Panoptic Multi Token Merkle Distributor

Security Review

Cantina Managed review by:

M4rio, Lead Security Researcher

September 22, 2025

Contents

1 Introduction

1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at cantina.xyz

1.2 Disclaimer

Cantina Managed provides a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While Cantina Managed endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that the Cantina Managed security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

1.3 Risk assessment

Severity level	**Impact: High**	**Impact: Medium**	**Impact: Low**
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings are a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

2 Security Review Summary

Panoptic is the perpetual, oracle-free options protocol built on the Ethereum blockchain.

From Sep 15th to Sep 16th the Cantina team conducted a review of multitoken-merkle-distributor on commit hash 0897d3ec. The team identified a total of **5** issues:

Issues Found

Severity	**Count**	**Fixed**	**Acknowledged**
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	1	1	0
Low Risk	1	1	0
Gas Optimizations	0	0	0
Informational	3	2	1
Total	**5**	**4**	**1**

3 Findings

3.1 Medium Risk

3.1.1 Anyone can call `claim` for any `account`

Severity: Medium Risk

Context: MerkleDistributor.sol#L61

Description: The function allows third parties to execute a claim for a given leaf, and tokens are always sent to the `account` encoded in the leaf. The trade‑off: if a leaf contains a wrong `account`, a third party can proactively claim to that wrong address, preventing later recovery via the admin's `withdrawUnclaimed`.

Recommendation: Consider using `msg.sender` to claim instead of account or a signature from the account.

Panoptic: We discussed this tradeoff when implementing. The risk that a third-party griefer identifies accounts that don't wish to receive their ERC20s feels low, to us. And the bright side of not requiring that `account == msg.sender` is that a user could execute a claim owned by their accountA from a second accountB - say, maybe accountA doesn't have gas or has some permissions that prevent it from calling `Distributor.claim`

However, upon further review, we're going to add this requirement. Users that fall into the bucket of being unable to call `Distributor.claim` from accountA should just work with us after the clawback period, and like you say we get the additional protection from third party griefers forcing claims onto people who don't wish to execute.

Fixed in commit 7733ed68.

Cantina Managed: Fix verified.

3.2 Low Risk

3.2.1 Admin key is a single EOA set to `msg.sender` and immutable

Severity: Low Risk

Context: MerkleDistributor.sol#L37

Description: `admin` is set to the deployer EOA and cannot be changed. If the key is lost or compromised, unclaimed funds after `withdrawableAt` can be withdrawn by an attacker, and there is no recovery path.

Recommendation: Pass the admin address in the constructor and use a multisig. Consider adding a one‑time admin transfer mechanism or making admin upgradable behind a multisig/DAO if governance changes are expected.

Panoptic: Fixed in commit 7733ed68.

Cantina Managed: Fix verified.

3.3 Informational

3.3.1 Token registry uses mapping + array instead of an enumerable set

Severity: Informational

Context: MerkleDistributor.sol#L17-L18

Description: `supportedTokens` is a `mapping(address => bool)` plus a separate `tokenList` array. This pattern works but can drift out of sync (e.g., accidental duplicates in `tokenList`, or forgetting to push/remove). Managing two data structures also increases maintenance. Furthermore they are not enforced anymore on claim so right now it seems they are only for offchain reading?

Recommendation: Use `EnumerableSet.AddressSet` from OZ for a single source of truth (add/remove/check/iterate safely). Consider enforcing them on claim or removing them at all.

Panoptic: Acknowledged. `supportedTokens` and `tokenList` are just informational, no need to optimise them further.

Cantina Managed: Acknowledged.

3.3.2 Lack of end-to-end Merkle tests before release

Severity: Informational

Context: *(No context files were provided by the reviewer)*

Description: Without fork/integration tests that attempt to claim every leaf, discrepancies in leaf encoding, amounts, or unsupported tokens can surface only after deployment, potentially DoS'ing some claims.

Recommendation: Add a pre-release script/test that:

1. Rebuilds the Merkle root from the JSON snapshot,
2. Attempts a fork dry-run claim for every leaf, and...
3. Checks contract balances cover all amounts for all tokens.

Panoptic: Fixed in commit 7733ed68.

Cantina Managed: Fix verified.

3.3.3 Using `MerkleProof.verify` (memory) instead of `verifyCalldata`

Severity: Informational

Context: MerkleDistributor.sol#L71

Description: `verify` copies the proof to memory. Since the proof is already `calldata`, using `verifyCalldata` avoids an unnecessary copy.

Recommendation: Replace with `MerkleProof.verifyCalldata(merkleProof, merkleRoot, node)`.

Panoptic: Fixed in commit 7733ed68.

Cantina Managed: Fix verified.