OpenZeppelin Security Audit Report

Ganjes DAO Smart Contracts

Audit Overview

Project: Ganjes DAO Smart Contracts

Date: August 20, 2025

Tools: Slither Static Analysis, OpenZeppelin Defender SDK

Contracts: 4 smart contracts analyzed

Total Findings: 80 security issues

Risk Level: HIGH (Critical vulnerabilities present)

Recommendation: Address critical issues before deployment

Risk Summary

Severity	Count	Priority
Critical	2	Immediate Fix Required
High	8	Fix Before Deployment
Medium	15	Fix in Next Version
Low/Info	55+	Optimization & Best Practices

■ CRITICAL FINDINGS (2)

RE-1: Reentrancy Vulnerabilities in Proposal Creation

• Location: GanjesDAOSimplified.sol:95-138, ProposalManagement.sol:135-228
• Issue: State variables written after external token transfers
• Impact: Attackers can manipulate proposal limits and bypass cooldowns
• Fix: Implement ReentrancyGuard or Checks-Effects-Interactions pattern

RE-2: Reentrancy in Voting Functions

• Location: GanjesDAOSimplified.sol:140-171
• Issue: Multiple state updates after external calls in vote() function
• Impact: Vote manipulation and potential double-spending attacks
• Fix: Apply reentrancy protection and reorder operations

■ HIGH RISK FINDINGS (8)

AC-1: Missing Access Control on Critical Functions

• Multiple administrative functions lack proper access control
• Fix: Implement OpenZeppelin's AccessControl or Ownable

TX-1: Transaction Order Dependence (MEV Vulnerability)

• Functions vulnerable to front-running attacks
• Fix: Implement commit-reveal schemes or timestamp-based ordering

COMP-1: Stack Too Deep Compilation Error

• GanjesDAOOptimized.sol fails to compile
• Fix: Enable --via-ir flag or reduce local variables

■ MEDIUM RISK FINDINGS (15)

EQ-1: Dangerous Strict Equality Check

• Using == for timestamp comparison in ProposalManagement.sol:418

• Fix: Use <= or range checks instead of strict equality

US-1: Unused State Variables

• Multiple unused variables increase gas costs

• Fix: Remove unused variables or mark as private

UF-1: Unused Functions (15+ functions)

• Dead code bloats contract size

• Fix: Remove unused functions or document if needed for future

■ LOW RISK & INFORMATIONAL (55+)

NC-1: Naming Convention Violations (25+ parameters)

• Parameters not following mixedCase convention

• Example: _projectName → projectName

GS-1: Gas Optimization Opportunities

• Variables that should be constant or immutable:

• SimpleToken.decimals, name, symbol → constant

• GanjesDAO.admin, votingDuration → immutable

LD-1: Large Number Literals

• Use scientific notation: 1000000 * 10**18 → 1e6 * 1e18

PRIORITIZED REMEDIATION PLAN

Phase 1: Immediate (Critical/High Risk)

1. Implement ReentrancyGuard on all external calls
2. Add proper access control to administrative functions
3. Fix compilation issues in GanjesDAOOptimized.sol
4. Secure all token transfer operations
5. Review and test all state-changing functions

Phase 2: Short-term (Medium Risk)

1. Replace strict equality checks with range checks
2. Remove unused state variables and functions
3. Implement comprehensive input validation
4. Add proper error handling throughout contracts

Phase 3: Long-term (Optimization)

1. Fix naming convention violations
2. Declare appropriate variables as constant/immutable
3. Optimize gas usage patterns
4. Improve code documentation and comments

OPENZEPPELIN INTEGRATION RECOMMENDATIONS

Security Modules to Import:

• @openzeppelin/contracts/security/ReentrancyGuard.sol
• @openzeppelin/contracts/security/Pausable.sol
• @openzeppelin/contracts/access/Ownable.sol
• @openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol

Consider Migration to OpenZeppelin Governor:

• Standardized DAO governance framework
• Battle-tested security implementations
• Community-reviewed codebase