HORUS Protocol Security Audit Report & Remediation

Protocol: HORUS Prediction Market Protocol **Blockchain**: Hedera Hashgraph (EVM-compatible) **Initial Audit Date**: February 7, 2026 **Remediation Completed**: February 8, 2026 **Solidity Version**: 0.8.20 **Scope**: All deployed and deployable contracts (~3,100 lines of production Solidity) **Post-Remediation Test Suite**: 486 tests passing, 95.24% line coverage

---

Executive Summary

The HORUS Protocol underwent a comprehensive security audit on February 7, 2026, conducted across 5 parallel audit workstreams covering Critical/High vulnerabilities, Medium/Low severity issues, Hedera-specific risks, Test Coverage, and Architecture review.

The initial audit identified **42 findings**: 7 Critical, 11 High, 10 Medium, 6 Low, and 8 Informational. All 7 Critical and all 11 High severity findings have been successfully remediated. The protocol has been redeployed to Hedera testnet (v2.0) with hardened contracts.

Post-Remediation Security Posture

Category	Before	After
Security	D+ (50/100)	A- (90/100)
Decentralization	D (35/100)	B (75/100)
Test Coverage	C+ (65/100)	A (95/100)
Feature Completeness	C (60/100)	B+ (85/100)

Remediation Statistics

Severity	Found	Fixed	Status
Critical	7	7	ALL RESOLVED
High	11	11	ALL RESOLVED
Medium	10	8	8 RESOLVED, 2 ACCEPTED
Low	6	4	4 RESOLVED, 2 DEFERRED
Informational	8	6	6 RESOLVED, 2 DEFERRED

Scope & Methodology

Contracts Audited

Contract	Path	Lines	Role
MasterOracle.sol	contracts/core/	~750	Central registry, staking, governance, timelock
OracleInstance.sol	contracts/core/	~507	Individual prediction market logic
FeeDistributor.sol	contracts/core/	~530	Revenue distribution, epoch system, timelock
VetoGuard.sol	contracts/core/	~260	DAO voting with minimum quorum

OracleInstanceFactory.sol	contracts/factories/	~200	EIP-1167 minimal proxy factory with CREATE2
VetoGuardFactory.sol	contracts/factories/	~120	VetoGuard deployment factory
BondCalculator.sol	contracts/libraries/	~104	Bond escalation math
HorusErrors.sol	contracts/utils/	~208	Custom error definitions

Audit Methodology

1. **Manual Code Review**: Line-by-line analysis of all contracts by 5 parallel auditors
2. **Architecture Analysis**: Contract dependency graph, trust boundaries, upgrade paths
3. **Game Theory Analysis**: Bond escalation economics, staking incentives, governance voting
4. **Hedera-Specific Review**: HTS integration, precompile usage, int64 constraints
5. **Test Coverage Audit**: Gap analysis of all test suites
6. **Deployment Script Review**: Deployment scripts analyzed for correctness
7. **Industry Benchmarking**: Comparison against UMA, Augur, and Polymarket

---

Critical Severity Findings (ALL RESOLVED)

CRIT-01: Deposit Inflation Without Token Verification

Contract: MasterOracle.sol | **CVSS**: 9.8

`depositForMarketCreation()` recorded deposit amounts without verifying token transfer. Attackers could create phantom deposits and drain the contract.

Fix (SEC-01, Phase 07): Implemented pull pattern with balance-before/after verification. The contract now calls `transferFrom` internally and records only the actual received amount (balance delta), preventing phantom deposits and fee-on-transfer token exploits.

Verification: 7 unit tests confirming the vulnerability is closed.

CRIT-02: Broken Authorization Chain in VetoGuard.finalize()

Contract: VetoGuard.sol | **CVSS**: 9.3

`VetoGuard.finalize()` called `OracleInstance.receiveDaoResult()` directly, but that function requires `msg.sender == masterOracle`. The call always reverted, making DAO dispute resolution non-functional.

Fix (SEC-05, Phase 08): Introduced `MasterOracle.relayDaoResult()` as an intermediary. VetoGuard now calls MasterOracle, which relays the DAO result to OracleInstance with proper authorization.

Verification: Full DAO lifecycle integration test covering escalation, VetoGuard deployment, voting, finalization, and resolution.

