PUBLIC ACCESS

CYBERSECURITY AUDIT REPORT

Version v1.2

This document details the process and results of the smart contract audit performed independently by Electropact from15/09/2025 to 20/09/2025.

Audited for

NEOV (NEOVault)

Audited by

Team Electropact

https://electropact.live
info@electropact.live

© 2025 Electro-Pact. All rights reserved.
Portions of this document and the templates used in its production are the property of Electro-Pact and cannot be copied (in fullor in part) without Electro-Pact's permission.
While precautions have been taken in the preparation of this document, Electro-Pact the publisher, and the author(s) assume noresponsibility for errors, omissions, or for damages resulting from the use of the information contained herein. Use of Electro-Pact's services does not guarantee the security of a system, or that computer intrusions will not occur.

Contents

1 Introduction **4**
1.1 Audit Details 4
1.2 Audit Goals 6
1.3 Audit Methodology 6
1.4 Audit Scope 8

2 Executive Summary **9**

3 Detailed Results **12**

4 Conclusion **20**

5 Appendices **21**
Appendix A – Security Issue Status Definitions 21
Appendix B – Severity Explanation 22
Appendix C – Smart Contract Weakness Classification Registry (SWC Registry)..... 23
Appendix D – Related Common Weakness Enumeration (CWE) 28

© 2025 Electro-Pact. All rights

Disclaimer

Smart Contract Audit only provides findings and recommendations for an exact commitment of a smart contract codebase. The results, hence, is not guaranteed to be accurate outside of the commitment, or after any changes or modifications made to the codebase. The evaluation result does not guarantee the nonexistence of any further findings of security issues.

Time-limited engagements do not allow for a comprehensive evaluation of all security controls, so this audit does not give any warranties on finding all possible security issues of the given smart contract(s). Electro-Pact prioritized the assessment to identify the weakest security controls an attacker would exploit.We recommend other token conducting similar assessments on an annual basis by internal, third-party assessors, or a public bug bounty program to ensure the security of smart contract(s).

This security audit should never be used as an investment advice.

Version History

Version	Date	Release notes
1.0	16/09/2025	The first report was sent to the client. All findings were in the open status.
1.1	18/09/2025	All findings are accepted and resolved in the new GitHub commit.
1.2	20/09/2025	<NEOV> 0xb8bD708d0E9A5d811b2e57dC3660769db2AB0A95 allowed Electro-Pact to publish the auditreport publicly.

© 2025 Electro-Pact. All rights

Auditors

Fullname	Role	Email address
Nikolai	Head of Security	info@electropact.live nikolai@electropact.live

© 2025 Electro-Pact. All rights

Introduction

From 15/09/2025 to 20/09/2025, Electro-Pact to evaluate the security postureof contract system. Our findings and recommendations are detailed here in this initial report.

The report will be continually updated to correctly reflect the mitigation and remediation state of each finding.

1.1 Audit Details

Audit Target

The NEOV connectedwith blockchain technology and based on BEP20 Token. The total supply of 99,000,000,000tokens Each Token is based on Tron based Reputation System, and having following information for public.

According to the roadmap of NEOV , in August 2025, staking functions for token will be released. To ensure the safety of every customers' assets, NEOV have requested asecurity assessment on the related file NeoVaultToken.sol.

