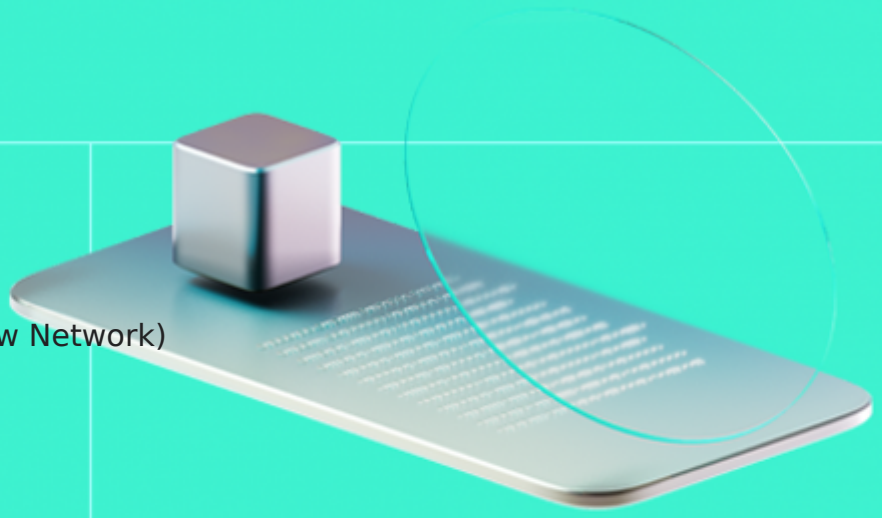
HACKEN

Smart Contract Code Review And Security Analysis Report



Customer: Openware (Yellow Network)

Date: 02/03/2023

We express our gratitude to the Openware (Yellow Network) team for the collaborative engagement that enabled the execution of this Smart Contract Security Assessment.

Yellow Network is a Layer-3 peer-to-peer network that uses state channels technology to scale and facilitate trading, clearing and settlement. The core technology is called ClearSync.

Platform: Ethereum, Solana, Polkadot, Avalanche, Other

Language: Solidity

Tags: ERC20

Timeline: 15/02/2023 - 02/03/2023

Methodology: https://hackenio.cc/sc_methodology

Review Scope

Repository	https://github.com/layer-3/clearsync
Commit	5b86a2134d295ac11af97d4f239782222e95fe24

Audit Summary

10/10	10/10	95.83%	10/10
Security score	Code quality score	Test coverage	Documentation quality score

Total 10/10

The system users should acknowledge all the risks summed up in the risks section of the report

2	0	0	1
Total Findings	Resolved	Accepted	Mitigated

Findings by severity

Critical	0
High	1
Medium	0
Low	1

Vulnerability	Status
F-2023-1091 - Highly Permissive Role Access	Mitigated
F-2023-1092 - Redundant View Functions	Pending Fix

This report may contain confidential information about IT systems and the intellectual property of the Customer, as well as information about potential vulnerabilities and methods of their exploitation.

The report can be disclosed publicly after prior consent by another Party. Any subsequent publication of this report shall be without mandatory consent.

Document

Name	Smart Contract Code Review and Security Analysis Report for Openware (Yellow Network)
Audited By	Hacken
Changelog	22/02/2023 – Initial Review 02/03/2023 - Second Review

Table of Contents

System Overview

YellowNetwork is a simple system with the following contracts:

- *Token* — simple ERC-20 token that allows mint, transfer and burn tokens with addition of blacklist. Addresses on blacklist cannot transfer tokens. While deploying contract developer needs to provide following attributes:
  -Name: token name.
  -Symbol: token symbol.
  -Supply Cap: minting over cap is not allowed.
  Contract uses roles to restrict access to important functions.
- *IBlacklist* — an interface that contains functions and events for blacklisting mechanisms.

Privileged roles

- DEFAULT_ADMIN_ROLE - address with that role can activate minting tokens.
- COMPLIANCE_ROLE - address with that role can add and remove given address from blacklist. Additionally, that role allows to burn all tokens from blacklisted addresses.
- MINTER_ROLE - address with that role can mint new tokens.

