FINGUARD
AI-POWERED WEB3 SECURITY

SAMPLE TECHNICAL TEMPLATE

# FULL TECHNICAL SMART CONTRACT AUDIT REPORT

Code-level findings, proof-of-concept, testing and remediation verification

PROJECT

## Example DeFi Protocol

FINAL SECURITY GRADE

**A**

86 / 100

PASSED WITH CONDITIONS

REPORT ID
FG-2026-TEMPLATE-001

COMMIT
4e9a1c6b2d7f

NETWORK
ETHEREUM

VERSION
FULL v1.0

Template data only. Replace with verified project scope, findings, tests and remediation evidence.

finguardaudit.com

# Document Control

## 01 Report Identity

| FIELD | VALUE |
| --- | --- |
| Report ID | FG-2026-TEMPLATE-001 |
| Report type | Full Technical Audit |
| Project | Example DeFi Protocol |
| Network | Ethereum Mainnet |
| Audit window | 01 - 10 July 2026 |
| Reviewed commit | 4e9a1c6b2d7f |
| Final version | v1.0 |
| Status | Verified Template |

## 02 Distribution & Classification

This full technical report is intended for the project development team, security reviewers, legal and compliance stakeholders, and authorized partners. The public summary report should be used for general publication.

Classification: Project Confidential until publication approval. The verified public copy must match the report fingerprint shown on Finguard.

# Executive Summary

| SECURITY SCORE | CONTRACTS | SOURCE LINES | FINDINGS |
|---|---|---|---|
| 86 /100 | 12 reviewed | 4,820 SLOC | 9 total |

## 01 Final Assessment

PASSED WITH CONDITIONS

No unresolved Critical or High severity vulnerabilities remain in the reviewed commit. One Medium severity centralization risk is acknowledged and requires operational controls. The assessment applies only to the submitted repository, commit and deployment assumptions.

## 02 Finding Distribution

| CRITICAL | HIGH | MEDIUM | LOW | INFO |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 3 |

# Audit Scope

## 01 Reviewed Components

| CONTRACT | PATH | LOC | STATUS |
|---|---|---|---|
| Token.sol | contracts/token/Token.sol | 286 | Audited |
| Vault.sol | contracts/core/Vault.sol | 512 | Audited |
| Staking.sol | contracts/staking/Staking.sol | 438 | Audited |
| OracleAdapter.sol | contracts/oracle/OracleAdapter.sol | 194 | Audited |
| FeeManager.sol | contracts/fees/FeeManager.sol | 221 | Audited |
| ProxyAdmin.sol | contracts/proxy/ProxyAdmin.sol | 158 | Audited |

## 02 Scope Binding

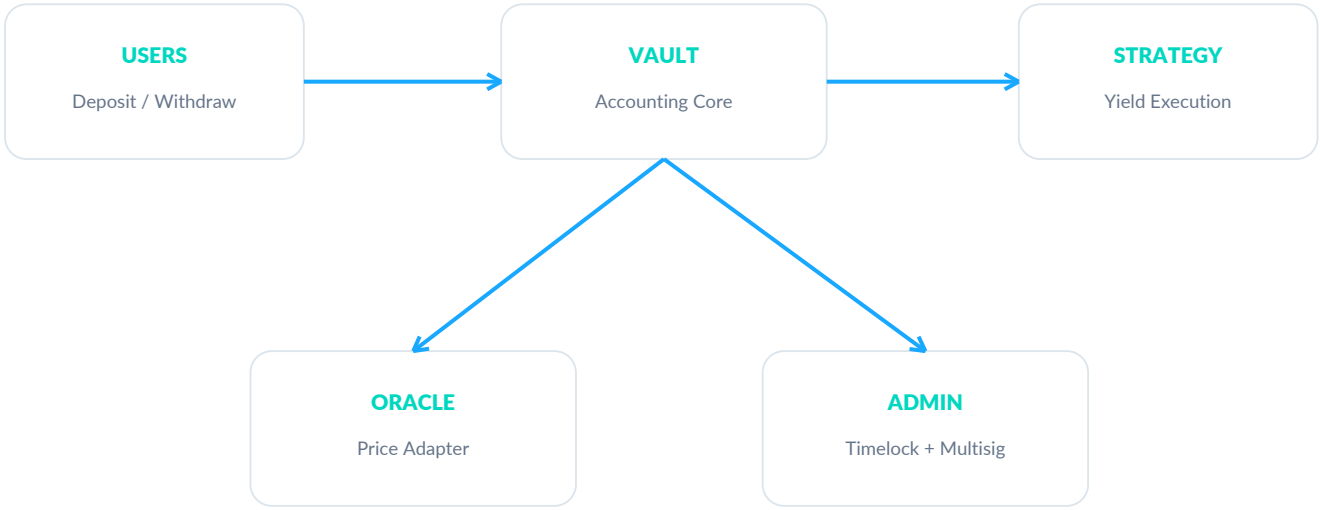
| | |
|---|---|
| REPOSITORY | `github.com/example/protocol` |
| COMMIT | `4e9a1c6b2d7f1ab9c2fd` |
| COMPILER | `Solidity 0.8.24` |
| OPTIMIZATION | `Enabled, 200 runs` |