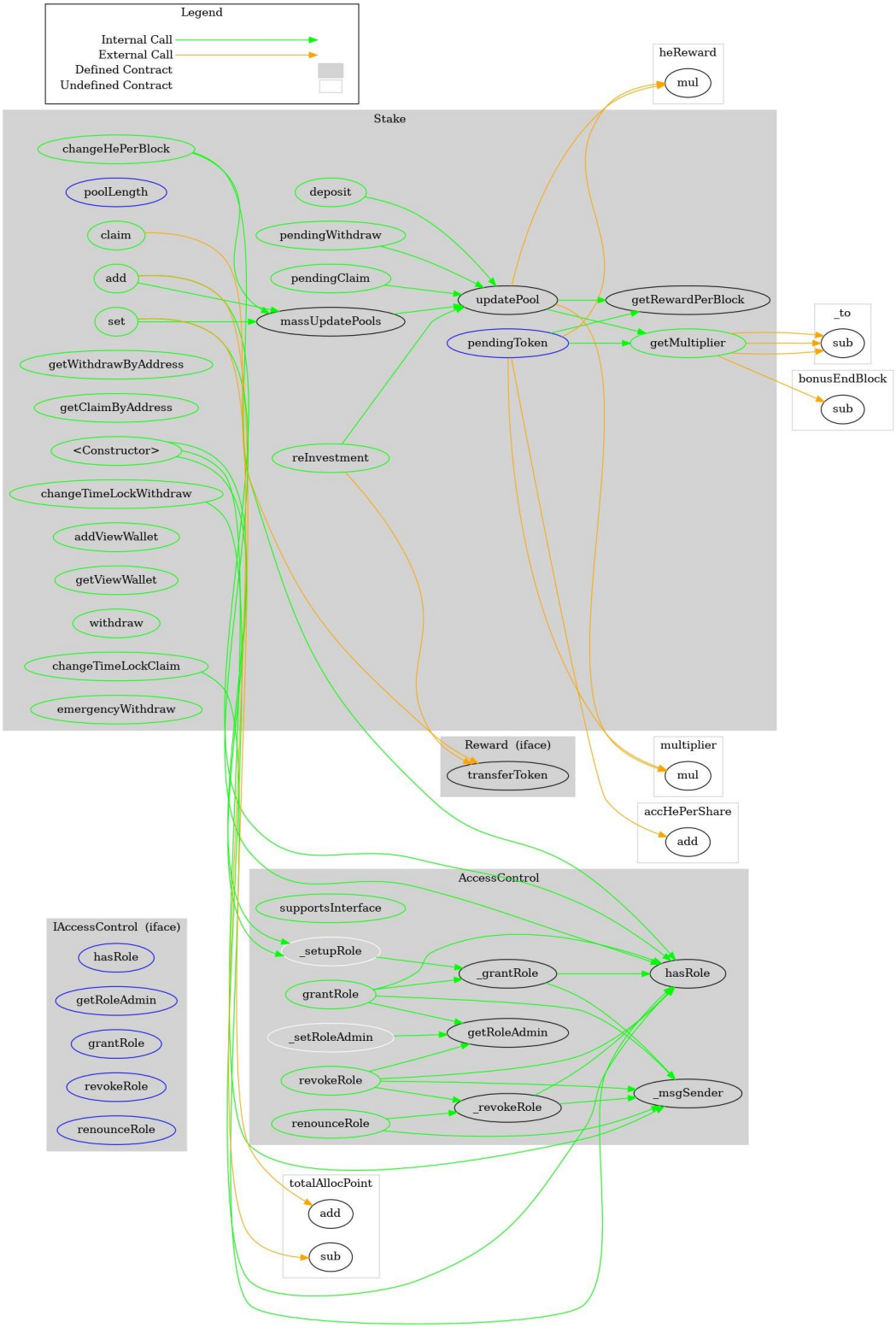
The basic information of this file is as follows:

Item	Description
Project Name	NeoVaultToken.sol (NEOV)
Issuer	https://bscscan.com/token/0xb8bD708d0E9A5d811b2e57dC3660769db2AB0A95#code
Website	https://neovault.llc/ / https://neovault.llc/
Platform	Smart Contract
Language	Solidity
Commit	0xb8bD708d0E9A5d811b2e57dC3660769db2AB0A95
Audit method	Whitebox

© 2025 Electro-Pact. All rights

With the contract NeoVaultToken.sol, users can receive rewards after certain amount of time, corresponding to the HE staking policy, by supplying into the pools a proper quantity of HE tokens. Users can claim all the rewards, withdraw their supply from the staking pools and make reinvestments to these pools anytime. The pools can only be initialized by users with the administration role (DEFAULT_ADMIN_ROLE).

The architecture of NeoVaultToken.sol is illustrated in the following graph:



© 2025 Electro-Pact. All rights

Audit Service Provider

Electro-Pact is a leading security company in USA with the goal of building the next generation of cybersecurity solutions to protect businesses against threats from the Internet.

1.2 Audit Goals

The focus of the audit was to verify that the smart contract system is secure, resilient and working according to its specifications. The audit activities can be grouped in the following three categories:

1. **Security:** Identifying security related issues within each contract and within the system of contracts.
2. **Sound Architecture:** Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.
3. **Code Correctness and Quality:** A full review of the contract source code. The primary areas of focus include:
    - Correctness
    - Readability
    - Sections of code with high complexity
    - Improving scalability
    - Quantity and quality of test coverage

