ENIGMA DARK

Securing the Shadows

Invariant Testing Engagement

Makina Core

July, 2025

Contents

Summary

Enigma Dark
Enigma Dark is a web3 security firm leveraging the best talent in the space to secure all kinds of blockchain protocols and decentralized apps. Our team comprises experts who have honed their skills at some of the best auditing companies in the industry. With a proven track record as highly skilled white-hats, they bring a wealth of experience and a deep understanding of the technology and the ecosystem.

Learn more about us at enigmadark.com

Makina Core
Makina is a protocol for executing advanced cross-chain investment strategies. It provides the infrastructure for operators to issue tokenized strategies with full DeFi composability and strong risk controls.

Engagement Overview

Over the course of 3 weeks, beginning 16 July 2025, the Enigma Dark team conducted an Invariant Testing engagement of the Makina Core project. The review was performed by one Lead Security Researcher: vnmrtz.

The following repositories were reviewed at the specified commits:

Repository	Commit
MakinaHQ/makina-core	`3021e35faf84e53ef663ca3135fcc5caffbef866`

Risk Classification

Severity	Description
Critical	Vulnerabilities that lead to a loss of a significant portion of funds of the system.
High	Exploitable, causing loss or manipulation of assets or data.
Medium	Risk of future exploits that may or may not impact the smart contract execution.
Low	Minor code errors that may or may not impact the smart contract execution.
Informational	Non-critical observations or suggestions for improving code quality, readability, or best practices.

Vulnerability Summary

Severity	Count	Fixed	Acknowledged
Critical	0	0	0
High	0	0	0
Medium	0	0	0
Low	0	0	0
Informational	1	1	0

Findings

Index	Issue Title	Status
I-01	Minor improvements to code and comments	Fixed

Invariants and Postconditions Summary

This table provides an overview of all invariants and postconditions defined in the suite specification files.

Property Types

On this invariant testing framework there exists two types of Properties:

INVARIANTS (INV)

- These are properties that should always hold true in the system
- They are implemented under `/invariants` folder

POSTCONDITIONS

- These are properties that should hold true after an action is executed
- They are implemented under `/hooks` and `/handlers` folders

There are two types of POSTCONDITIONS:

GLOBAL POSTCONDITIONS (GPOST)

- These are properties that should always hold true after any action is executed
- They are checked in the `_checkPostConditions` function within the HookAggregator contract

HANDLER-SPECIFIC POSTCONDITIONS (HSPOST)

- These are properties that should hold true after a specific action is executed in a specific context
- They are implemented within each handler function, under the HANDLER-SPECIFIC POSTCONDITIONS section

Summary

- **Total Properties**: 24
- **INVARIANTS**: 11
- **GPOSTCONDITIONS**: 4
- **HPOSTCONDITIONS**: 9
- **Status Distribution**:
    - PENDING: 0
    - PASS: 24
    - FAIL: 0
- **Resolution Distribution** (for FAIL properties):
    - FIXED: TODO fill this with values
    - ACKNOWLEDGED: TODO fill this with values

Table Structure

- **Property ID**: The unique identifier for each property
- **Description**: The detailed explanation of what the property should ensure
- **Status**: Current testing status (PASS/FAIL/PENDING)
- **Issues**: Related issues or notes when status is not PASS

INVARIANTS Properties

Property ID	Description	Status	Issues
INV_GAC_A	totalSupply = sum of all minted shares + accumulatedFees	PASS	-
INV_GAC_B	totalSupply == sum of balanceOf(actors) + accumulatedFees	PASS	-
INV_GAC_E	caliberBridgesIn <= machineBridgesOut	PASS	-
INV_GAC_F	machineBridgesIn <= caliberBridgesOut	PASS	-
INV_GAC_G	if machineBridgesOut > caliberBridgesIn, a bridge transfer is pending	PASS	-
INV_GAC_H	if caliberBridgesOut > machineBridgesIn, a bridge transfer is pending	PASS	-
INV_AVAIL_A	maxMint MUST NOT revert	PASS	-
INV_TIME_A	lastGlobalAccountingTime <= block.timestamp	PASS	-
INV_TIME_B	_lastMintedFeesTime <= block.timestamp	PASS	-
INV_BRIDGE_A	bridgeHubAdapter _reservedBalances[accountingToken] <= accountingToken balanceOf(bridgeHubAdapter)	PASS	-
INV_BRIDGE_B	bridgeSpokeAdapter _reservedBalances[baseToken] <= baseToken balanceOf(bridgeSpokeAdapter)	PASS	-

