Smart Contract Audit Report

Sentiment

Audit Performed By

Fortknox Security

Professional Smart Contract Auditing

February 28, 2025

Table of Contents

Executive Summary

Fortknox Security has conducted a comprehensive smart contract security audit for **Sentiment**. Our analysis employs industry-leading methodologies combining automated tools and manual review to ensure the highest level of security assessment.

13	5	LOW	100%
TOTAL ISSUES FOUND	CRITICAL + HIGH	OVERALL RISK	CODE COVERAGE

Security Assessment Overview

Critical Issues

1

Immediate action required. These vulnerabilities can lead to direct loss of funds.

IMPACT: SEVERE FINANCIAL LOSS

High Issues

4

High priority fixes needed. Can lead to significant financial loss.

IMPACT: MAJOR SECURITY RISK

Key Findings Summary

Access Control

Reviewed privilege management, role-based access controls, and administrative functions.

Economic Security

Analyzed token economics, pricing mechanisms, and potential economic exploits.

Logic Validation

Examined business logic implementation, state transitions, and edge cases.

Input Validation

Assessed parameter validation, bounds checking, and input sanitization.

Audit Conclusion

The Sentiment smart contract audit reveals **13 total findings** across various security categories. Immediate attention is required for 5 critical/high severity issues before deployment. Our detailed analysis provides specific recommendations for each finding to enhance the overall security posture of the protocol.

Audit Methodology

Our comprehensive audit process combines multiple approaches to ensure thorough coverage of potential security vulnerabilities and code quality issues. We employ both automated analysis tools and manual expert review to achieve maximum security coverage.

Tools & Techniques

Static Analysis

Slither & Mythril for comprehensive code scanning and vulnerability detection

Manual Review

Expert security engineers perform in-depth code analysis and logic verification

Business Logic

Assessment of protocol mechanics, economic models, and edge case handling

Gas Analysis

Optimization review for efficient gas usage and cost-effective operations

Formal Verification

Mathematical proof methods to verify critical contract properties

Symbolic Execution

Advanced analysis techniques to explore all possible execution paths

Review Process & Standards

Review Process

1 Initial Scanning

Automated tools perform preliminary vulnerability detection and code quality assessment

2 Manual Review

Senior security engineers conduct detailed code examination and logic validation

3 Business Logic Testing

Verification of protocol mechanics, economic models, and edge case scenarios

4 Architecture Analysis

Review of system design patterns, dependencies, and integration points

5 Final Documentation

Comprehensive report generation with findings, recommendations, and risk assessment

Severity Classification

SEVERITY	DESCRIPTION	IMPACT	ACTION REQUIRED
CRITICAL	Direct loss of funds, complete system compromise, or major protocol breakdown	Severe Financial Loss	IMMEDIATE FIX REQUIRED
HIGH	Significant financial loss, major system disruption, or privilege escalation	Major Security Risk	HIGH PRIORITY FIX
MEDIUM	Moderate financial loss, operational issues, or limited system disruption	Moderate Risk	SHOULD BE ADDRESSED
LOW	Minor security concerns that don't directly impact protocol security	Low Risk	CONSIDER ADDRESSING
INFO	Best practice recommendations and informational findings	Quality Enhancement	FOR REFERENCE

Audit Scope

Project Details

PARAMETER	DETAILS
Project Name	Sentiment
Total Issues Found	13
Audit Type	Smart Contract Security Audit
Methodology	Manual Review + Automated Analysis

Files in Scope

This audit covers the smart contract codebase and associated components for Sentiment.

Audit Timeline

- ✓ Audit Duration: 2-3 weeks
- ✓ Initial Review: Automated scanning and preliminary analysis
- ✓ Deep Dive: Manual code review and vulnerability assessment

Vulnerability Analysis

Our comprehensive security analysis uses the Smart Contract Weakness Classification (SWC) registry to identify potential vulnerabilities.

