut() (gas: 1174455)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 13.10s (4.92s CPU time)

Ran 1 test for test/protocol/dcaDotFun/fillOrder/dcaDotFun_fillOrder_revert.t.sol:FillOrderRevert
[PASS] test_dcaDotFun_fillOrder_FillOrderPaused() (gas: 1318245)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 11.60s (3.25s CPU time)

Ran 1 test for test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder_staked_native.t.sol:CreateOrderStakedNative
[PASS] test_dcaDotFun_createOrder_staked_native() (gas: 1118893)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 14.71s (5.46s CPU time)

Ran 3 tests for test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder_native_revert.t.sol:CreateOrderNativeRevert
[PASS] test_dcaDotFun_createOrder_native_revert_InvalidEthAmount() (gas: 102241)
[PASS] test_dcaDotFun_createOrder_native_revert_NotZeroAddress() (gas: 122700)
[PASS] test_dcaDotFun_createOrder_native_revert_msgValue_0_MinExecutionValue() (gas: 233102)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 10.67s (2.14s CPU time)

Ran 4 tests for
↪  test/protocol/dcaDotFun/cancelOrder/dcaDotFun_cancelOrder_staked_staked.t.sol:CancelOrderStakedTokenInStakedToken
Out
[PASS] test_dcaDotFun_cancelOrder_staked_tokenIn_staked_tokenOut_no_fills_no_yield() (gas: 1266719)
[PASS] test_dcaDotFun_cancelOrder_staked_tokenIn_staked_tokenOut_no_fills_with_yield() (gas: 1366607)
[PASS] test_dcaDotFun_cancelOrder_staked_tokenIn_staked_tokenOut_with_fill_no_yield() (gas: 1982804)
[PASS] test_dcaDotFun_cancelOrder_staked_tokenIn_staked_tokenOut_with_fill_with_yield() (gas: 2105575)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 23.55s (11.72s CPU time)

Ran 5 tests for
↪  test/protocol/dcaDotFun/cancelOrder/dcaDotFun_cancelOrder_notStaked_staked.t.sol:CancelOrderNotStakedTokenInStaked
TokenOut
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_staked_tokenOut_cancelled_add_pre_cancel() (gas: 1324266)
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_staked_tokenOut_no_fill_with_yield() (gas: 233)
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_staked_tokenOut_no_fills_no_yield() (gas: 231)
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_staked_tokenOut_with_fill_no_yield() (gas: 1712035)
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_staked_tokenOut_with_fill_with_yield() (gas: 1758413)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 16.43s (15.05s CPU time)

Ran 21 tests for test/protocol/dcaDotFun/dcaDotFun_eventEmissions.t.sol:DcaDotFunEventEmissions
[PASS] test_dcaDotFun_cancelOrder_event() (gas: 856539)
[PASS] test_dcaDotFun_createOrderNative_event() (gas: 860981)
[PASS] test_dcaDotFun_createOrder_event() (gas: 868180)
```

```
[PASS] test_dcaDotFun_fillOrder_event() (gas: 1312483)
[PASS] test_dcaDotFun_setAavePool_event() (gas: 9437011)
[PASS] test_dcaDotFun_setMaxFeedAge_event() (gas: 25950)
[PASS] test_dcaDotFun_setMaxScalingInterval_event() (gas: 20368)
[PASS] test_dcaDotFun_setMinMaxSlippage_event() (gas: 25974)
[PASS] test_dcaDotFun_setMinOrderFrequencyInterval_event() (gas: 20389)
[PASS] test_dcaDotFun_setNativeToken_event() (gas: 28705)
[PASS] test_dcaDotFun_setVaultFactory_event() (gas: 20412)
[PASS] test_dcaDotFun_setYieldSplit_event() (gas: 17508)
[PASS] test_dotFun_pauseCreateOrder_event() (gas: 29331)
[PASS] test_dotFun_pauseFillOrder_event() (gas: 29270)
[PASS] test_dotFun_setFeeCollector_event() (gas: 20643)
[PASS] test_dotFun_setMinExecutionValue_event() (gas: 25978)
[PASS] test_dotFun_setProtocolFee_event() (gas: 20351)
[PASS] test_dotFun_setTimestampTolerance_event() (gas: 20390)
[PASS] test_dotFun_setTokenProps_event() (gas: 72348)
[PASS] test_dotFun_setTokenState_event() (gas: 19886)
[PASS] test_dotFun_setVerifierDotFun_event() (gas: 20580)
Suite result: ok. 21 passed; 0 failed; 0 skipped; finished in 10.72s (6.64s CPU time)