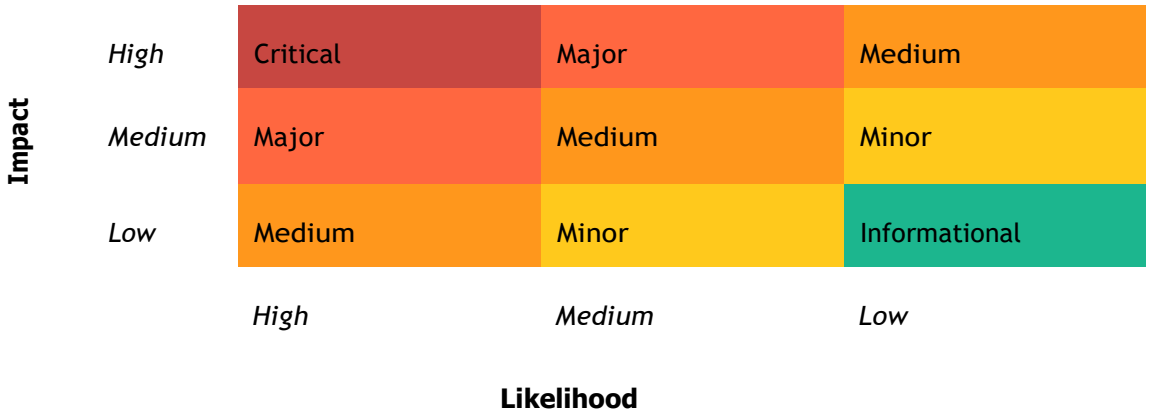
1.3 Audit Methodology

To standardize the evaluation, we define the following terminology based on OWASP Risk Rating Methodology:

- **Likelihood** represents how likely a particular vulnerability is to be uncovered and exploited in the wild;
- **Impact** measures the technical loss and business damage of a successful attack;
- **Severity** demonstrates the overall criticality of the risk.

Likelihood and impact are categorized into three ratings: High, Medium and Low, i.e., H, M and L respectively. Severity is determined by likelihood and impact and can be classified into four categories accordingly, i.e., Critical, Major, Medium, Minor and Informational (Info) as the table below:

© 2025 Electro-Pact. All rights



Electro-Pact firstly analyses the smart contract with open-source and also our own security assessment tools to identify basic bugs related to general smart contracts. These tools include Slither, securify,Mythril, Sūrya, Solgraph, Truffle, Geth, Ganache, Mist, Metamask, solhint, mythx, etc. Then, our security specialists will verify the tool results manually, make a description and decide the severity for each of them.

After that, we go through a checklist of possible issues that could not be detected with automatic tools, conduct test cases for each and indicate the severity level for the results. If no issues are found after manual analysis, the contract can be considered safe within the test case. Else, if any issues are found, we might further deploy contracts on our private testnet and run tests to confirm the findings. We would additionally build a PoC to demonstrate the possibility of exploitation, if required or necessary.

The standard checklist, which applies for every SCA, strictly follows the Smart Contract Weakness Classification Registry (SWC Registry). SWC Registry is an implementation of the weakness classification scheme proposed in The Ethereum Improvement Proposal project under the code EIP-1470. The checklist of testing according to SWC Registry is shown in Appendix A.

In general, the auditing process focuses on detecting and verifying the existence of the following issues:

- **Coding Specification Issues:** Focusing on identifying coding bugs related to general smart contract coding conventions and practices.
- **Design Defect Issues:** Reviewing the architecture design of the smart contract(s) and working on test cases, such as self-DoS attacks, incorrect inheritance implementations, etc.
- **Coding Security Issues:** Finding common security issues of the smart contract(s), for example integer overflows, insufficient verification of authenticity, improper use of cryptographic signature, etc.
- **Coding Design Issues:** Testing the code logic and error handlings in the smart contract code base, such as initializing contract variables, controlling the balance and flows of token transfers, verifying strong randomness, etc.
- **Coding Hidden Dangers:** Working on special issues, such as data privacy, data reliability, gas consumption optimization, special cases of authentication and owner permission, fallback functions, etc.

© 2025 Electro-Pact. All rights

For better understanding of found issues' details and severity, each SWC ID is mapped to the most closely related Common Weakness Enumeration (CWE) ID. CWE is a category system for software weaknesses and vulnerabilities to help identify weaknesses surrounding software jargon. The list in Appendix B provides an overview on specific similar software bugs that occur in Smart Contract coding.

The final report will be sent to the smart contract issuer with an executive summary for overview and detailed results for acts of remediation.

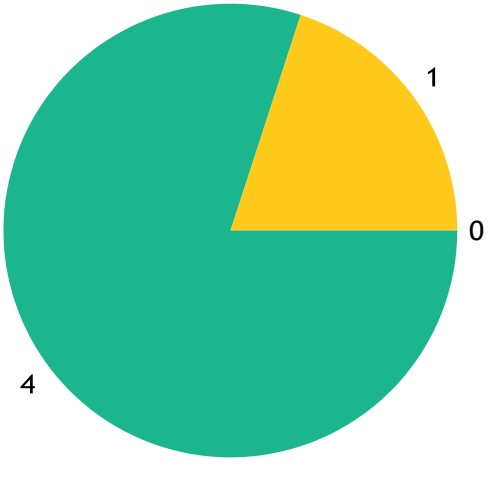
1.4 Audit Scope

Assessment	Target	Type
White-box testing	NeoVaultToken.sol	Solidity code file

© 2025 Electro-Pact. All rights

Executive Summary

Security issues by severity



Legend

- Critical
- Major
- Medium
- Minor
- Info

Security issues by SWC

Function Default Visibility (SWC-100)	1	
Floating Pragma (SWC-103)	1	
Use of Deprecated Solidity Functions (SWC-111)	1	
Requirement Violation (SWC-123)	1	
Code With No Effects (SWC-135)	1	