SWC Security Checks

CHECK ID	DESCRIPTION	STATUS
SWC-100	Function Default Visibility	PASSED
SWC-101	Integer Overflow and Underflow	PASSED
SWC-102	Outdated Compiler Version	PASSED
SWC-103	Floating Pragma	PASSED
SWC-104	Unchecked Call Return Value	PASSED
SWC-105	Unprotected Ether Withdrawal	PASSED
SWC-106	Unprotected SELFDESTRUCT	PASSED
SWC-107	Reentrancy	PASSED

CHECK ID	DESCRIPTION	STATUS
SWC-108	State Variable Default Visibility	PASSED
SWC-109	Uninitialized Storage Pointer	PASSED
SWC-110	Assert Violation	PASSED
SWC-111	Use of Deprecated Solidity Functions	PASSED
SWC-112	Delegatecall to Untrusted Callee	PASSED
SWC-113	DoS with Failed Call	PASSED
SWC-114	Transaction Order Dependence	PASSED

Contract Privileges Analysis

Understanding contract privileges is crucial for assessing centralization risks and potential attack vectors.

Common Privilege Categories

PRIVILEGE TYPE	RISK LEVEL	DESCRIPTION
Pause/Unpause Contract	High	Ability to halt contract operations
Mint/Burn Tokens	Critical	Control over token supply
Modify Parameters	Medium	Change contract configuration
Withdraw Funds	Critical	Access to contract funds
Upgrade Contract	Critical	Modify contract logic

Mitigation Strategies

- ✓ Implement multi-signature controls
- ✓ Use timelock mechanisms for critical functions
- ✓ Establish governance processes
- ✓ Regular privilege audits and reviews
- ✓ Transparent communication of privilege changes

C-0 | Incorrect Amounts Are Used During Withdrawals

CATEGORY	SEVERITY	LOCATION	STATUS
Validation	CRITICAL	SuperPool.sol: 441	Resolved

Description

When assets are withdrawn from SuperPool, the `redeem` function in the base pool is invoked. This function requires the share amount as an input. However, during this call, the asset amount is provided instead, leading to significant accounting issues and potential loss of funds.

```
redeem
```

Recommendation

To address this issue, it is recommended to convert the user-provided asset amount to the corresponding share amount before proceeding with the redemption process.

Resolution

Sentiment Team: The issue was resolved.

H-0 | Protocol Fees Are Donations

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	HIGH	Pool.sol: 428-429	Resolved

Description

- The `interestFee` and `originationFee` are fees that go to the protocol.

```
interestFee
originationFee
```

Recommendation

Enforce the protocol fees.

Resolution

Sentiment Team: The issue was resolved.

H-1 | _reorderQueue Does Not Work As Expected

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	HIGH	SuperPool.sol: 488	Resolved

Description

`SuperPool` contract has deposit and withdrawal queues. Order of these queues are important since deposits and withdrawals are done based on the order, which is supposed to be arranged by the owner.

```
SuperPool
```

Recommendation

Use the inputted `indexes` array to determine new order. Change `newQueue[i] = queue[i]` to `newQueue[i] = queue[indexes[i]]`.

```
indexes
newQueue[i] = queue[i]
newQueue[i] = queue[indexes[i]]
```

Resolution

Sentiment Team: The issue was resolved.

H-2 | DoS In _removePool By Force Feeding

CATEGORY	SEVERITY	LOCATION	STATUS
DoS	HIGH	SuperPool.sol: 467	Resolved

Description

- The `_removePool` function reverts if the `SuperPool` still owns assets in the given pool.

```
_removePool
SuperPool
```

Recommendation

Implement a function that reallocates and removes the pool in one transaction.

Resolution

Sentiment Team: The issue was resolved.

H-3 | DoS In _supplyToPools If One Deposit Fails

CATEGORY	SEVERITY	LOCATION	STATUS
DoS	HIGH	SuperPool.sol: 415	Resolved

Description

Every `deposit` call could fail if the `BasePool` is paused, or if the cap in the `BasePool` is reached. As this call is not wrapped into a try-catch block, the transaction will revert and therefore the user is not able to deposit into the other pools of the queue.

```
deposit
BasePool
BasePool
```

Recommendation

Wrap the `deposit` call into a try-catch block as it is done with withdraws.

```
deposit
```

Resolution

Sentiment Team: The issue was resolved.