Executive Summary

This report presents an in-depth analysis and scoring of the customer's smart contract project. Detailed scoring criteria can be referenced in the scoring methodology.

Documentation quality

The total Documentation quality score is **10** out of **10**.

- Business logic is provided.
- Use cases are provided.
- Whitepaper is provided.
- Functional requirements are provided..

Code quality

The total Code quality score is **10** out of **10**.

- NatSpec is present.
- Code follows Solidity style guidelines.

Test coverage

Code coverage of the project is **95.83%** (branch coverage).

Security score

Upon auditing, the code was found to contain **0** critical, **1** high, **0** medium, and **1** low severity issues. Out of these, **0** issues have been addressed and resolved, leading to a security score of **10** out of **10**.

All identified issues are detailed in the “Findings” section of this report.

Summary

The comprehensive audit of the customer's smart contract yields an overall score of **10**. This score reflects the combined evaluation of documentation, code quality, test coverage, and security aspects of the project.

Risks

- **An account with the COMPLIANCE_ROLE can blacklist and then an account with DEFAULT_ADMIN_ROLE can burn tokens of any account without allowance. Tokens can be burned from contracts such as LPs**.
- The repository contains contracts that are out of the audit scope. Secureness and reliability of those contracts may not be guaranteed by the current audit.

Findings

Vulnerability Details

F-2023-1091 - Highly Permissive Role Access - High

Description: An account with the `COMPLIANCE_ROLE` can blacklist and then burn tokens of any account without allowance. Tokens can be burned from contracts such as LPs, which can lead to market manipulation.

Assets:

- contracts/Token.sol [https://github.com/layer-3/clearsync]

Status: Mitigated

Classification

Severity: High

Recommendations

Remediation: Do not burn tokens without allowance.

Resolution: The functionality is needed to protect the protocol. `COMPLIANCE_ROLE` can blacklist accounts. `DEFAULT_ADMIN_ROLE` can burn blacklisted funds. Both roles should be granted only to multisigs with ⅔ signatures required.

F-2023-1092 - Redundant View Functions - Low

Description: The variable `TOKEN_SUPPLY_CAP` is marked public. View function for the public variable is generated automatically by the compiler.

Assets:

- contracts/Token.sol [https://github.com/layer-3/clearsync]

Status: Pending Fix

Classification

Severity: Low

Recommendations

Remediation: Delete `cap()` function or mark `OKEN_SUPPLY_CAP` variable as private.

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed based on best industry practices at the time of the writing of this report, with cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The report contains no statements or warranties on the identification of all vulnerabilities and security of the code. The report covers the code submitted and reviewed, so it may not be relevant after any modifications. Do not consider this report as a final and sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements.

While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

English is the original language of the report. The Consultant is not responsible for the correctness of the translated versions.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the Consultant cannot guarantee the explicit security of the audited smart contracts.

Appendix 1. Severity Definitions

When auditing smart contracts, Hacken is using a risk-based approach that considers **Likelihood**, **Impact**, **Exploitability** and **Complexity** metrics to evaluate findings and score severities.

Reference on how risk scoring is done is available through the repository in our Github organization:

[hknio/severity-formula]

Severity	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to the loss of user funds or contract state manipulation.
High	High vulnerabilities are usually harder to exploit, requiring specific conditions, or have a more limited scope, but can still lead to the loss of user funds or contract state manipulation.
Medium	Medium vulnerabilities are usually limited to state manipulations and, in most cases, cannot lead to asset loss. Contradictions and requirements violations. Major deviations from best practices are also in this category.
Low	Major deviations from best practices or major Gas inefficiency. These issues will not have a significant impact on code execution, do not affect security score but can affect code quality score.

Appendix 2. Scope

The scope of the project includes the following smart contracts from the provided repository:

Scope Details

Repository	https://github.com/layer-3/clearsync
Commit	5b86a2134d295ac11af97d4f239782222e95fe24
Whitepaper	Provided
Requirements	Provided
Technical Requirements	Provided

Contracts in Scope

contracts/Token.sol

contracts/interfaces/IBlacklist.sol