CRIT-03: Unauthorized setResolver Allows Resolver Theft

Contract: FeeDistributor.sol | **CVSS**: 9.0

`setResolver()` had no caller verification. Anyone could overwrite the resolver address and steal 500 HORUS per market.

Fix (SEC-03, Phase 07): Added `require(msg.sender == instance)` check with only-once semantics. The resolver can only be set by the OracleInstance contract itself.

Verification: 8 unit tests proving unauthorized calls revert.

CRIT-04: Bond Overwrite Vulnerability

Contract: OracleInstance.sol | **CVSS**: 9.1

`_recordBond()` overwrote the entire `BondInfo` struct on subsequent escalation rounds, losing previously bonded amounts permanently.

Fix (SEC-02, Phase 07): Changed to field-level accumulation (`bonds[user].amount += amount`) with same-side enforcement guard. Role and timestamp track the latest action; only amount accumulates.

Verification: 5 unit tests including multi-round escalation scenarios.

CRIT-05: DAO Vote Logic Not Implemented

Contract: MasterOracle.sol | **CVSS**: 8.8

`_castVote()` validated voter stake but never forwarded votes to the VetoGuard contract.

Fix (SEC-06, Phase 08): `_castVote()` now calls `IVetoGuard(vetoGuard).castVeto()` or `counterVeto()` with the voter's stake amount read from on-chain state. VetoGuard's `castVeto` / `counterVeto` restricted to MasterOracle only.

Verification: Integration tests confirm votes are properly recorded and counted.

CRIT-06: Integer Overflow in HTS Transfer

Contracts: OracleInstance.sol, FeeDistributor.sol | **CVSS**: 8.5

`_transferHorus()` cast `uint256` to `int64` without bounds checking. Values exceeding `int64` max could reverse transfer direction or silently truncate.

Fix (SEC-04, Phase 07): Added overflow guard `require(amount <= uint256(uint64(type(int64).max)))` before every HTS transfer call across all contracts.

Verification: Unit tests with boundary values at `int64` max.

CRIT-07: resolveDaoVote() Deletes State Without Finalizing

Contract: MasterOracle.sol | **CVSS**: 8.7

`resolveDaoVote()` deleted the VetoGuard mapping without calling `finalize()` , orphaning the VetoGuard and leaving markets stuck in VOTING state.

Fix (SEC-07, Phase 08): `resolveDaoVote()` now calls `VetoGuard.finalize()` before clearing state, which relays the result through `MasterOracle.relayDaoResult()` to the OracleInstance.

Verification: Full lifecycle test verifying state transitions from VOTING to RESOLVED.

High Severity Findings (ALL RESOLVED)

HIGH-01: Cross-VetoGuard Double Voting (SEC-12)

Fix: Added `hasVotedInGuard` mapping in MasterOracle tracking voter participation per VetoGuard. Prevents stake reuse across concurrent disputes.

HIGH-02: OracleInstance.initialize() No Caller Access (SEC-13)

Fix: `initialize()` restricted to factory/MasterOracle caller. Implementation template locked in constructor with dummy initialization.

HIGH-03: Public Token Association (SEC-14)

Fix: Factory now calls `associateWithToken()` on every clone after initialization. All clones are HTS-associated at deployment time.

HIGH-04: Current Stake Used for Historical Rewards (SEC-11)

Fix: Implemented OpenZeppelin Checkpoints-based stake snapshots. Rewards calculated using historical stake at each epoch, preventing stake inflation attacks.

HIGH-05: Arbitrary totalStaked in advanceEpoch (SEC-08)

Fix: `advanceEpoch()` now reads `totalStaked` directly from MasterOracle via cross-contract call. External parameter removed entirely.

HIGH-06: No Parameter Bounds (SEC-18)

Fix: Enforced bounds: min bond >= 100 HORUS, challenge period 4h-7d, veto threshold >= 1%, max escalations <= 5.

HIGH-07: Gas DoS via Unbounded Loop (SEC-09)

Fix: Added `instanceToId` reverse mapping (O(1) lookup). The O(n) `_findInstanceId()` loop is eliminated.

HIGH-08: Inconsistent Token Transfers (SEC-10)

Fix: Unified all contracts to use HTS precompile pattern for outbound HORUS transfers.

HIGH-09: Admin Omnipotence Without Timelock (SEC-15)

Fix: 48-hour timelock on all admin parameter changes via propose/execute pattern with hash-based change tracking.

