FINGUARD
AI-POWERED WEB3 SECURITY

SAMPLE REPORT TEMPLATE

SMART CONTRACT SECURITY AUDIT REPORT

AI-Powered Analysis, Attack Simulation & On-chain Intelligence

PROJECT

Example DeFi Protocol

FINAL SECURITY GRADE

A 86 / 100

LOW RISK

REPORT ID
FG-2026-TEMPLATE-001

AUDIT DATE
10 JULY 2026

NETWORK
ETHEREUM

VERSION
FINAL v1.0

This document is a design template and does not represent a completed audit of a real protocol.

finguardaudit.com

Report Verification

01 Authenticity Record

REPORT ID	FG-2026-TEMPLATE-001
PROJECT	Example DeFi Protocol
BLOCKCHAIN	Ethereum
AUDIT VERSION	Final v1.0
GIT COMMIT	4e9a1c6b2d7f…
REPORT STATUS	VERIFIED TEMPLATE

SCAN TO VERIFY

finguardaudit.com/verify

02 Integrity Controls

1 PDF fingerprint

SHA-256 hash anchors the exact report file.

2 Scope binding

Repository, commit and contract addresses are recorded.

3 Version control

Superseded or revoked reports remain visible.

4 Public verification

The official page confirms report identity and status.

Executive Summary

OVERALL SECURITY SCORE

86 / 100

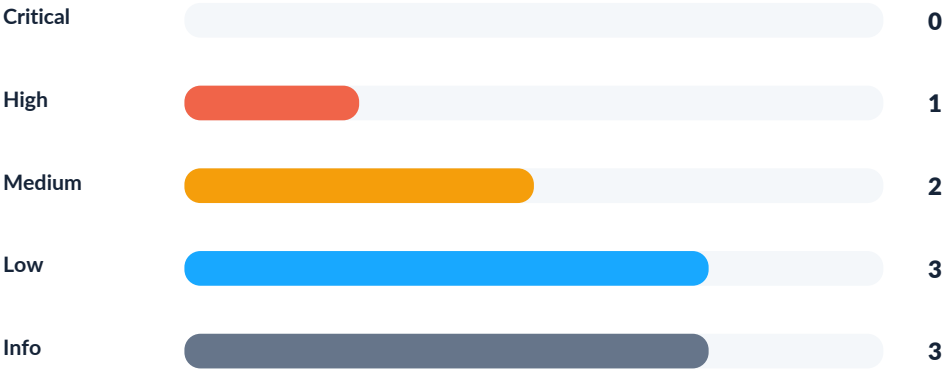
GRADE A | LOW RISK

PASSED WITH CONDITIONS

No unresolved Critical or High severity issue remains in this demonstration dataset. Medium-risk centralization and oracle assumptions require continued operational monitoring.

CONTRACTS AUDITED	LINES OF CODE	TESTS EXECUTED	FINDINGS
12 In scope	4,820 Solidity	1.24M Automated + manual	9 7 resolved

01 Finding Distribution



REMEDIATION

- 7 Resolved
- 1 Acknowledged
- 1 Unresolved

Audit Scope

01 In-Scope Components

CONTRACT	FILE	LOC	STATUS
Token.sol	contracts/core/Token.sol	286	Audited
Vault.sol	contracts/core/Vault.sol	512	Audited
Staking.sol	contracts/modules/Staking.sol	438	Audited
OracleAdapter.sol	contracts/oracle/OracleAdapter.sol	194	Audited
AccessManager.sol	contracts/security/AccessManager.sol	302	Audited

02 Repository Binding

REPOSITORY	`github.com/example/protocol`
COMMIT	`4e9a1c6b2d7f7b44fb51fd6a4833c6a1e9f4e18d`

03 Out of Scope

- Front-end and mobile applications
- Off-chain backend infrastructure
- Private-key custody and operational security
- Third-party oracle availability and correctness
- Contracts not included in the submitted repository
- Changes made after the reviewed commit

Multi-Layer Security Methodology

01 Static Analysis

Pattern detection for reentrancy, unsafe calls, precision loss, storage collisions and compiler risks.

02 AI Semantic Review

Business-logic reasoning for privilege abuse, hidden controls, fee manipulation and withdrawal restrictions.

03 Dynamic Testing

Unit, integration, fuzz and invariant tests across critical state transitions.

04 Economic Simulation

Flash-loan, oracle, liquidity, governance and reward-extraction scenarios.