© 2025 Electro-Pact. All rights

Security issues by CWE

Use of Obsolete Function (CWE-477)	1
Improper Following of Specification by Caller (CWE-573)	1
Improper Control of a Resource Through its Lifetime (CWE-664)	1
Improper Adherence to Coding Standards (CWE-710)	1
Irrelevant Code (CWE-1164)	1

Table of security issues

ID	Status	Vulnerability	Severity
#hne-001	Resolved	Floating pragma	INFO
#hne-002	Resolved	Code with no effects *owner*	INFO
#hne-003	Resolved	Inefficient function declarations	INFO
#hne-004	Resolved	Ignored constructor visibility	INFO
#hne-005	Resolved	Requirements on always-true conditions	MINOR

© 2025 Electro-Pact. All rights

Recommendations

Based on the results of this smart contract audit, Electro-Pact has the following high-level key recommendations:

Key recommendations	
Issues	Electro-Pact conducted a security assessment of smart contracts for NEOV. No issues with severity higher than low had been found. These issues do not represent actual bugs or security problems. After NEOV committed the new codebase for staking functions on GitHub, Electro-Pact produced the re-test and confirmed that all issues were resolved.
Recommendations	Electro-Pact recommends NEOV to evaluate the audit results with several different security audit third-parties for the most accurate conclusion.

© 2025 Electro-Pact. All rights

Detailed Results

1. Floating pragma

Issue ID	#hne-001
Category	SWC-103 - Floating Pragma
Description	Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.
Severity	INFO
Location(s)	NeoVaultToken.sol
Status	Resolved
Reference	CWE-664 - Improper Control of a Resource Through its Lifetime
Remediation	Lock the pragma version and also consider known bugs (BEP-20 Token | Address: 0xb8bD708d0E9A5d811b2e57dC3660769db2AB0A95 | BscScan) for the compiler version that ischosen.

Description

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

/**
 * @title NEO Vault (NEOV) Token Contract
 * @dev Secure ERC20 token contract with burning and forwarding capabilities
 * @author Smart Contract Developer
 *
 * Features:
 * - Fixed total supply of 21 CR (21,000,000,000,000,000,000) tokens
 * - Burning functionality enabled
 * - Forwarding functionality enabled
 * - No minting after initial deployment
 * - No blocking/blacklisting
 * - No callbacks
 * - No locking mechanisms
 */

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/ReentrancyGuard.sol";
```

© 2025 Electro-Pact. All rights

```
import "@openzeppelin/contracts/utils/Pausable.sol";

contract NEOVault is ERC20, ERC20Burnable, Ownable(msg.sender), ReentrancyGuard, Pausable
{

    // Token constants
    uint256 private constant TOTAL_SUPPLY = 210_000_000 * 10**18; // 21 CR tokens
    uint256 public constant MAX_SUPPLY = 210_000_000 * 10**18;    // Max supply cap

    // Events
    event TokensForwarded(address indexed from, address indexed to, uint256 amount);
    event EmergencyWithdraw(address indexed token, address indexed to, uint256 amount);

    /**
     * @dev Constructor that initializes the token with fixed supply
     * All tokens are minted to the contract deployer
     */
    constructor() ERC20("NEO Vault", "NEOV") {
        // Mint total supply to contract deployer
        _mint(msg.sender, TOTAL_SUPPLY);
    }

    /**
     * @dev Returns the number of decimals used for token amounts
     */
    function decimals() public pure override returns (uint8) {
        return 18;
    }

    /**
     * @dev Forward tokens from sender to specified recipients
     * @param recipients Array of recipient addresses
     * @param amounts Array of amounts to forward to each recipient
     * Requirements:
     * - Recipients and amounts arrays must have same length
     * - Sender must have sufficient balance
     * - Contract must not be paused
     */
    function forwardTokens(
        address[] calldata recipients,
        uint256[] calldata amounts
    ) external nonReentrant whenNotPaused {
        require(recipients.length == amounts.length, "NEOV: Arrays length mismatch");
        require(recipients.length > 0, "NEOV: No recipients specified");
        require(recipients.length <= 100, "NEOV: Too many recipients");

        uint256 totalAmount = 0;

