QuillAudits

AUDIT REPORT

April , 2025

For

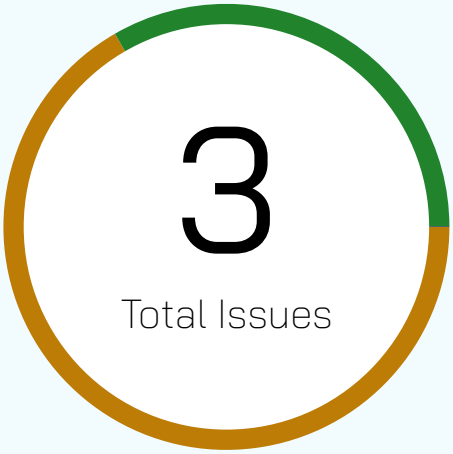
Sypher

Table of Content

Executive Summary

Project name	SypherFianace
Overview	SypherToken is a governance-enabled ERC20 token with built-in delegation. It tracks voting power via checkpoints (block-based snapshots) and allows users to delegate votes for on-chain governance. The system ensures efficient updates while maintaining historical vote accuracy
Audit Scope	The scope of this Audit was to analyze the Poly fund Smart Contracts for quality, security, and correctness. https://github.com/SypherFinance/sypherdao
Contracts in Scope	SypherToken.sol
Commit Hash	75a620282b1f34287dee1add3e1b53430eb21f79
Language	Solidity
Blockchain	Ethereum
Method	Manual Analysis, Functional Testing, Automated Testing
Review 1	21st April 2025 to 24th April 2025
Updated Code Received	NA
Review 2	NA
Fixed In	NA

Number of Issues per Severity



High	0 (0.00%)
Medium	2 (66.67%)
Low	1 (33.33%)
Informational	0 (0.00%)

Severity

Issues	High	Medium	Low	Informational
Open	0	0	0	0
Resolved	0	0	0	0
Acknowledged	0	**2**	**1**	0
Partially Resolved	0	0	0	0

Checked Vulnerabilities

- ✓ Access Management
- ✓ Arbitrary write to storage
- ✓ Centralization of control
- ✓ Ether theft
- ✓ Improper or missing events
- ✓ Logical issues and flaws
- ✓ Arithmetic Computations Correctness
- ✓ Race conditions/front running
- ✓ SWC Registry
- ✓ Re-entrancy
- ✓ Timestamp Dependence
- ✓ Gas Limit and Loops
- ✓ Exception Disorder
- ✓ Gasless Send
- ✓ Use of tx.origin
- ✓ Malicious libraries
- ✓ Compiler version not fixed
- ✓ Address hardcoded
- ✓ Divide before multiply
- ✓ Integer overflow/underflow
- ✓ ERC's conformance
- ✓ Dangerous strict equalities
- ✓ Tautology or contradiction
- ✓ Return values of low-level calls
- ✓ Missing Zero Address Validation
- ✓ Private modifier
- ✓ Revert/require functions
- ✓ Multiple Sends
- ✓ Using suicide
- ✓ Using delegatecall
- ✓ Upgradeable safety
- ✓ Using throw

- [x] Using inline assembly
- [x] Unsafe type inference
- [x] Style guide violation
- [x] Implicit visibility level

Techniques and Methods

Throughout the audit of smart contracts, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments, match logic and expected behavior.
- Token distribution and calculations are as per the intended behavior mentioned in the whitepaper.
- Implementation of ERC standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods, and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behavior of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools And Platforms Used For Audit

Remix IDE, Foundry, Solhint, Mythril, Slither, Solidity statistical analysis.

Types of Severity

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below

High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational Issues

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

Security vulnerabilities identified that must be resolved and are currently unresolved.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/ impact of the security issue, but are not completely resolved.

Medium Severity Issues

Partial Vote Delegation Not Supported.

Acknowledged

Path

SypherToken.sol

Function

_delegate()

Description

The current delegation system forces users to delegate their full balance, preventing partial delegation. This limits flexibility in governance participation, as token holders cannot split voting power between multiple delegates. The _delegate function transfers all votes without allowing partial amounts.

Recommendation

Modify the delegation logic to support partial delegation by allowing users to specify an amount when delegating.

Same-Block Delegation Vote Tracking Issue

Acknowledged

Path

SypherToken.sol

Function

_writeCheckpoint()

Description

The current checkpoint system does not properly handle multiple delegation or transfer actions occurring within the same block. When multiple vote changes happen in a single block:

1. The first transaction creates a checkpoint with fromBlock = currentBlock and the vote count at that time.

2. Subsequent transactions in the same block overwrite the vote value in the existing checkpoint instead of creating new entries.

Inaccurate Governance Snapshots: Only the final vote count is recorded for the block, discarding intermediate changes.
Potential Manipulation Risk: Malicious actors could exploit batched transactions to hide vote changes.

Recommendation

Set COOLDOWN_BLOCKS (e.g., 1 block) to prevent same-block updates.

Low Severity Issues

Lack of Contract Upgradeability in SypherToken

Acknowledged

Path

SypherToken.sol

Description

The current SypherToken contract is deployed as a non-upgradeable implementation, permanently locking all logic and storage layout without any mechanism for future improvements, bug fixes, or security patches, requiring full redeployment and token migration for any changes while breaking existing integrations and potentially jeopardizing user funds.

Recommendation

Implement an upgradeable proxy pattern (UUPS recommended).

Functional Tests

Some of the tests performed are mentioned below:

- The ERC20 contract works as expected
- A malicious user front-run the delegateBySig function by reading the signature ;but there was no impact on functionality
- In the given delegateBySig function
- the nonce is being used correctly
- The _delegate function works as expected
- Multiple votes can be delegated within the same block
- A huge iteration in a while loop can cause a gas issue
- No external/user can directly call _mint

Automated Tests

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of SypherFianace. We performed our audit according to the procedure described above.

Issues of medium and low severities were found , SypherFianace team acknowledged all of them.

Disclaimer

At QuillAudits, we have spent years helping projects strengthen their smart contract security. However, security is not a one-time event—threats evolve, and so do attack vectors. Our audit provides a security assessment based on the best industry practices at the time of review, identifying known vulnerabilities in the received smart contract source code.

This report does not serve as a security guarantee, investment advice, or an endorsement of any platform. It reflects our findings based on the provided code at the time of analysis and may no longer be relevant after any modifications. The presence of an audit does not imply that the contract is free of vulnerabilities or fully secure.

While we have conducted a thorough review, security is an ongoing process. We strongly recommend multiple independent audits, continuous monitoring, and a public bug bounty program to enhance resilience against emerging threats.

Stay proactive. Stay secure.

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem

QuillAudits

7+ Years of Expertise	**1M+** Lines of Code Audited
$30B+ Secured in Digital Assets	**1400+** Projects Secured

Follow Our Journey

AUDIT REPORT

April , 2025

For

Sypher

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com audits@quillaudits.com