## 03 Out of Scope

- Front-end application and DNS configuration - Private key custody and operational security - Third-party oracle infrastructure availability - Contracts not present in the reviewed commit - Future upgrades after the audit date

# System Architecture

## 01 Component Flow




## 02 Trust Boundaries

| ID | BOUNDARY | SECURITY CONCERN | RISK |
|---|---|---|---|
| TB-01 | Admin to Proxy | Upgrade authority | High |
| TB-02 | Oracle to Vault | External price input | High |
| TB-03 | Vault to Strategy | Asset movement | High |
| TB-04 | User to Vault | Untrusted calldata | Medium |

# Audit Methodology

## 01 Static Analysis

Pattern detection, dataflow review, storage and proxy checks.

## 02 AI Semantic Review

Business logic, hidden privileges and unsafe state transitions.

## 03 Manual Code Review

Line-by-line validation of critical asset and authorization paths.

## 04 Fuzz Testing

Property-based randomized execution across boundary conditions.

## 05 Invariant Testing

Continuous validation of accounting and solvency properties.

## 06 Economic Simulation

Flash-loan, oracle and reward extraction attack scenarios.

# Automated Toolchain

## 01 Execution Record

| ENGINE | VERSION | PURPOSE | STATUS |
|---|---|---|---|
| Slither | 0.10.x | Static analysis | Completed |
| Foundry | 1.2.x | Unit / fuzz / invariant | Completed |
| Echidna | 2.2.x | Property fuzzing | Completed |
| Mythril | 0.24.x | Symbolic execution | Completed |
| Finguard AI | 2026.07 | Semantic review | Completed |
| Finguard Monte Carlo | v3 | Economic paths | Completed |