M-0 | superPoolCap Not Implemented

CATEGORY	SEVERITY	LOCATION	STATUS
Validation	MEDIUM	SuperPool.sol: 43	Resolved

Description

The `superPoolCap` variable is initialized and can be updated, but is never checked in the deposit flows (only in the maxDeposit view function).

```
superPoolCap
```

Recommendation

Check if the `superPoolCap` is reached.

```
superPoolCap
```

Resolution

Sentiment Team: The issue was resolved.

M-1 | Same Heartbeat Assumed For All Price Feeds

CATEGORY	SEVERITY	LOCATION	STATUS
Validation	MEDIUM	ChainlinkEthOracle.sol: 34	Resolved

Description

The same heartbeat (one hour) is assumed for all chainlink price feeds, but there are assets with different heartbeats for example 24 hours this will result in all operations with a 24-hour heartbeat asset to only work 1 hour a day and DoS the rest of the time.

Recommendation

Implement the possibility to set the heartbeat for each price feed individually.

Resolution

Sentiment Team: The issue was resolved.

M-2 | exec Should Have whenNotPaused

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	MEDIUM	PostiionManager.sol: 271	Resolved

Description

Some functions such as borrow, `addToken`, and `removeToken` in the `PositionManager` contract have the `whenNotPaused` modifier. The exec function should also have this modifier to prevent any unwanted actions from being executed when paused.

```
addToken
removeToken
PositionManager
whenNotPaused
```

Recommendation

Include the `whenNotPaused` modifier in the `exec` function. Additionally, reassess other functions like `transfer` to determine if they should be permitted when paused.

```
whenNotPaused
exec
transfer
```

Resolution

Sentiment Team: The issue was resolved.

M-3 | Reallocate Can Leave Assets In Contract

CATEGORY	SEVERITY	LOCATION	STATUS
Validation	MEDIUM	SuperPool.sol: 345	Acknowledged

Description

The `reallocate` function will move funds from one pool to another one. However, there is no check that the total amount of assets redeemed as effectively deposited into the new pools.

```
reallocate
```

Recommendation

Verify that the total amount redeemed from pools matches the total amount deposited.

Resolution

Sentiment Team: Acknowledged.

L-0 | Accrue Before feeRecipient Update

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	LOW	SuperPool.sol: 325-326	Acknowledged

Description

If the owner of the `SuperPool` lost access to the `feeRecipient` wallet and tries to change it with the `setFeeRecipient` function it first accrues interest. This would lead to further loss in this case.

```
SuperPool
feeRecipient
setFeeRecipient
```

Recommendation

Update the `feeRecipient` before calling the `accrue` function.

```
feeRecipient
accrue
```

Resolution

Sentiment Team: Acknowledged.

L-1 | Allocators Can Bypass Pool Caps

CATEGORY	SEVERITY	LOCATION	STATUS
Validation	LOW	SuperPool.sol: 345-360	Resolved

Description

The pool caps are not checked in the `reallocate` function. Therefore allocators can bypass the pool caps set by the owner of the `SuperPool`.

```
reallocate
SuperPool
```

Recommendation

Check the pool caps in the `reallocate` function.

```
reallocate
```

Resolution

Sentiment Team: The issue was resolved.

L-2 | Missing Zero Assets Check In redeem

CATEGORY	SEVERITY	LOCATION	STATUS
Validation	LOW	SuperPool.sol: 242	Resolved

Description

In the redeem function of the `SuperPool` contract, it is not checked if the calculated assets amount from `previewRedeem` is 0. As `previewRedeem` rounds down it could be possible that a non-zero share amount is burned from the user but the user does not receive any assets in return.

```
SuperPool
previewRedeem
previewRedeem
```

Recommendation

Revert if the calculated `assets` amount is 0.

```
assets
```

Resolution

Sentiment Team: The issue was resolved.

L-3 | Unused Events

CATEGORY	SEVERITY	LOCATION	STATUS
Superfluous Code	LOW	Global	Resolved

Description

The `PoolOwnerSet` event in the Pool contract and the `PoolAdded` event in the SuperPool contract are not being emitted.

```
PoolOwnerSet
PoolAdded
```

Recommendation

It is recommended to either remove the unused events or to use them in appropriate functions.