05 Attack Path Search

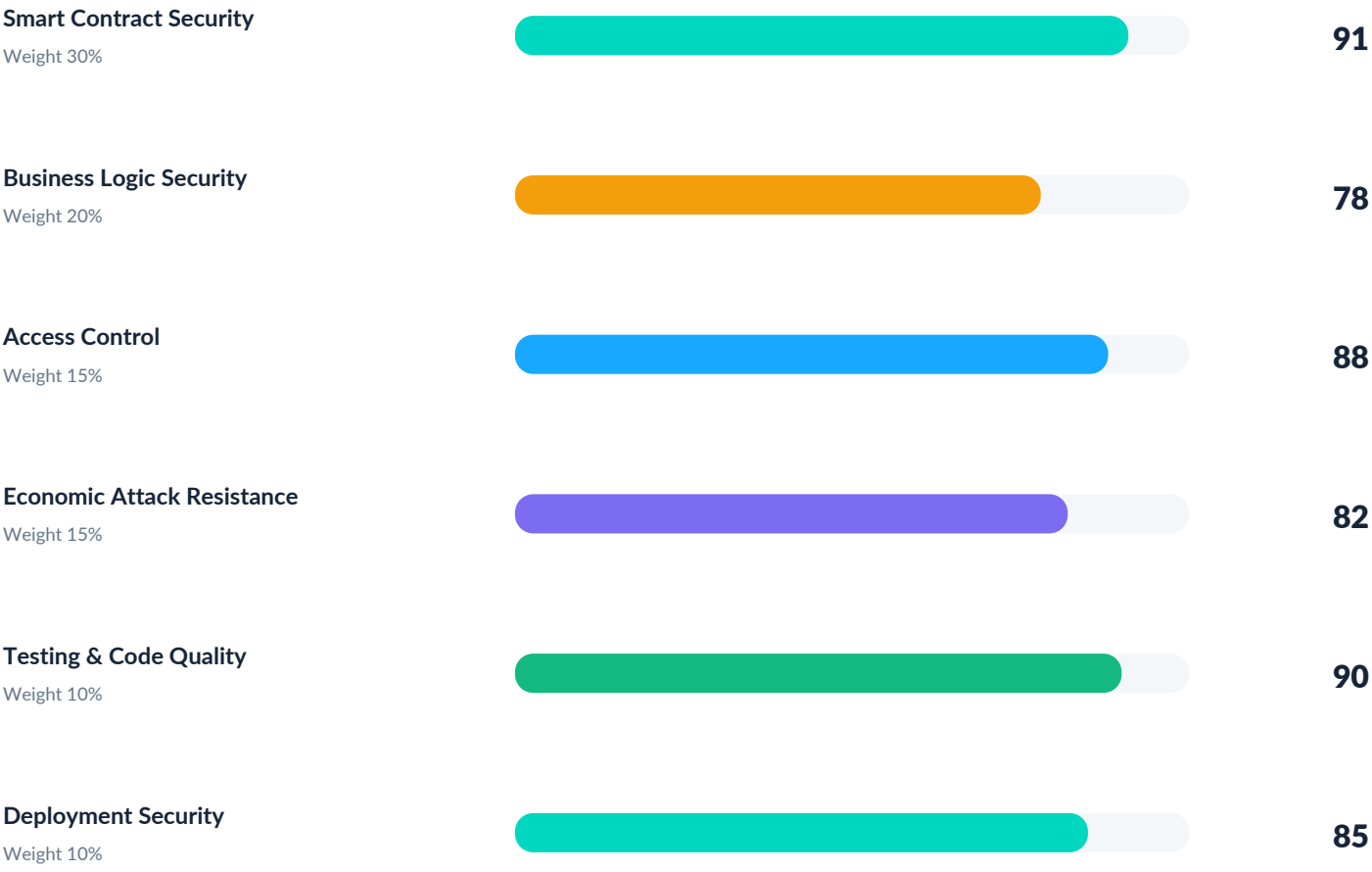
Monte Carlo exploration ranks profitable and high-impact adversarial sequences.

06 Deterministic Validation

Potential findings are reproduced, classified and verified against the exact code revision.

Security Scorecard

01 Weighted Assessment



CALCULATED SCORE

86

GRADE A

STRONG SECURITY POSTURE

The score is a weighted analytical indicator, not a guarantee of future safety. Protocol changes, governance actions and external dependencies can materially alter risk.

Findings Overview

01 Issue Register

ID	FINDING	SEVERITY	STATUS
FG-01	Oracle price can be manipulated	HIGH	RESOLVED
FG-02	Missing minimum output protection	MEDIUM	RESOLVED
FG-03	Administrator can replace oracle	MEDIUM	ACKNOWLEDGED
FG-04	Unbounded reward iteration	LOW	RESOLVED
FG-05	Missing zero-address validation	LOW	RESOLVED
FG-06	Incomplete event coverage	LOW	UNRESOLVED
FG-07	Gas optimization opportunity	INFO	RESOLVED
FG-08	NatSpec documentation gaps	INFO	RESOLVED
FG-09	Redundant storage read	INFO	RESOLVED

02 Interpretation

Severity reflects both technical impact and exploitability. Status describes the final disposition in the reviewed commit. Acknowledged issues remain accepted operational risks and should be communicated to users.