## 02 Evidence Handling

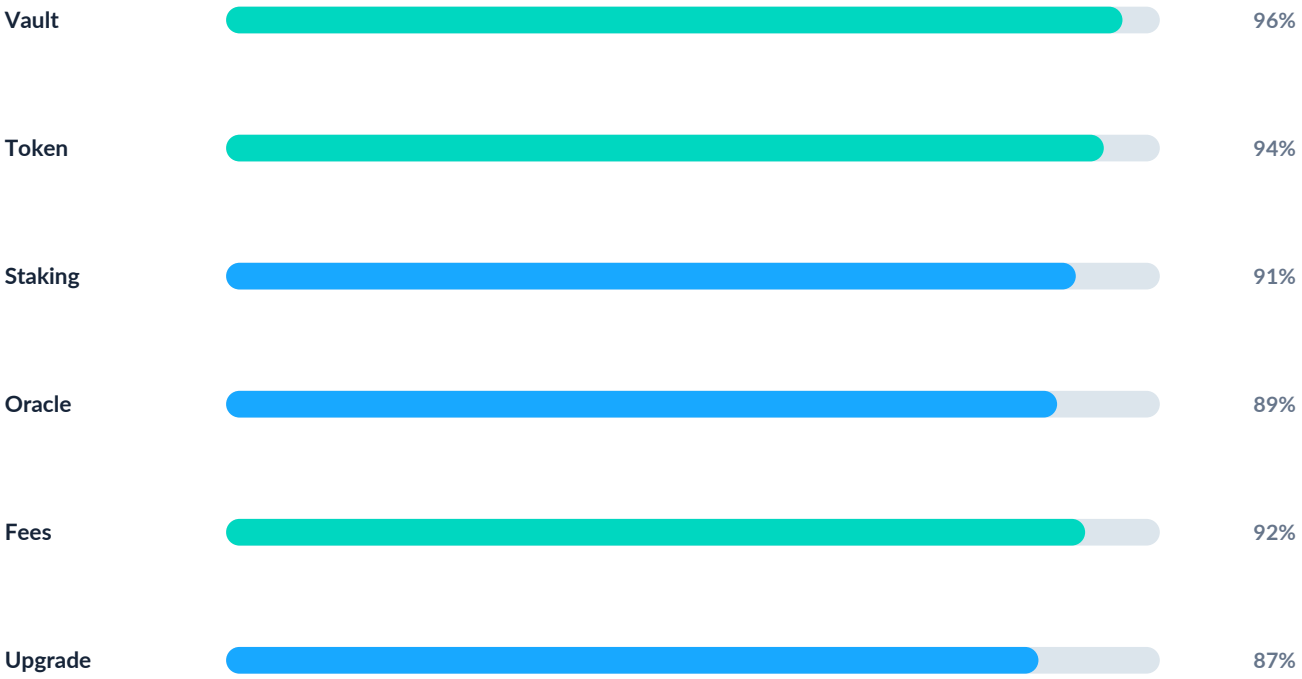
Tool output was treated as evidence requiring validation, not as a final finding. Each confirmed issue was reproduced, assigned impact and exploitability, mapped to affected code, and re-tested after remediation.

False positives, duplicate detections and non-exploitable patterns were excluded from the final finding count.

# Testing & Coverage

| UNIT TESTS | FUZZ CASES | INVARIANTS | ATTACK PATHS |
|---|---|---|---|
| 214 passed | 1.2M executions | 18 properties | 10M simulated |

## 01 Coverage by Module




## 02 Key Invariants

- Total shares never exceed issued accounting shares. - Vault assets remain greater than or equal to withdrawable liabilities. - Unauthorized callers cannot upgrade, mint or change fees. - Oracle deviation checks reject abnormal price movements. - Fees never exceed the configured maximum.

# Threat Model

## 01 Adversary Classes

| ACTOR | CAPABILITIES | PRIMARY CONCERN |
|---|---|---|
| External attacker | No privileges, arbitrary capital | Flash loans, callbacks, calldata |
| Malicious user | Valid account and deposits | Accounting and reward abuse |
| Compromised operator | Operator role only | Pause and parameter misuse |
| Compromised admin | Upgrade or ownership access | Implementation replacement |
| Oracle manipulator | Market or feed influence | Price distortion |

## 02 Security Objectives

- **Asset safety** — No unauthorized transfer or irreversible lock.
- **Solvency** — Liabilities remain fully backed by assets.
- **Authorization** — Critical functions require explicit trusted roles.
- **Price integrity** — External prices cannot be cheaply manipulated.
- **Upgrade safety** — Changes are delayed, visible and multi-party approved.

# Findings Overview

| ID | FINDING | SEVERITY | STATUS |
|---|---|---|---|
| FG-01 | Manipulable spot-price oracle | High | Resolved |
| FG-02 | Missing minimum-out protection | Medium | Resolved |
| FG-03 | Upgrade admin centralization | Medium | Acknowledged |
| FG-04 | Unbounded reward iteration | Low | Resolved |
| FG-05 | Missing zero-address validation | Low | Resolved |
| FG-06 | Inconsistent event emission | Low | Resolved |
| FG-07 | NatSpec documentation gaps | Info | Open |
| FG-08 | Gas optimization opportunity | Info | Open |
| FG-09 | Explicit compiler pinning | Info | Resolved |

# Finding Detail

FG-01

## Manipulable Spot-Price Oracle

HIGH RESOLVED

### 01 Description

The vault used the current reserve ratio of a low-liquidity automated market maker pair as the collateral price. An attacker could temporarily distort the pool using a flash loan, cause the vault to overvalue collateral, borrow assets, then restore the market within the same transaction.