        // Calculate total amount needed
        for (uint256 i = 0; i < amounts.length; i++) {
            require(amounts[i] > 0, "NEOV: Amount must be greater than 0");
            totalAmount += amounts[i];
        }

        require(balanceOf(msg.sender) >= totalAmount, "NEOV: Insufficient balance");

        // Execute transfers
        for (uint256 i = 0; i < recipients.length; i++) {
            require(recipients[i] != address(0), "NEOV: Transfer to zero address");
            require(recipients[i] != msg.sender, "NEOV: Cannot forward to self");
```

© 2025 Electro-Pact. All rights

```
            _transfer(msg.sender, recipients[i], amounts[i]);

            emit TokensForwarded(msg.sender, recipients[i], amounts[i]);
        }
    }

    /**
     * @dev Batch transfer tokens to multiple recipients with same amount
     * @param recipients Array of recipient addresses
     * @param amount Amount to send to each recipient
     */
    function batchTransfer(
        address[] calldata recipients,
        uint256 amount
    ) external nonReentrant whenNotPaused {
        require(recipients.length > 0, "NEOV: No recipients specified");
        require(recipients.length <= 100, "NEOV: Too many recipients");
        require(amount > 0, "NEOV: Amount must be greater than 0");

        uint256 totalAmount = amount * recipients.length;
        require(balanceOf(msg.sender) >= totalAmount, "NEOV: Insufficient balance");

        for (uint256 i = 0; i < recipients.length; i++) {
            require(recipients[i] != address(0), "NEOV: Transfer to zero address");
            require(recipients[i] != msg.sender, "NEOV: Cannot transfer to self");

            _transfer(msg.sender, recipients[i], amount);
        }
    }

    /**
     * @dev Enhanced burn function with event emission
     * @param amount Amount of tokens to burn
     */
    function burn(uint256 amount) public override whenNotPaused {
        require(amount > 0, "NEOV: Burn amount must be greater than 0");
        super.burn(amount);
    }

    /**
     * @dev Enhanced burnFrom function with event emission
     * @param account Account to burn tokens from
     * @param amount Amount of tokens to burn
     */
    function burnFrom(address account, uint256 amount) public override whenNotPaused {
        require(amount > 0, "NEOV: Burn amount must be greater than 0");
        require(account != address(0), "NEOV: Burn from zero address");
        super.burnFrom(account, amount);
    }

    /**
     * @dev Override transfer to add pause functionality
     */
    function transfer(address to, uint256 amount) public override whenNotPaused returns
(bool) {
        return super.transfer(to, amount);
    }

    /**
     * @dev Override transferFrom to add pause functionality
     */
```

© 2025 Electro-Pact. All rights

```
    function transferFrom(address from, address to, uint256 amount) public override
whenNotPaused returns (bool) {
        return super.transferFrom(from, to, amount);
    }

    /**
     * @dev Override approve to add pause functionality
     */
    function approve(address spender, uint256 amount) public override whenNotPaused
returns (bool) {
        return super.approve(spender, amount);
    }

    /**
     * @dev Pause all token transfers - Emergency function
     * Can only be called by contract owner
     */
    function pause() external onlyOwner {
        _pause();
    }

    /**
     * @dev Unpause all token transfers
     * Can only be called by contract owner
     */
    function unpause() external onlyOwner {
        _unpause();
    }

    /**
     * @dev Emergency function to withdraw any ERC20 tokens sent to this contract by
mistake
     * @param token Address of the token to withdraw
     * @param to Address to send the tokens to
     * @param amount Amount of tokens to withdraw
     */
    function emergencyWithdrawToken(
        address token,
        address to,
        uint256 amount
    ) external onlyOwner nonReentrant {
        require(token != address(this), "NEOV: Cannot withdraw NEOV tokens");
        require(to != address(0), "NEOV: Invalid recipient address");
        require(amount > 0, "NEOV: Amount must be greater than 0");

        IERC20(token).transfer(to, amount);

        emit EmergencyWithdraw(token, to, amount);
    }

    /**
     * @dev Emergency function to withdraw BNB sent to this contract by mistake
     * @param to Address to send BNB to
     * @param amount Amount of BNB to withdraw in wei
     */
    function emergencyWithdrawBNB(
        address payable to,
        uint256 amount
    ) external onlyOwner nonReentrant {
        require(to != address(0), "NEOV: Invalid recipient address");
        require(amount > 0, "NEOV: Amount must be greater than 0");
```

© 2025 Electro-Pact. All rights

```
        require(address(this).balance >= amount, "NEOV: Insufficient BNB balance");