Ran 2 tests for
↪  test/protocol/dcaDotFun/fillOrder/dcaDotFun_fillOrder_usdc_weth_notStaked.t.sol:FillOrderUsdcWethNotStaked
[PASS] test_dcaDotFun_fillOrder_not_staked_slippage_100pct_usdc_weth() (gas: 1653696)
[PASS] test_dcaDotFun_fillOrder_not_staked_slippage_x_usdc_weth() (gas: 1315041)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 13.02s (6.53s CPU time)

Ran 1 test for
↪  test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder_staked_notStaked.t.sol:CreateOrderStakedTokenInNotStaked
TokenOut
[PASS] test_dcaDotFun_createOrder_staked_tokenIn_notStaked_tokenOut() (gas: 1129977)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 11.78s (4.40s CPU time)

Ran 1 test for
↪  test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder_notStaked_native.t.sol:CreateOrderNotStakedNative
[PASS] test_dcaDotFun_createOrder_notStaked_native() (gas: 903095)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 9.89s (2.18s CPU time)

Ran 26 tests for test/protocol/dcaDotFun/createOrder/dcaDotFun_createOrder_revert.t.sol:CreateOrderRevert
[PASS] test_dcaDotFun_OrderDoesNotExist() (gas: 14131)
[PASS] test_dcaDotFun_createOrder_CreateOrderPaused() (gas: 879368)
[PASS] test_dcaDotFun_createOrder_ExpiredReport() (gas: 278844)
[PASS] test_dcaDotFun_createOrder_FeedIdMismatch() (gas: 272379)
[PASS] test_dcaDotFun_createOrder_InsufficientAllowance() (gas: 139034)
[PASS] test_dcaDotFun_createOrder_InsufficientBalance() (gas: 170604)
[PASS] test_dcaDotFun_createOrder_InvalidFrequencyInterval() (gas: 163782)
[PASS] test_dcaDotFun_createOrder_InvalidReportLength() (gas: 227948)
[PASS] test_dcaDotFun_createOrder_InvalidSlippage_1() (gas: 159526)
[PASS] test_dcaDotFun_createOrder_InvalidSlippage_2() (gas: 161669)
[PASS] test_dcaDotFun_createOrder_MinExecutionValue() (gas: 287166)
[PASS] test_dcaDotFun_createOrder_PriceIsZero() (gas: 175428)
[PASS] test_dcaDotFun_createOrder_RecipientIsZeroAddress() (gas: 146096)
[PASS] test_dcaDotFun_createOrder_RepeatsIsZero() (gas: 123544)
[PASS] test_dcaDotFun_createOrder_TokenInAndOutSame() (gas: 150459)
[PASS] test_dcaDotFun_createOrder_TokenInNotActive() (gas: 161830)
[PASS] test_dcaDotFun_createOrder_TokenOutNotActive() (gas: 163969)
[PASS] test_dcaDotFun_createOrder_notStaked_tokenIn_staked_tokenOut_InvalidAaveAsset() (gas: 1329697)
[PASS] test_dcaDotFun_createOrder_notStaked_tokenIn_staked_tokenOut_TokenNotStakable() (gas: 170176)
[PASS] test_dcaDotFun_createOrder_scalingFactort_gt_half_of_freqInterval_InvalidScalingFactor() (gas: 161937)
[PASS] test_dcaDotFun_createOrder_scalingFactort_gt_maxScalingFactor_InvalidScalingFactor() (gas: 163918)
[PASS] test_dcaDotFun_createOrder_staked_tokenIn_TokenNotStakable() (gas: 158212)
[PASS] test_dcaDotFun_createOrder_staked_tokenIn_staked_tokenOut_DepositExceedsMax() (gas: 443753)
[PASS] test_dcaDotFun_createOrder_staked_tokenIn_staked_tokenOut_InvalidAaveAsset() (gas: 1368605)
[PASS] test_dcaDotFun_createOrder_staked_tokenIn_staked_tokenOut_TokenNotStakable() (gas: 170198)
[PASS] test_dcaDotFun_getOrderTokens() (gas: 863616)
Suite result: ok. 26 passed; 0 failed; 0 skipped; finished in 22.53s (7.22s CPU time)

Ran 4 tests for
↪  test/protocol/dcaDotFun/cancelOrder/dcaDotFun_cancelOrder_notStaked_notStaked.t.sol:CancelOrderNotStakedTokenInNot
StakedTokenOut
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_notStaked_tokenOut_no_fills_no_yield() (gas: 968532)
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_notStaked_tokenOut_no_fills_with_yield() (gas: 232)
```