### 02 Affected Code

SOLIDITY / TEST EXCERPT

```
01  function assetPrice() public view returns (uint256) {
02      (uint112 r0, uint112 r1,) = pair.getReserves();
03      return uint256(r1) * 1e18 / uint256(r0);
04  }
```

### 03 Impact

A successful exploit could create under-collateralized debt and cause direct protocol losses. Impact was rated High because profitability depended on available pool liquidity and vault borrowing limits.

# Attack & Remediation

## 01 Proof-of-Concept Sequence

1. Borrow temporary capital through a flash loan.
2. Swap against the target AMM pair to distort reserves.
3. Deposit collateral while the vault reads the manipulated price.
4. Borrow protocol assets against the inflated valuation.
5. Reverse the swap, repay the flash loan and retain profit.

## 02 Remediation

The project replaced the spot-price calculation with a 30-minute time-weighted average price oracle. A maximum 3% deviation check and stale-price validation were added. Borrowing is paused when the oracle is invalid.

## 03 Verification

| EVIDENCE | RESULT |
| --- | --- |
| Commit | e28c91f3b14a |
| Regression tests | 12 passed |
| Fuzz executions | 350,000 |
| Result | Exploit no longer reproducible |

# Finding Detail

FG-02

## Missing Minimum-Out Protection

MEDIUM RESOLVED

### 01 Description

The strategy executed token swaps without a user- or governance-defined minimum output. During periods of high volatility or miner-extractable value activity, the transaction could settle at an unfavorable price.

### 02 Vulnerable Pattern

SOLIDITY / TEST EXCERPT

```
01  router.swapExactTokensForTokens(
02      amountIn,
03      0, // no minimum output protection
04      path,
05      address(this),
06      block.timestamp
07  );
```

### 03 Remediation

A quote-based minimum output parameter is now calculated from the configured maximum slippage. The function reverts when the returned amount is below the threshold.

### 04 Status Evidence

RESOLVED Verified commit: 2d99a74f

# Finding Detail

FG-03

## Upgrade Administrator Centralization

MEDIUM ACKNOWLEDGED

### 01 Description

A single externally owned account was initially able to upgrade the proxy implementation. A compromised key could replace trusted logic and gain control over vault assets.

### 02 Risk Context

| ATTRIBUTE | VALUE |
| --- | --- |
| Current owner | `0x7A2c...91F4` |
| Account type | Externally owned account |
| Upgrade delay | None in initial deployment |
| Potential impact | Complete implementation replacement |

### 03 Project Response

The team accepted the risk for the initial test deployment and committed to transfer upgrade authority to a 3-of-5 multisig protected by a 48-hour timelock before production activation.

### 04 Required Operational Control

- Publish multisig signers and timelock address. - Monitor queued upgrades on-chain. - Require independent review before execution.

# Low Severity Findings

FG-04

RESOLVED

## Unbounded reward iteration

A loop over all reward epochs could exceed the block gas limit after long inactivity. Processing is now capped per transaction.

FG-05

RESOLVED

## Missing zero-address validation

Administrative setters accepted the zero address and could disable required integrations. Explicit checks were added.

FG-06

RESOLVED

## Inconsistent event emission

Two parameter updates did not emit events, reducing monitoring visibility. Events were added.

# Informational Findings

| ID | TOPIC | STATUS | RECOMMENDATION |
|---|---|---|---|
| FG-07 | NatSpec documentation gaps | Open | Add complete parameter and return descripti... |
| FG-08 | Gas optimization opportunity | Open | Cache repeated storage reads in reward calcu... |
| FG-09 | Explicit compiler pinning | Resolved | Use exact 0.8.24 compiler version in product... |

## 01 Interpretation

Informational findings do not represent direct exploitable vulnerabilities. They improve maintainability, monitoring, auditability or deployment consistency and should be addressed as part of normal engineering work.

# Privileged Roles

| ROLE | ADDRESS | AUTHORITY | RISK |
|---|---|---|---|
| Proxy Admin | `0x7A2c...91F4` | Upgrade implementation | High |
| Protocol Admin | `0x31D9...A201` | Fees, limits, integrations | Medium |
| Guardian | `0x88F1...D90C` | Pause deposits / borrows | Medium |
| Treasury | `0x00B4...77E2` | Withdraw collected fees | Medium |

## 01 Control Recommendations

### Multisig

Use 3-of-5 or stronger approval for upgrade and treasury roles.

### Timelock

Apply at least 48 hours to implementation and critical parameter changes.

### Monitoring

Alert users when upgrades are queued, executed or cancelled.

### Least privilege

Separate pause, fee, treasury and upgrade permissions.

# Economic Attack Simulation

| ATTACK PATHS | PROFITABLE | MAX LOSS | POST-FIX |
|---|---|---|---|
| 10,000,000 simulated | 18 pre-fix | $284K pre-fix | 0 confirmed |

## 01 Vector Results

| VECTOR | RESULT | RISK | STATE |
|---|---|---|---|
| Oracle manipulation | Exploitable pre-fix | High | Resolved |
| Flash-loan composition | Enabled oracle exploit | High | Resolved |
| Reward extraction | Not profitable | Low | Passed |
| Liquidity drain | Blocked by accounting | Low | Passed |
| Governance abuse | Role dependent | Medium | Operational |

## 02 Interpretation

Monte Carlo output is a simulation result and not a guarantee of future security. Confirmed findings require deterministic reproduction and code-level validation. Post-remediation tests did not reproduce a profitable oracle attack under the reviewed assumptions.

# Remediation Verification

| RESOLVED | ACKNOWLEDGED | OPEN | REGRESSION |
|---|---|---|---|
| 7 findings | 1 finding | 1 info only | 100% passed |

## 01 Retest Matrix

| ID | VERIFICATION | COMMIT / EVIDENCE | RESULT |
|---|---|---|---|
| FG-01 | Oracle attack | e28c91f3 | Passed |
| FG-02 | Slippage protection | 2d99a74f | Passed |
| FG-03 | Admin centralization | Policy action | Acknowledged |
| FG-04 | Reward iteration | 875bca11 | Passed |
| FG-05 | Address validation | 9ae2b1c0 | Passed |
| FG-06 | Event emission | 55e71fc2 | Passed |
| FG-09 | Compiler pinning | 70c93dd1 | Passed |

# Final Assessment

PASSED WITH CONDITIONS

## Security Grade A - 86 / 100

The reviewed codebase demonstrates a strong security posture after remediation. The confirmed High severity oracle issue and associated slippage weakness were corrected and verified. Production deployment remains conditional on transferring upgrade authority to the planned multisig and timelock.

### 01 Deployment Conditions

- Deploy the exact verified commit or document every code change. - Transfer upgrade authority to the published multisig and timelock. - Configure oracle deviation and stale-price parameters before activation. - Enable monitoring for upgrades, pauses and abnormal asset movement. - Obtain a focused re-audit after material upgrades.

### 02 Verification

**Official Report Verification**

Scan the QR code or verify the report ID on Finguard. Only the official verification page confirms the current status, file fingerprint, scope and version.

# Disclaimer

## 01 Important Notice

This report reflects a security review of the specified source code, repository commit and assumptions available during the audit period. Security audits reduce risk but cannot prove that software is free from every defect, vulnerability or future exploit. Changes made after the reviewed commit are not covered unless separately verified. The assessment does not cover private key compromise, malicious governance decisions, third-party infrastructure failure, front-end compromise, DNS attacks, market conditions or operational negligence unless explicitly listed in scope. This report is not financial, legal or investment advice. Finguard does not guarantee token value, protocol solvency, future performance or uninterrupted availability. The project remains responsible for secure deployment, configuration, monitoring, incident response and ongoing maintenance. Only a report verified through the official Finguard verification page should be treated as authentic.

## 02 Template Notice

This PDF contains sample project data for design and workflow demonstration. It must not be represented as a completed audit of a real protocol.