        (bool success, ) = to.call{value: amount}("");
        require(success, "NEOV: BNB withdrawal failed");
    }

    /**
     * @dev Returns the remaining number of tokens that can be minted
     * Since minting is disabled, this will always return 0
     */
    function remainingMintableSupply() external pure returns (uint256) {
        return 0; // No minting allowed
    }

    /**
     * @dev Check if the contract supports a specific interface
     * @param interfaceId Interface identifier
     */
    function supportsInterface(bytes4 interfaceId) external pure returns (bool) {
        return interfaceId == type(IERC20).interfaceId ||
               interfaceId == type(IERC20Metadata).interfaceId;
    }

    /**
     * @dev Fallback function to receive BNB
     */
    receive() external payable {
        // Accept BNB but don't do anything with it
        // Can be withdrawn using emergencyWithdrawBNB if sent by mistake
    }

    /**
     * @dev Get contract information
     */
    function getContractInfo() external pure returns (
        string memory name,
        string memory symbol,
        uint8 decimalsValue,
        uint256 totalSupplyValue,
        uint256 maxSupplyValue
    ) {
        return (
            "NEO Vault",
            "NEOV",
            18,
            TOTAL_SUPPLY,
            MAX_SUPPLY
        );
    }
}
```

© 2025 Electro-Pact. All rights

Contract ABI

[{"inputs":[],"stateMutability":"nonpayable","type":"constructor"},{"inputs":[{"internalType":"address","name":"spender","type":"address"},{"internalType":"uint256","name":"allowance","type":"uint256"},{"internalType":"uint256","name":"needed","type":"uint256"}],"name":"ERC20InsufficientAllowance","type":"error"},{"inputs":[{"internalType":"address","name":"sender","type":"address"},{"internalType":"uint256","name":"balance","type":"uint256"},{"internalType":"uint256","name":"needed","type":"uint256"}],"name":"ERC20InsufficientBalance","type":"error"},{"inputs":[{"internalType":"address","name":"approver","type":"address"}],"name":"ERC20InvalidApprover","type":"error"},{"inputs":[{"internalType":"address","name":"receiver","type":"address"}],"name":"ERC20InvalidReceiver","type":"error"},{"inputs":[{"internalType":"address","name":"sender","type":"address"}],"name":"ERC20InvalidSender","type":"error"},{"inputs":[{"internalType":"address","name":"spender","type":"address"}],"name":"ERC20InvalidSpender","type":"error"},{"inputs":[],"name":"EnforcedPause","type":"error"},{"inputs":[],"name":"ExpectedPause","type":"error"},{"inputs":[{"internalType":"address","name":"owner","type":"address"}],"name":"OwnableInvalidOwner","type":"error"},{"inputs":[{"internalType":"address","name":"account","type":"address"}],"name":"OwnableUnauthorizedAccount","type":"error"},{"inputs":[],"name":"ReentrancyGuardReentrantCall","type":"error"},{"anonymous":false,"inputs":[{"indexed":true,"internalType":"address","name":"owner","type":"address"},{"indexed":true,"internalType":"address","name":"spender","type":"address"},{"indexed":false,"internalType":"uint256","name":"value","type":"uint256"}],"name":"Approval","type":"event"},{"anonymous":false,"inputs":[{"indexed":true,"internalType":"address","name":"token","type":"address"},{"indexed":true,"internalType":"address","name":"to","type":"address"},{"indexed":false,"internalType":"uint256","name":"amount","type":"uint256"}],"name":"EmergencyWithdraw","type":"event"},{"anonymous":false,"inputs":[{"indexed":true,"internalType":"address","name":"previousOwner","type":"address"},{"indexed":true,"internalType":"address","name":"newOwner","type":"address"}],"name":"OwnershipTransferred","type":"event"},{"anonymous":false,"inputs":[{"indexed":false,"internalType":"address","name":"account","type":"address"}],"name":"Paused","type":"event"},{"anonymous":false,"inputs":[{"indexed":true,"internalType":"address","name":"from","type":"address"},{"indexed":true,"internalType":"address","name":"to","type":"address"},{"indexed":false,"internalType":"uint256","name":"amount","type":"uint256"}],"name":"TokensForwarded","type":"event"},{"anonymous":false,"inputs":[{"indexed":true,"internalType":"address","name":"from","type":"address"},{"indexed":true,"internalType":"address","name":"to","type":"address"},{"indexed":false,"internalType":"uint256","name":"value","type":"uint256"}],"name":"Transfer","type":"event"},{"anonymous":false,"inputs":[{"indexed":false,"internalType":"address","name":"account","type":"address"}],"name":"Unpaused","type":"event"},{"inputs":[],"name":"MAX_SUPPLY","outputs":[{"internalType":"uint256","name":"","