HIGH-10: Guardian Indefinite Pause (SEC-16)

Fix: Emergency withdrawal available after 7 days of continuous pause. Auto-unpause after 30 days prevents permanent fund lock.

HIGH-11: No Admin Transfer (SEC-17)

Fix: Two-step admin transfer (proposeAdmin/acceptAdmin) on both MasterOracle and FeeDistributor.

Medium Severity Findings

ID	Title	Status	Resolution
MED-01	EIP-1167 / HTS Association Gap	RESOLVED (SEC-14)	Factory calls associateWithToken()
MED-02	CREATE2 vs CREATE Mismatch	RESOLVED (SEC-20)	Deterministic CREATE2 clones with salt
MED-03	Market State Never Updates	ACCEPTED	By design -- OracleInstance is source of truth

MED-04	uint128 Truncation	RESOLVED	Covered by int64 overflow guard
MED-05	Delegation Issues	ACCEPTED	Delegation feature deferred to v2
MED-06	Fee Rounding Errors	RESOLVED (SEC-21)	Remainder assigned to last recipient
MED-07	Unbounded bondHolders	RESOLVED	Bounded by maxEscalations (max 5)
MED-08	No Minimum Quorum	RESOLVED (SEC-22)	5% minimum quorum enforced
MED-09	Epoch Depends on Markets	RESOLVED (SEC-23)	Permissionless epoch advancement
MED-10	No Upgrade Path	ACCEPTED	Immutable by design for security

---

Dead Code Cleanup (SEC-19)

651 lines of dead code removed:

- **Administrable.sol** (208 lines) -- features integrated into MasterOracle/FeeDistributor directly
- **HorusTokenIntegration.sol** (217 lines) -- int64 overflow protection now in all production contracts
- **EpochManager.sol** (226 lines) -- epoch logic consolidated in FeeDistributor

---

Test Coverage

Post-Remediation Test Suite

Metric	Value
Total Tests	486
Passing	486 (100%)
Line Coverage	95.24%
Branch Coverage	82.57%
Security Test Suites	8 dedicated attack simulation suites

Security Test Categories

Category	Tests	Coverage
Reentrancy Attack Simulation	10	All nonReentrant functions tested
Flash Loan Resistance	7	Stake-vote-unstake same-block blocked
Gas DoS Resistance	8	1000+ markets, 50+ epochs under gas limits
Protocol Invariants (INV-1 to INV-8)	20	totalStaked, totalBonds, fee splits verified
DAO Lifecycle Edge Cases	10	Quorum failure, tied votes, late finalization
Economic Security	19	Parameter manipulation, stake inflation blocked

Admin Hardening	35	Timelock bypass, emergency withdrawal timing
Coverage Gap Tests	46	View functions, error paths, boundary conditions

---

Deployment Information

Testnet Deployment (v2.0)

Contract	Hedera ID	Status
HorusToken	0.0.7819788	Pre-existing
FeeDistributor	0.0.7864586	Deployed Feb 8, 2026
VetoGuardFactory	0.0.7864590	Deployed Feb 8, 2026
OracleInstanceFactory	0.0.7864594	Deployed Feb 8, 2026
MasterOracle	0.0.7864600	Deployed Feb 8, 2026

Post-Deployment Verification

- Cross-reference validation: 15/15 PASS
- Parameter validation: 5/5 PASS
- All contracts correctly wired together
- Nonce-precomputed MasterOracle address matched actual deployment

---

Conclusion

All 7 Critical and 11 High severity findings from the initial audit have been successfully remediated. The protocol now includes:

- **48-hour timelock** on all admin parameter changes
- **Emergency withdrawal** after 7 days of continuous pause
- **Auto-unpause** after 30 days to prevent permanent fund lock
- **Two-step admin transfer** for key rotation
- **Historical stake snapshots** for accurate reward calculation
- **Minimum quorum** (5%) for DAO governance votes
- **Deterministic CREATE2 clones** for predictable market addresses
- **Unified HTS transfer pattern** across all contracts
- **486 passing tests** with 95.24% line coverage

The security-hardened contracts have been deployed to Hedera testnet and verified via automated cross-reference and parameter validation.

This report documents the initial audit findings and their remediation. The HORUS Protocol team is committed to ongoing security improvements and welcomes responsible disclosure of any additional vulnerabilities.

Report generated: February 8, 2026