QVAULT

Security Assessment and Code Audit

Mundus

19 Sept 2025

Rev 2

INTRODUCTION

The QVAULT Smart Contract's changes focus on updating several different types of proposals, each with its own cost and voting requirements. Below are how they work

- QCAP general proposal: electing team members, no impact on QCAP's investment
- Quorum requirement: adjustments to the quorum requirement
- IPO Participation: Investors cast votes to determine their investment amount in an IPO. QVAULT computes a weighted average, proportional to the number of $QCAP tokens held by each investor. Proposals are limited by the total Qubic amount available within the smart contract.
- QEarn participation: after a proposal gets approval, QVAULT locks a capital amount for a set period. Upon return, only the earned interest is distributed as revenue; the original capital remains in the QVAULT.
- Fundraising Proposal: If a proposal, such as the sale of assets, is approved, QVAULT will facilitate the sale of those tokens. The Qubic raised from this type of proposal will not be considered revenue and, therefore, should not be distributed.
- Marketplace proposal: a proposal, such as the sale of assets, is approved, the user receives payment. If not, or if QVAULT lacks sufficient funds, the assets are returned to the user.
    - An example selling assets: 10 Quottery and 1 Qx for 10B $Qubic and 10,000 $QCAP
- Allocation percentage: a proposal can change revenue allocation percentages, but the QVAULT revenue share is permanently fixed at 3%. The team's 2% fee will be zeroed out on January 1, 2027, and those funds will be reallocated to revenue distribution.
- Fund management: IPOs have the highest priority, followed by the marketplace, and then QEarn.

AUDIT PROCESS

Following considerations have been tested and reviewed by Mundus team:

- Reviewing google test files that are written by SC developer.
- Trying several attack vectors to exploit the SC.
- Ensuring contract logic meets the specifications.
- Thorough line-by-line manual review.

RESULTS

We identified a range of security issues in our assessment, from critical to minor. We recommend fixing them to meet security standards and best practices.

OVERVIEW

1. Project name: **QVAULT**
2. Platform: **QUBIC SC system**
3. Is new SC deployment: **NO**
4. Is upgraded SC: **YES**
5. Project size: Large
6. List of github commits to review:
    a. d45b621d7b4167aac3af9a9d4e3574e432ef2965
    b. 4de38c79bf829d68764a95b1675839d1d71bc0ca

RESULTS SUMMARY

Total Issues	6
Critical	0
Major	1
Medium	4
Minor	1

AUDIT SCOPE

Filename	SHA	Latest commit
QVault.h	4de38c79bf829d68764a95b1675839d1d71bc0ca	fix: second commit for audit
contract_qvault.cpp (gtest)	d45b621d7b4167aac3af9a9d4e3574e432ef2965	fix: fixed some parts by audit

STATUS OF FINDINGS IN REV 1

ID	Title	Category	Severity	Status
QVAULT-01	Numerous problems were identified in the current voteInProposal procedure's implementation	Logic	Major	Partial fixed
QVAULT-02	Centralization issue	Logic	Major	Not fixed
QVAULT-03	Not expandable size when QUBIC SC expanding	Logic	Medium	Fixed
QVAULT-04	The API call needs a check to see if it failed or succeeded	Logic	Medium	Partial fixed
QVAULT-05	Not verify that the number of stakers is within the accepted range	Logic	Medium	Fixed
QVAULT-06	Verify input amount for unStake procedure	Logic	Medium	Fixed
QVAULT-07	Invocators with QVAULT shares can vote regardless of voting power	Logic	Medium	Fixed
QVAULT-08	Incorrect updates if not verify input proposalType	Logic	Medium	Fixed
QVAULT-09	Lack of a return code and unvalidated input proposalId	Logic	Medium	Fixed
QVAULT-10	Found several issues with getIdentitiesHvVtPw	Logic	Medium	Fixed