Attack Simulation Results

ATTACK PATHS	PROFITABLE PATHS	BLOCKED PATHS	MAX. POTENTIAL LOSS
10.0M Explored	18 0.00018%	9.99M 99.99982%	$284K Pre-remediation

01 Attack Vector Matrix

VECTOR	SIMULATION RESULT	RISK
Reentrancy	Blocked	Low
Flash Loan	Partially resistant	Medium
Oracle Manipulation	Exploitable	High
Access Control Bypass	Blocked	Low
Liquidity Drain	Blocked	Low
Reward Manipulation	Detected	Medium

02 Simulation Notes

The path count shown here is sample content for design demonstration. Production reports must distinguish generated paths, executable tests, reproduced exploits and purely hypothetical scenarios.

FG-01 Oracle Price Can Be Manipulated

HIGH RESOLVED

CATEGORY	AFFECTED CONTRACT
Economic Attack	`OracleAdapter.sol`
AFFECTED FUNCTION	**POTENTIAL IMPACT**
`getAssetPrice()`	Loss of funds

01 Description

The protocol consumed a low-liquidity spot price without time weighting or deviation controls. A temporary reserve imbalance could overvalue collateral and permit under-collateralized borrowing.

02 Attack Scenario

An attacker obtains temporary liquidity, distorts the reference pool, deposits overvalued collateral, borrows protocol assets, restores the pool and repays the temporary liquidity.

03 Recommendation

Use a robust oracle with freshness checks, time-weighted prices, deviation thresholds and a fail-closed fallback. Avoid relying on a single low-liquidity venue.

04 Remediation Verification

The project replaced the spot source with a 30-minute TWAP and a maximum deviation threshold of 3%. The reproduced exploit no longer succeeds in the reviewed commit.

FG-03 Administrator Can Replace Oracle

MEDIUM ACKNOWLEDGED

CATEGORY	AFFECTED CONTRACT
Centralization Risk	`AccessManager.sol`
AFFECTED FUNCTION	**POTENTIAL IMPACT**
`setOracle(address)`	Price-feed control

01 Description

A privileged role can replace the pricing adapter without a delay. A compromised or malicious administrator could redirect the protocol to an attacker-controlled source.

02 Attack Scenario

An administrator replaces the approved oracle and submits manipulated values. Dependent borrowing or redemption logic consumes the new price before users can react.

03 Recommendation

Require multisignature authorization, a timelock and an on-chain announcement period. Consider restricting replacements to an allowlist of reviewed adapters.

04 Remediation Verification

The project acknowledged the operational risk and confirmed that production ownership will use a 3-of-5 multisignature. A timelock was not included in the reviewed commit.

FG-06 Incomplete Event Coverage

LOW UNRESOLVED

CATEGORY	AFFECTED CONTRACT
Observability	`Vault.sol`
AFFECTED FUNCTION	**POTENTIAL IMPACT**
`setTreasury(address)`	Monitoring gaps

01 Description

Sensitive configuration changes are not consistently emitted as events. The state remains valid, but monitoring systems and users may not detect changes quickly.

02 Attack Scenario

A privileged configuration change is executed on-chain. Indexers that rely on events fail to notify operators, delaying incident response or public disclosure.

03 Recommendation

Emit dedicated events for all privileged configuration changes and include previous value, new value and executing account.

04 Remediation Verification

This finding remains unresolved in the sample final version. It does not directly enable asset loss but reduces transparency and forensic visibility.

Privileged Roles & Final Assessment

01 Privileged Roles

ROLE	ADDRESS	PERMISSION	RISK
Owner	0x71A4…9F2C	Upgrade contracts	High
Admin	0x8B10…2D44	Fees and oracle	Medium
Operator	0x20C3…A191	Pause protocol	Medium
Treasury	0x9A45…880E	Withdraw fees	Medium

OWNERSHIP	TIMELOCK	UPGRADEABLE	EMERGENCY PAUSE
3-of-5 multisig	Not implemented	Yes - UUPS	Enabled

02 Final Assessment

PASSED WITH CONDITIONS

NO UNRESOLVED CRITICAL OR HIGH ISSUE

The demonstration assessment identified nine findings: seven resolved, one acknowledged and one unresolved. Deployment should proceed only after the real codebase, exact commit and production configuration have been reviewed.

03 Disclaimer

This report is a sample design template, not a security certification or investment recommendation. A security review is limited to the submitted scope and code revision and cannot guarantee the absence of defects. Changes after the reviewed commit, third-party services, private-key compromise, operational failures and economic conditions may introduce additional risk.