NERO: Security Audit Report

DogScan Security Team

August 4th, 2025

Contents

DogScan Security Audit Report

Project	NERO
Contract File	`StakingAnytimeForPeriodNoProxy_v8.3.sol`
Audit Date	August 4th, 2025
Report Version	1.0

1. Executive Summary

We conducted a comprehensive security audit of the `StakingAnytimeForPeriod` contract. This contract implements a flexible staking mechanism that allows users to deposit and withdraw tokens at any time while earning APY-based rewards. The audit results show a generally sound overall structure, but with one medium-severity issue that requires attention.

The identified main vulnerability relates to interest calculation inconsistencies where users may receive different rewards than expected based on changing APY and period values.

The audit results revealed **one medium-severity issue** primarily related to reward calculation logic.

Overall Risk Rating: Medium

Based on the project team's clarification that parameters will not be modified, the actual risk of this issue has been significantly reduced.

2. Audit Scope

The audit scope covers the `StakingAnytimeForPeriod` contract and its related components:

Contract Information:

- Contract Type: Staking Reward Contract
- Main Functions: Token staking and withdrawal, APY reward calculation

Key Audit Areas:

- Staking and withdrawal mechanisms
- APY reward calculation logic

- Access control patterns
- Parameter setting and management functions
- Token handling and security mechanisms

3. Audit Methodology

This audit employed a multi-agent AI security analysis framework specifically designed for smart contract security assessment:

1. Specialized Analysis Modules:

- **Reward Calculation Expert**: Reviews APY calculation and interest distribution logic
- **Access Control Analyst**: Evaluates permission management and administrator functions
- **Token Security Expert**: Examines token transfers and balance management
- **Staking Mechanism Expert**: Analyzes deposit/withdrawal processes and time locks
- **Code Quality Expert**: Evaluates code standards and maintainability

2. Comprehensive Analysis:

- Static code analysis focused on reward calculation logic
- Automated static analysis tools to identify common vulnerability patterns
- Manual code review to discover context-specific design flaws
- Economic model analysis to ensure reasonable reward mechanisms

4. Findings Summary

ID	Title	Severity	Status
M-01	Interest Calculation Uses Current APY and Period Instead of Deposit-Time Values	**Medium**	**Known**

5. Detailed Findings

[M-01] Interest Calculation Uses Current APY and Period Instead of Deposit-Time Values

Severity: Medium

Description When users withdraw their stakes, the interest calculation uses the current `apy_` and `period_` values instead of the values that were active when they initially deposited. The contract stores only the deposit timestamp and amount in the `StakeInfo` struct, but does not preserve the APY and period values that were active at the time of deposit.

Project Team Clarification The project team confirmed that APY and period parameters are determined at contract initialization and will not be modified subsequently. Based on this clarification, the actual impact of this issue is relatively minor, as parameter stability reduces the risk of calculation inconsistencies.

Technical Details

```
// In _earnedAmount function (Line 1773)
function _earnedAmount(uint256 _amount, uint256 _timestamp) internal view returns (uint256) {
    // Uses current apy_ and period_ values instead of deposit-time values
    uint256 reward = _amount * apy_ * (block.timestamp - _timestamp) / (period_ * 10000);
    return reward;
}

// Deposit function only stores basic information (Lines 1654 and 1615)
struct StakeInfo {
    uint256 amount;
    uint256 timestamp;
    // Missing: uint256 apy;
    // Missing: uint256 period;
}
```

Impact

- Users may receive significantly different interest amounts than they expected at the time of deposit
- If the APY decreases or the period changes unfavorably, users receive less than anticipated
- Conversely, if conditions improve, they may receive more
- This unpredictability undermines user trust and makes it impossible for users to accurately calculate their expected returns

Recommendation Modify the `StakeInfo` struct to include the APY and period values that were active at the time of deposit:

```solidity
struct StakeInfo {
    uint256 amount;
    uint256 timestamp;
    uint256 apy;        // Add: APY at deposit time
    uint256 period;     // Add: Period at deposit time
}

function _earnedAmount(uint256 _amount, uint256 _timestamp, uint256 _apy, uint256 _period)
    internal pure returns (uint256) {
    uint256 reward = _amount * _apy * (block.timestamp - _timestamp) / (_period * 10000);
    return reward;
}
```

6. Architecture and Design Assessment

Design Strengths

1. **Clear and Simple Architecture**: Contract structure is simple and easy to understand, with clearly separated main functional modules
2. **Flexible Staking Mechanism**: Supports anytime deposits and withdrawals, providing users with good fund liquidity
3. **Configurable Parameters**: APY and period can be adjusted by administrators, providing operational flexibility
4. **Standardized Implementation**: Follows basic smart contract development patterns

Key Architectural Issues

1. **Inconsistent Reward Calculation**: Uses current parameters instead of deposit-time parameters for reward calculation

Systemic Risk Assessment

This contract serves as a staking reward system with the following characteristics:

1. **Reward Calculation Risk**: Dynamic parameter changes may cause user returns to differ from expectations

2. **Administrator Centralization**: Key parameters are completely controlled by administrators
3. **User Fund Security**: Basic deposit/withdrawal mechanisms are relatively secure, but lack some boundary checks

7. Conclusion

This security audit identified **one medium-severity issue** in the `StakingAnytimeForPeriod` contract.

Key Findings Summary:

- **Reward Calculation Defects**: Uses current parameters instead of deposit-time parameters for interest calculation

Overall Risk Rating: Medium

While this contract has no direct security risks, the inconsistency in reward calculation significantly impacts user experience and trust. By making reward calculations unpredictable, it undermines user confidence in the platform.

Priority Remediation Recommendations:

1. **High Priority**: [M-01] Fix reward calculation logic to ensure use of deposit-time parameters

Based on the project team's clarification that APY and period parameters will remain stable, the contract provides users with a predictable staking experience under the current configuration. We recommend the project team consider implementing the suggested improvements if parameter adjustments become necessary in the future.

Disclaimer

This audit report is provided for informational purposes only and does not constitute investment advice. The analysis is based on smart contract source code provided at a specific point in time and does not constitute an endorsement of the project. Smart contracts carry inherent risks, and users should exercise caution and conduct their own due diligence. The findings in this report are based on automated analysis and manual review, and while extensive, they cannot guarantee the complete absence of vulnerabilities.