QVAULT-11	Found two issues with getQcapBurntAmountInLastEpoches	Logic	Medium	Fixed
QVAULT-12	The temporary counter was not initialized	Logic	Medium	Fixed
QVAULT-13	Found several issues with getAmountForQearnInUpcomingEpoch	Logic	Medium	Fixed
QVAULT-14	Avoid hardcoded numbers	Logic	Minor	Fixed
QVAULT-15	Make a definition MAX_URLS_COUNT	Logic	Minor	Fixed
QVAULT-16	Make a definition MIN_VOTING_POWER	Logic	Minor	Fixed
QVAULT-17	Make definitions for proposal results	Logic	Minor	Fixed
QVAULT-18	Limit proposals not verified	Logic	Minor	Fixed
QVAULT-19	Port utility functions into QVAULT	Logic	Minor	Ack
QVAULT-20	Make a definition for allocation percentages	Logic	Minor	Fixed
QVAULT-21	Make definition for maximum of countOfVote	Logic	Minor	Fixed
QVAULT-22	Lack of a returnCode for getStakedAmountAndVotingPower	Logic	Minor	Fixed
QVAULT-23	Lack of a returnCode for ppCreationPower	Logic	Minor	Fixed

QVAULT-24	Verify the amount before transferring	Logic	Minor	Fixed

SUMMARY OF FINDINGS

ID	Title	Category	Severity
QVAULT-25	Lack of asset handling mechanism for transferred assets when the number of stakers hits the limit	Logic	Major
QVAULT-26	The API call needs a check to see if it failed or succeeded	Logic	Medium
QVAULT-27	Port utility functions into Qvault	Logic	Minor
QVAULT-28	Need to verify proposalId with number-of-proposals of input proposal's type	Logic	Medium
QVAULT-29	Need to verify input proposalId in multiple procedures/functions	Logic	Medium
QVAULT-30	Need to re-exam returnCode in ppCreationPower	Logic	Medium

FINDINGS

QVAULT-25 Lack of asset handling mechanism for transferred assets when the number of stakers hits the limit

- File(s) affected: QVault.h:732-736
- Description: For new stakers, if the number of stakers hits the QVAULT_X_MULTIPLIER limit, the assets transferred on line #716 must be rolled back.
- Recommendation:

QVAULT-26 The API call needs a check to see if it failed or succeeded

- File(s) affected: QVault.h:789
- Description: The API call needs a check to see if it failed or succeeded.
- Recommendation:

QVAULT-27 Port utility functions into Qvault

- File(s) affected: QVault.h:1300, 1301, 1525, 1526, 1832, 1833, 3237, 3238, 3287, 3288
- Description: Port these utility functions into Qvault, so that if they are removed or changed from outside then Qvault's logic will not be impacted.
- Recommendation:

QVAULT-28 Need to verify proposalId with number-of-proposals of input proposal's type

- File(s) affected: QVault.h:1638
- Description: Based on input.proposalType, retrieve the total number of proposals. Verify that input.proposalId is within the valid range (i.e., less than the total number of proposals). Failure to do so may lead to unexpected results in subsequent calls to get(input.proposalId).
- Recommendation:

QVAULT-29 Need to verify input proposalId in multiple procedures/functions

- File(s) affected: QVault.h:2050, 2086, 2122, 2158, 2194, 2230, 2266
- Description: Verify that input.proposalId is within the valid range (i.e., less than the current number-of-proposals **numberOfGP/numberOfQCP/numberOfIPOP…**). Failure to do so may lead to unexpected results in subsequent calls to get(input.proposalId).
- Recommendation:

QVAULT-30 Need to re-exam returnCode in ppCreationPower

- File(s) affected: QVault.h:2394,2401
- Description: Issues updating returnCode for the ppCreationPower function
  - Line 2394: Potential returnCode overwrite of line 2390
  - Line 2401: returnCode should be QVAULT_SUCCESS.
- Recommendation: