OpenZeppelin-Style Security Audit

Ganjes DAO Smart Contract

Audit Date:	**August 20, 2025**
Contract:	**GanjesDAOOptimized.sol**
Solidity Version:	**^0.8.20**
Methodology:	**OpenZeppelin Standards**
Security Score:	**6.5/10**

■ EXECUTIVE SUMMARY

- DAO contract with proposal-based governance and investment voting
- Multiple critical and high-severity security vulnerabilities identified
- Contract demonstrates good security practices but has fundamental flaws
- **RECOMMENDATION: DO NOT DEPLOY** until critical issues are resolved
- Professional audit recommended before mainnet deployment

■ CONTRACT OVERVIEW

Component	Details
Main Contract	GanjesDAOOptimized
Inheritance	ReentrancyGuard, Pausable
Libraries	SafeERC20 (custom), Address
Token Standard	ERC-20 governance token
Core Functions	Proposal creation, voting, execution, refunds

■ CRITICAL SECURITY ISSUES

C1: Inconsistent Vote Counting Logic

Severity: CRITICAL | **Location:** Lines 451-469

- **Issue:** Vote weights based on voter balance but votes can be changed with increased investments
- **Impact:** Vote manipulation, incorrect outcomes, potential fund loss
- **Root Cause:** Mixed voting mechanisms (balance + investment)
- **Recommendation:** Use consistent weighting (investment-only or snapshot voting)

C2: Missing SafeERC20 Dependencies

Severity: CRITICAL | **Location:** Line 78

- **Issue:** SafeERC20 calls functionCall() method that doesn't exist
- **Impact:** Runtime failures on all token transfers, contract unusable
- **Root Cause:** Incomplete Address library implementation
- **Recommendation:** Import OpenZeppelin's complete SafeERC20 library

■■ HIGH SEVERITY ISSUES

H1: Potential Integer Overflow

Severity: HIGH | **Location:** Line 445

- Vote counting may accumulate to very large numbers
- Potential DoS through overflow reverts
- Recommendation: Implement reasonable upper bounds

H2: Admin Privilege Escalation Risk

Severity: HIGH | **Location:** Lines 889-917

- Up to 10 admins can be added without sufficient safeguards
- Risk of governance takeover and unauthorized withdrawals
- Recommendation: Multi-signature requirements for admin operations

H3: Emergency Withdrawal Miscalculation

Severity: HIGH | **Location:** Lines 943-945

- Emergency withdrawal limit doesn't exclude committed proposal funds
- Risk of withdrawing funds needed for approved proposals
- Recommendation: Calculate limit based on free funds only

■ MEDIUM & LOW SEVERITY ISSUES

ID	Issue	Severity	Impact
M1	Proposal Spam Prevention Insufficient	Medium	DoS attacks, increased gas costs
M2	Vote Changing Logic Complexity	Medium	State inconsistencies, gas exploitation
M3	No Automatic Proposal Extensions	Medium	Loss of viable proposals
L1	Event Parameter Indexing	Low	Suboptimal dApp integration
L2	String Comparison Optimization	Low	Minor gas optimization
L3	Unused Error Definition	Low	Code cleanliness

■ OPENZEPPELIN STANDARDS COMPLIANCE

Standard	Status	Notes
ReentrancyGuard	■ PASS	Properly implemented with CEI pattern
Pausable	■ PASS	Correct implementation with admin controls
SafeERC20	■ FAIL	Custom implementation with missing dependencies
Custom Errors	■ PASS	Proper use for gas efficiency
Events	■ PASS	Comprehensive event logging
Access Control	■■ PARTIAL	Role-based but lacks multi-sig
Upgradability	■ FAIL	No upgrade mechanism implemented
Timelock	■ FAIL	No timelock for administrative changes

■ PRIORITY RECOMMENDATIONS

1. IMMEDIATE (Critical):

- Fix vote counting logic - use consistent weighting mechanism
- Resolve SafeERC20 dependency issues

2. BEFORE DEPLOYMENT (High Priority):

- Implement multi-signature for admin operations
- Fix emergency withdrawal calculation
- Add proper bounds checking for vote counting

3. SECURITY ENHANCEMENTS:

- Import OpenZeppelin's AccessControl system
- Add timelock controller for admin functions
- Implement snapshot voting mechanism
- Add circuit breakers for emergency scenarios

4. TESTING REQUIREMENTS:

- Comprehensive vote counting accuracy tests
- Reentrancy attack simulation
- Emergency scenario testing
- Multi-user concurrent operation testing

■ FINAL ASSESSMENT

Metric	Score	Comments
Security	6.5/10	Good practices but critical flaws
Code Quality	8/10	Well-structured with good documentation
Gas Efficiency	7/10	Reasonable optimizations implemented
Upgradability	3/10	No upgrade mechanism
Admin Security	4/10	Basic access control, needs multi-sig
Overall Risk	HIGH	Critical issues must be resolved

■■ PRE-DEPLOYMENT CHECKLIST

■ Fix critical vote counting logic flaw
■ Resolve SafeERC20 dependency issues
■ Implement multi-signature admin controls
■ Add proper emergency withdrawal calculations
■ Complete comprehensive testing suite
■ Professional security audit by certified auditors
■ Deploy to testnet for extensive testing
■ Set up monitoring and alerting systems
■ Prepare emergency response procedures
■ Verify all contract parameters and constants

CONCLUSION

This contract shows good security awareness but contains critical vulnerabilities that make it unsafe for deployment. The vote counting mechanism and SafeERC20 implementation issues are particularly concerning and must be resolved immediately. We strongly recommend engaging professional auditors before mainnet deployment.

This audit follows OpenZeppelin's security review framework. Report generated on August 20, 2025 for educational and security assessment purposes.