type":"uint256"}],"stateMutability":"view","type":"function"},{"inputs":[{"internalType":"address","name":"owner","type":"address"},{"internalType":"address","name":"spender","type":"address"}],"name":"allowance","outputs":[{"internalType":"uint256","name":"","type":"uint256"}],"stateMutability":"view","type":"function"},{"inputs":[{"internalType":"address","name":"spender","type":"address"},{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"approve","outputs":[{"internalType":"bool","name":"","type":"bool"}],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"address","name":"account","type":"address"}],"name":"balanceOf","outputs":[{"internalType":"uint256","name":"","type":"uint256"}],"stateMutability":"view","type":"function"},{"inputs":[{"internalType":"address[]","name":"recipients","type":"address[]"},{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"batchTransfer","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"burn","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"address","name":"account","type":"address"},{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"burnFrom","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[],"name":"decimals","outputs":[{"internalType":"uint8","name":"","type":"uint8"}],"stateMutability":"pure","type":"function"},{"inputs":[{"internalType":"address payable","name":"to","type":"address"},{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"emergencyWithdrawBNB","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"address","name":"token","type":"address"},{"internalType":"address","name":"to","type":"address"},{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"emergencyWithdrawToken","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"address[]","name":"recipients","type":"address[]"},{"internalType":"uint256[]","name":"amounts","type":"uint256[]"}],"name":"forwardTokens","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[],"name":"getContractInfo","outputs":[{"internalType":"string","name":"name","type":"string"},{"internalType":"string","name":"symbol","type":"string"},{"internalType":"uint8","name":"decimalsValue","type":"uint8"},{"internalType":"uint256","name":"totalSupplyValue","type":"uint256"},{"internalType":"uint256","name":"maxSupplyValue","type":"uint256"}],"stateMutability":"pure","type":"function"},{"inputs":[],"name":"name","outputs":[{"internalType":"string","name":"","type":"string"}],"stateMutability":"view","type":"function"},{"inputs":[],"name":"owner","outputs":[{"internalType":"address","name":"","type":"address"}],"stateMutability":"view","type":"function"},{"inputs":[],"name":"pause","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[],"name":"paused","outputs":[{"internalType":"bool","name":"","type":"bool"}],"stateMutability":"view","type":"function"},{"inputs":[],"name":"remainingMintableSupply","outputs":[{"internalType":"uint256","name":"","type":"uint256"}],"stateMutability":"pure","type":"function"},{"inputs":[],"name":"renounceOwnership","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"bytes4","name":"interfaceId","type":"bytes4"}],"name":"supportsInterface","outputs":[{"internalType":"bool","name":"","type":"bool"}],"stateMutability":"pure","type":"function"},{"inputs":

© 2025 Electro-Pact. All rights

[],"name":"symbol","outputs":[{"internalType":"string","name":"","type":"string"}],"stateMutability":"view","type":"function"},{"inputs":[],"name":"totalSupply","outputs":[{"internalType":"uint256","name":"","type":"uint256"}],"stateMutability":"view","type":"function"},{"inputs":[{"internalType":"address","name":"to","type":"address"},{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"transfer","outputs":[{"internalType":"bool","name":"","type":"bool"}],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"address","name":"from","type":"address"},{"internalType":"address","name":"to","type":"address"},{"internalType":"uint256","name":"amount","type":"uint256"}],"name":"transferFrom","outputs":[{"internalType":"bool","name":"","type":"bool"}],"stateMutability":"nonpayable","type":"function"},{"inputs":[{"internalType":"address","name":"newOwner","type":"address"}],"name":"transferOwnership","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":[],"name":"unpause","outputs":[],"stateMutability":"nonpayable","type":"function"},{"stateMutability":"payable","type":"receive"}]

Contract Creation code