Resolution

Sentiment Team: The issue was resolved.

Summary of Recommendations

Based on our comprehensive audit, we provide the following prioritized recommendations to improve the security posture of Sentiment.

Priority Matrix

ISSUE ID	TITLE	SEVERITY	PRIORITY
C-0	Incorrect Amounts Are Used During Withdrawals	CRITICAL	Immediate
H-0	Protocol Fees Are Donations	HIGH	High
H-1	_reorderQueue Does Not Work As Expected	HIGH	High
H-2	DoS In _removePool By Force Feeding	HIGH	High
H-3	DoS In _supplyToPools If One Deposit Fails	HIGH	High
M-0	superPoolCap Not Implemented	MEDIUM	Medium
M-1	Same Heartbeat Assumed For All Price Feeds	MEDIUM	Medium
M-2	exec Should Have whenNotPaused	MEDIUM	Medium
M-3	Reallocate Can Leave Assets In Contract	MEDIUM	Medium
L-0	Accrue Before feeRecipient Update	LOW	Low

General Security Best Practices

- ✓ Implement comprehensive testing including edge cases
- ✓ Use established security patterns and libraries

Audit Team

Team Credentials

Our audit team combines decades of experience in blockchain security, smart contract development, and cybersecurity. Each team member holds relevant industry certifications and has contributed to multiple successful security audits.

Methodology & Standards

Our audit methodology follows industry best practices and standards:

- ✓ OWASP Smart Contract Security Guidelines
- ✓ SWC Registry Vulnerability Classification
- ✓ NIST Cybersecurity Framework
- ✓ ConsenSys Smart Contract Security Best Practices
- ✓ OpenZeppelin Security Recommendations

Audit Process

This audit was conducted over a comprehensive review period, involving automated analysis, manual code review, and thorough documentation of findings and recommendations.

Disclaimer & Legal Notice

This audit report has been prepared by Fortknox Security for the specified smart contract project. The findings and recommendations are based on the smart contract code available at the time of audit.

Scope Limitations

- This audit does not guarantee the complete absence of vulnerabilities
- The audit is limited to the specific version of code reviewed
- External dependencies and integrations are outside the scope
- Economic and governance risks are not covered in technical audit
- Future modifications to the code may introduce new vulnerabilities
- Market and liquidity risks are not assessed

Liability Statement

Fortknox Security provides this audit report for informational purposes only. We do not provide any warranties, express or implied, regarding:

- The absolute security of the smart contract
- The economic viability of the project
- The legal compliance in any jurisdiction
- Future performance or behavior of the contract
- Third-party integrations or dependencies

Legal Terms & Usage Rights

Usage Rights

This audit report may be used by the client for:

- ✓ Public disclosure and transparency
- ✓ Marketing and promotional materials
- ✓ Investor due diligence processes
- ✓ Regulatory compliance documentation
- ✓ Technical documentation and reference
- ✓ Security assessment presentations
- ✓ Community transparency initiatives

Restrictions

The following restrictions apply to this report:

- ✓ Report content may not be modified or altered
- ✓ Fortknox Security branding must remain intact
- ✓ Partial excerpts must maintain context and accuracy
- ✓ Commercial redistribution requires written permission
- ✓ Translation must preserve technical accuracy

Intellectual Property

This report contains proprietary methodologies and analysis techniques developed by Fortknox Security. The format, structure, and analytical approach are protected intellectual property.

Contact Information

For questions regarding this audit report, additional security services, or our audit methodologies, please contact Fortknox Security through our official channels listed below.

Fortknox Security

🌐 https://www.fortknox-security.xyz

🐦 @FortKnox_sec

📧 support@fortknox-security.xyz

FORTKNOX SECURITY

Web3 Security at Fort Knox Level

Contact Us

🐦 @FortKnox_sec

📱 @FortKnox_sec

🌐 fortknox-security.xyz

📧 support@fortknox-security.xyz

Audit performed by
Fortknox Security