```
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_notStaked_tokenOut_with_fill_no_yield() (gas: 1406597)
[PASS] test_dcaDotFun_cancelOrder_notStaked_tokenIn_notStaked_tokenOut_with_fill_with_yield() (gas: 231)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 25.91s (7.57s CPU time)

Ran 3 tests for test/protocol/dcaDotFun/fillOrder/dcaDotFun_fillOrder_usdc_weth_staked.t.sol:FillOrderUsdcWethNotStaked
[PASS] test_dcaDotFun_fillOrder_not_staked_tokenIn_staked_tokenOut() (gas: 2036650)
[PASS] test_dcaDotFun_fillOrder_staked_tokenIn_not_staked_tokenOut() (gas: 2105726)
[PASS] test_dcaDotFun_fillOrder_staked_tokenIn_staked_tokenOut() (gas: 2434892)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 13.29s (12.64s CPU time)

Ran 1 test for test/protocol/dcaDotFun/dcaDotFun_feeManagement.t.sol:DcaDotFunFeeManagement
[PASS] test_dcaDotFun_zero_fee_scenario() (gas: 1288084)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 12.39s (3.16s CPU time)
Ran 7 tests for test/protocol/dcaDotFun/dcaDotFun_orderManagement.t.sol:DcaDotFunOrderManagement
[PASS] test_dcaDotFun_createOrderNative_success() (gas: 903095)
[PASS] test_dcaDotFun_createOrder_invalid_amounts() (gas: 434635)
[PASS] test_dcaDotFun_createOrder_invalid_tokens() (gas: 206309)
[PASS] test_dcaDotFun_createOrder_success() (gas: 906259)
[PASS] test_dcaDotFun_getOrders_by_token_pair() (gas: 2285056)
[PASS] test_dcaDotFun_getOrders_by_user() (gas: 2475985)
[PASS] test_dcaDotFun_order_id_generation() (gas: 2487273)
Suite result: ok. 7 passed; 0 failed; 0 skipped; finished in 38.61s (7.58s CPU time)

