



# Smart Contract Audit Report

AnimeCoin-Tech

## Audit Performed By

Fortknox Security  
Professional Smart Contract Auditing

June 14, 2024



## Table of Contents

|                              |    |
|------------------------------|----|
| Executive Summary            | 3  |
| Audit Methodology            | 5  |
| Audit Scope                  | 8  |
| Vulnerability Analysis       | 9  |
| Contract Privileges Analysis | 11 |
| Detailed Findings            | 8  |
| Recommendations              | 9  |
| Audit Team                   | 20 |
| Disclaimer & Legal Notice    | 21 |
| Legal Terms & Usage Rights   | 22 |



## Executive Summary

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

Q

7

TOTAL ISSUES FOUND

⚠

0

CRITICAL + HIGH

i

LOW

✓

100%

OVERALL RISK

CODE COVERAGE

## Security Assessment Overview



### Critical Issues

0

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

IMPACT: SEVERE FINANCIAL LOSS



### High Issues

0

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 AnimeCoin-Tech smart contract audit reveals **7 total findings** across various security categories. **No critical or high severity issues were identified.** 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       | AnimeCoin-Tech                     |
| Total Issues Found | 7                                  |
| 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 AnimeCoin-Tech.

## 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



# M-0 | setReadChannel Always Assigns lzReadChannel

| CATEGORY      | SEVERITY | LOCATION              | STATUS   |
|---------------|----------|-----------------------|----------|
| Logical Error | MEDIUM   | AnimeClaimer.sol: 710 | Resolved |

## Description

In the `setReadChannel` function the `active` parameter determines whether the channel is being activated or deactivated. However the `$.lzReadChannel = channelId;` assignment is always made no matter the `active` value.

```
setReadChannel
active
$.lzReadChannel = channelId;
active
```

## Recommendation

Do not assign the `lzReadChannel` value if active is false.

```
lzReadChannel
```

## Resolution

Animecoin Team: Resolved.



## M-1 | ERC721 Delegations With Rights Are Missed

| CATEGORY      | SEVERITY | LOCATION             | STATUS   |
|---------------|----------|----------------------|----------|
| Logical Error | MEDIUM   | ClaimChecker.sol: 90 | Resolved |

### Description

In the `_checkNFTClaim` function the following boolean condition is used to determine if the claimer is authorized to initiate the ANIME claim for the `nftOwner`.

```
_checkNFTClaim  
nftOwner
```

### Recommendation

Create a new `_checkDelegateV2RightsForNft` function which uses `checkDelegateForERC721`.

### Resolution

Animecoin Team: Resolved



## L-0 | Unused DelegateCheckerLib

| Category      | Severity | Location              | Status   |
|---------------|----------|-----------------------|----------|
| Logical Error | LOW      | ClaimChecker.sol: 101 | Resolved |

### Description

In the `ClaimChecker` contract the `_DELEGATE_REGISTRY_V2` is called directly in the `_checkDelegateV2Rights` function without using the `DelegateCheckerLib`.

```
ClaimChecker
_DELEGATE_REGISTRY_V2
_checkDelegateV2Rights
DelegateCheckerLib
```

### Recommendation

Consider using the `DelegateCheckerLib` for the `SUBDELEGATION_RIGHTS_KEY` check as well.

```
DelegateCheckerLib
SUBDELEGATION_RIGHTS_KEY
```

### Resolution

Animecoin Team: Resolved.



## L-1 | Claim Requests Allowed When At Daily Limit

| CATEGORY      | SEVERITY | LOCATION         | STATUS       |
|---------------|----------|------------------|--------------|
| Logical Error | LOW      | AnimeClaimer.sol | Acknowledged |

### Description

In the `requestClaim` function there is no validation that prevents users from requesting claims that will trivially fail when the daily withdrawal limit has been reached.

```
requestClaim
```

### Recommendation

Consider adding validation which prevents user3 from requesting claims when the daily withdrawal limit has already been reached to save users from potentially wasting gas and LayerZero fees.

### Resolution

Animecoin Team: Acknowledged.



## L-2 | Duplicate DVN Configured

| CATEGORY      | SEVERITY | LOCATION                | STATUS   |
|---------------|----------|-------------------------|----------|
| Logical Error | LOW      | layerzero.config.ts: 42 | Resolved |

### Description

The `LayerZero` DVN is unnecessarily repeated as a required DVN and an optional DVN, however for the desired behavior of 2/3 DVNs with one of them always being `LayerZero` then the `LayerZero` DVN should be included only once in the `requiredDvns` list with the optional DVNs list being the other DVNs with an `optionalDVNTreshold` of 1.

```
LayerZero
LayerZero
LayerZero
requiredDvns
optionalDVNTreshold
```

### Recommendation

Remove the `LayerZero` DVN from the `optionalDVNs` list as well as another of the DVNs from the `optionalDVNs` list. Then reduce the `optionalDVNTreshold` to 1 to achieve a 2/3 total DVNs verification where one DVN must always be `LayerZero`.

```
LayerZero
optionalDVNs
optionalDVNs
optionalDVNTreshold
LayerZero
```

### Resolution



## L-3 | Lacking Maximum Configs Validation

| CATEGORY      | SEVERITY | LOCATION              | STATUS   |
|---------------|----------|-----------------------|----------|
| Logical Error | LOW      | AnimeClaimer.sol: 302 | Resolved |

### Description

In the `AnimeClaimer` contract the `requestClaim` function does not include any validation on the maximum length of the `config` list.

AnimeClaimer  
config

### Recommendation

Consider validating that the `configs.length` is within a reasonable limit.

### Resolution

Animecoin Team: Resolved.



## L-4 | Contracts Are Whitelisted For All Claims

| CATEGORY      | SEVERITY | LOCATION                  | STATUS       |
|---------------|----------|---------------------------|--------------|
| Logical Error | LOW      | ClaimChecker.sol: 81, 102 | Acknowledged |

### Description

The `explicitContractClaimers` mapping is used both for NFT claims and collector claims. Therefore if a user explicitly designates a contract claimer then that claimer is able to claim all NFT claims and collector level claims for the user.

`explicitContractClaimers`

### Recommendation

Be aware of this behavior, if intended then be sure to document it clearly to users.

### Resolution

Animecoin Team: Acknowledged.



# Summary of Recommendations

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

## Priority Matrix

| Issue ID | Title                                       | Severity | Priority |
|----------|---|----------|----------|
| M-0      | setReadChannel Always Assigns IzReadChannel | MEDIUM   | Medium   |
| M-1      | ERC721 Delegations With Rights Are Missed   | MEDIUM   | Medium   |
| L-0      | Unused DelegateCheckerLib                   | LOW      | Low      |
| L-1      | Claim Requests Allowed When At Daily Limit  | LOW      | Low      |
| L-2      | Duplicate DVN Configured                    | LOW      | Low      |
| L-3      | Lacking Maximum Configs Validation          | LOW      | Low      |
| L-4      | Contracts Are Whitelisted For All Claims    | LOW      | Low      |

## General Security Best Practices

- ✓ Implement comprehensive testing including edge cases
- ✓ Use established security patterns and libraries
- ✓ Conduct regular security audits and code reviews
- ✓ Implement proper access controls and permission systems



## 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](http://fortknox-security.xyz)

 [support@fortknox-security.xyz](mailto:support@fortknox-security.xyz)

Audit performed by  
Fortknox Security