GPOSTCONDITIONS Properties

Property ID	Description	Status	Issues
GPOST_BASE_A	_lastGlobalAccountingTime should increase monotonically	PASS	-
GPOST_GAC_B	_lastMintedFeesTime should increase monotonically	PASS	-
GPOST_GAC_D	If totalSupplyAfter > _shareLimit, then totalSupplyBefore >= totalSupplyAfter, except for fee minting	PASS	-
GPOST_GAC_E	If fees were just minted, they cannot be minted again until cooldown passes	PASS	-

HPOSTCONDITIONS Properties

Property ID	Description	Status	Issues
HSPOST_USER_A	Exact assets of accountingToken should be transferred in to the machine on deposit	PASS	-
HSPOST_USER_A_2	Exact convertToAssets(shares) of accountingToken should be transferred out of the machine to the user on redeem	PASS	-
HSPOST_USER_C	lastTotalAum should increase by the deposited amount on deposit	PASS	-
HSPOST_USER_C_2	lastTotalAum should decrease by the redeemed amount on redeem	PASS	-
HSPOST_USER_D	redeem(deposit(a)) <= a	PASS	-
HSPOST_USER_E	deposit(redeem(a)) <= a	PASS	-
HSPOST_AVL_A	updateTotalAum under specific conditions MUST NOT REVERT	PASS	-
HSPOST_CNV_A	convertToAssets(convertToShares(a)) <= a	PASS	-
HSPOST_CNV_B	convertToShares(convertToAssets(a)) <= a	PASS	-

Status Definitions

- **PENDING**: Property has not been tested yet or testing is in progress
- **PASS**: Property passes all tests and behaves as expected
- **FAIL**: The property has been tested and failed

Resolution Options (for FAIL status properties):

- **FIXED**: The underlying issue has been resolved by the protocol team, property should pass on re-testing
- **ACKNOWLEDGED**: The issue has been reviewed and acknowledged by the protocol team as acceptable risk/design decision

Note: Detailed explanations for all failing properties and their related issues can be found in the Findings section below.

Testing Workflow

1. Properties start as **PENDING** before testing
2. After initial testing, properties become **PASS** or **FAIL**
3. For **FAIL** status properties, the protocol team responds with either:
    - **FIXED**: Issue gets resolved (property should pass on re-testing)
    - **ACKNOWLEDGED**: Issue is accepted as acceptable risk/design decision

Detailed Findings

High Risk

No issues found.

Medium Risk

No issues found.

Low Risk

No issues found.

Informational

I-01 - Minor improvements to code and comments

Severity: Informational

Context:
See below.

Technical Details:

- DecimalsUtils.sol#L15 - Unnecessary cast to address `asset_`.
- BridgeController.sol#L132 - Consider rounding up the calculation of the minimum acceptable output amount in order to follow DeFi best practices and ensure bridge loss limits are never exceeded due to precision errors. In a more recent commit, the same round-down calculation was introduced in BridgeAdapter.sol#L254. Consider updating this logic to round up consistently across both contracts.

Developer Response:

- 1 - Fixed in commit: `3f461b8`.
- 2 - Fixed in commit: `7178dd4`.

Disclaimer

This report does not endorse or critique any specific project or team. It does not assess the economic value or viability of any product or asset developed by parties engaging Enigma Dark for security assessments. We do not provide warranties regarding the bug-free nature of analyzed technology or make judgments on its business model, proprietors, or legal compliance.

This report is not intended for investment decisions or project participation guidance. Enigma Dark aims to improve code quality and mitigate risks associated with blockchain technology and cryptographic tokens through rigorous assessments.

Blockchain technology and cryptographic assets inherently involve significant risks. Each entity is responsible for conducting their own due diligence and maintaining security measures. Our assessments aim to reduce vulnerabilities but do not guarantee the security or functionality of the technologies analyzed.

This security engagement does not guarantee against a hack. It is a review of the codebase during a specific period of time. Enigma Dark makes no warranties regarding the security of the code and does not warrant that the code is free from defects. By deploying or using the code, the project and users of the contracts agree to use the code at their own risk. Any modifications to the code will require a new security review.