Ran 44 test suites in 38.63s (452.00s CPU time): 208 tests passed, 0 failed, 0 skipped (208 total tests)
```

7.2 Automated Tools

7.2.1 AuditAgent

All the relevant issues raised by the AuditAgent have been incorporated into this report. The AuditAgent is an AI-powered smart contract auditing tool that analyses code, detects vulnerabilities, and provides actionable fixes. It accelerates the security analysis process, complementing human expertise with advanced AI models to deliver efficient and comprehensive smart contract audits. Available at https://app.auditagent.nethermind.io.

8 About Nethermind

Nethermind is a Blockchain Research and Software Engineering company. Our work touches every part of the web3 ecosystem - from layer 1 and layer 2 engineering, cryptography research, and security to application-layer protocol development. We offer strategic support to our institutional and enterprise partners across the blockchain, digital assets, and DeFi sectors, guiding them through all stages of the research and development process, from initial concepts to successful implementation.

We offer security audits of projects built on EVM-compatible chains and Starknet. We are active builders of the Starknet ecosystem, delivering a node implementation, a block explorer, a Solidity-to-Cairo transpiler, and formal verification tooling. Nethermind also provides strategic support to our institutional and enterprise partners in blockchain, digital assets, and decentralized finance (DeFi). In the next paragraphs, we introduce the company in more detail.

Blockchain Security: At Nethermind, we believe security is vital to the health and longevity of the entire Web3 ecosystem. We provide security services related to Smart Contract Audits, Formal Verification, and Real-Time Monitoring. Our Security Team comprises blockchain security experts in each field, often collaborating to produce comprehensive and robust security solutions. The team has a strong academic background, can apply state-of-the-art techniques, and is experienced in analyzing cutting-edge Solidity and Cairo smart contracts, such as ArgentX and StarkGate (the bridge connecting Ethereum and StarkNet). Most team members hold a Ph.D. degree and actively participate in the research community, accounting for 240+ articles published and 1,450+ citations in Google Scholar. The security team adopts customer-oriented and interactive processes where clients are involved in all stages of the work.

Blockchain Core Development: Our core engineering team, consisting of over 20 developers, maintains, improves, and upgrades our flagship product - the Nethermind Ethereum Execution Client. The client has been successfully operating for several years, supporting both the Ethereum Mainnet and its testnets, and now accounts for nearly a quarter of all synced Mainnet nodes. Our unwavering commitment to Ethereum's growth and stability extends to sidechains and layer 2 solutions. Notably, we were the sole execution layer client to facilitate Gnosis Chain's Merge, transitioning from Aura to Proof of Stake (PoS), and we are actively developing a full-node client to bolster Starknet's decentralization efforts. Our core team equips partners with tools for seamless node set-up, using generated docker-compose scripts tailored to their chosen execution client and preferred configurations for various network types.

DevOps and Infrastructure Management: Our infrastructure team ensures our partners' systems operate securely, reliably, and efficiently. We provide infrastructure design, deployment, monitoring, maintenance, and troubleshooting support, allowing you to focus on your core business operations. Boasting extensive expertise in Blockchain as a Service, private blockchain implementations, and node management, our infrastructure and DevOps engineers are proficient with major cloud solution providers and can host applications in-house or on clients' premises. Our global in-house SRE teams offer 24/7 monitoring and alerts for both infrastructure and application levels. We manage over 5,000 public and private validators and maintain nodes on major public blockchains such as Polygon, Gnosis, Solana, Cosmos, Near, Avalanche, Polkadot, Aptos, and StarkWare L2. Sedge is an open-source tool developed by our infrastructure experts, designed to simplify the complex process of setting up a proof-of-stake (PoS) network or chain validator. Sedge generates docker-compose scripts for the entire validator set-up based on the chosen client, making the process easier and quicker while following best practices to avoid downtime and being slashed.

Cryptography Research: At Nethermind, our cryptography Research team conducts cutting-edge internal research and collaborates closely with external partners on cryptographic protocols, consensus design, succinct arguments and folding schemes, elliptic curve-based STARK protocols, post-quantum security and zero-knowledge proofs (ZKPs). Our research has led to influential contributions, including Zinc (Crypto '25), Mova, FLI (Asiacrypt '24), and foundational results in Fiat-Shamir security and STARK proof batching. Complementing this theoretical work, our engineering expertise is demonstrated through implementations such as the Latticefold aggregation scheme, the Labrador proof system, zkvm-benchmarks, and Plonk Verifier in Cairo. This combined strength in theory and engineering enables us to deliver cutting-edge cryptographic solutions to partners and clients.

Smart Contract Development & DeFi Research: Our smart contract development and DeFi research team comprises 40+ world-class engineers who collaborate closely with partners to identify needs and work on value-adding projects. The team specializes in Solidity and Cairo development, architecture design, and DeFi solutions, including DEXs, AMMs, structured products, derivatives, and money market protocols, as well as ERC20, 721, and 1155 token design. Our research and data analytics focuses on three key areas: technical due diligence, market research, and DeFi research. Utilizing a data-driven approach, we offer in-depth insights and outlooks on various industry themes.

Our suite of L2 tooling: Warp is Starknet's approach to EVM compatibility. It allows developers to take their Solidity smart contracts and transpile them to Cairo, Starknet's smart contract language. In the short time since its inception, the project has accomplished many achievements, including successfully transpiling Uniswap v3 onto Starknet using Warp.

- **Voyager** is a user-friendly Starknet block explorer that offers comprehensive insights into the Starknet network. With its intuitive interface and powerful features, Voyager allows users to easily search for and examine transactions, addresses, and contract details. As an essential tool for navigating the Starknet ecosystem, Voyager is the go-to solution for users seeking in-depth information and analysis;
- **Horus** is an open-source formal verification tool for StarkNet smart contracts. It simplifies the process of formally verifying Starknet smart contracts, allowing developers to express various assertions about the behavior of their code using a simple assertion language;
- **Juno** is a full-node client implementation for Starknet, drawing on the expertise gained from developing the Nethermind Client. Written in Golang and open-sourced from the outset, Juno verifies the validity of the data received from Starknet by comparing it to proofs retrieved from Ethereum, thus maintaining the integrity and security of the entire ecosystem.

General Advisory to Clients

As auditors, we recommend that any changes or updates made to the audited codebase undergo a re-audit or security review to address potential vulnerabilities or risks introduced by the modifications. By conducting a re-audit or security review of the modified codebase, you can significantly enhance the overall security of your system and reduce the likelihood of exploitation. However, we do not possess the authority or right to impose obligations or restrictions on our clients regarding codebase updates, modifications, or subsequent audits. Accordingly, the decision to seek a re-audit or security review lies solely with you.

Disclaimer

This report is based on the scope of materials and documentation provided by you to Nethermind in order that Nethermind could conduct the security review outlined in **1. Executive Summary** and **2. Audited Files**. The results set out in this report may not be complete nor inclusive of all vulnerabilities. Nethermind has provided the review and this report on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. This report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on this report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, Nethermind disclaims any liability in connection with this report, its content, and any related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. Nethermind does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and Nethermind will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.