60806040523480156200001157600080fd5b50336040518060400160405280600981526020016813915 3c815985d5b1d60ba1b8152506040518060400160405280600481526020016327 22a7ab60e11b8152508160039081620000639190620003 3b565b506004620000728282620003 3b565b505050600160016 0a01b038116620000a557604051631e4fbdf760e01b8152 6000600482015260240 15b60405180910390fd5b620000b081620000d3565b506001600655620000cd336aadb53acfa41ae e120000006200012 5565b6200042f565b600580546001600160a01b038381166001600160a01b0319831681179093556040 5191169190829 07f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e090600090a35050565b 6001600160a01b038216620001515760405163ec442f0560e01b8152600060048201526024016200009c565b6200015f60 0083836200016 3565b5050565b6001600160a01b038316620001925780600260008282546200018691 9062000407565b9 0915550620002069 0505565b6001600160a01b038316600090815260208190526040902054818110156200001e757604051 63391434e360e21b81526001600160a01b038516600482015260248101829052604481018390526064016200009c565b6 001600160a01b038416600090815260208190526040902090829003905 55b6001600160a01b038216620002245760028 5482900390556200024 3565b6001600160a01b038216600090815260208190526040902080548201905 55b816001600160 0a01b031683600160016 0a01b03167fddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef8360 4051620002899181526020019056 5b60405180910390a3505050565b634e487b7160e01b600052604160045260246000fd 5b600181811c90821680620002c157607f821691505b602082108103620002e257634e487b7160e01b6000526022600452 60246000fd5b50919050565b601f821115620003365760008181526020812060 1f850160051c810160208610156200031 5750805b601f850160051c820191505b818110156200033257828155600101620003 1d565b5050505b505050565b815160 01600160401b038111156200035757620003576200029656 5b6200036f816200036884545 4620002ac565b84620002e8565b 602080601f831160018114620003a75760008415620003 8e5750858301515b600019600386901b1c1916600185901b1785 5562000332565b6000858152602081206 01f198616915b82811015620003 d857888601518255948401946001909101908401620003b7565b5085821015620003f757878501516000196003889 01b60f8161c191681555b5050505050600190811b01 905550565b80820180821115620004295763 4e487b7160e01b600052601160045260246000fd5b92915050565b611a2d8 06200043f6000396000f3fe6080604052600436 1061016a5760003560e01c806370a08231116100d15780638da5cb5b116 1008a578063a9059cbb11610064578063a9059cbb14610450578063d1a75c5414610470578063dd62ed3e146104905780 63f2fde38b146104d657600080fd5b80638da5cb5b146103f357806395d89b411461041b578063a0ea0dd6146104305760 0080fd5b806370a08231146102ea578063715018a6146103205780637cc6790146103355780637cc1f867146103555780 6383f12fec146103be5780638456cb59146103de57600080fd5b8063313ce567116101023578063313ce5671461024e5780 6332cb6b0c1461026a578063349f0b9014610289578633f4ba83a1461029d57806342966c68146102b25780635c975abb 146102d257600080fd5b806301ffc9a71461017657806306fdde03146101ab578063095ea7b3146101cd57806318160ddd 146101ed57806323b872dd1461020c578063277327a51461022c57600080fd5b3661017157005b600080fd5b3480156101 8257600080fd5b506101966101913660046115e3565b6104f6565b6040519015158152602001 5b60405180910390f35b34 80156101b757600080fd5b506101c061052d565b6040516101a29190611653565b3480156101d957600080fd5b50610196 6101e836600461167b565b6105bf565b3480156101f957600080fd5b506002545b604051908152602001 6101a2565b3480 1561021857600080fd5b506101966102273660046116a7565b6105da565b348015610238 57600080fd5b5061024c610247 3660046116a7565b6105f7565b005b34801561025a57600080fd5b506040516012815260200161 01a2565b348015610276 57600080fd5b506101fe6aadb53acfa41aee1200000081565b3480156102957600080fd5b5060006101fe565b34801561 02a957600080fd5b5061024c6107b4565b3480156102be57600080fd5b5061024c6102cd3660046116e8565b6107c6565b 3480156102de57600080fd5b5060075460ff16610196565b3480156102f657600080fd5b506101fe6103053660046117015 65b6001600160a01b0316600090815260208190526040902054905 90565b34801561032c57600080fd5b5061024c6107fa565b 3480156103415760080fd5b5061024c6103536600461167b565b61080c565b3480156103615760080fd5b506040805 1808201825260098152681 39153c815985d5b1d60ba1b602080830191909152825180840190935260048352632722a7ab 60e11b908301529060126aadb53acfa41aee1200000080604051610 1a2959493929190611171e565b3480156103ca576000 80fd5b5061024c6103d93660046117b0565b610898565b3480156103ea57600080fd5b5061024c610b27565b3480156103 ff57600080fd5b506005546040516001600160a01b0390911681526020016101a2565b34801561042757600080fd5b5061 01c0610b37565b34801561043c57600080fd5b5061024c61044b36600461167b565b610b46565b34801561045c57600080 fd5b5061019661046b36600461167b565b610cca565b34801561047c57600080fd5b5061024c61048b3660046117fc565b 610cde565b34801561049c57600080fd5b506101fe6104ab3660046118 68565b6001600160a01b0391821660009081526001602090815260408083209390941682529190915220 5490565b3480156104e257600080fd5b5061024c6104f136600461 1701565b6110aa565b60006001600160e01b031982166336372b0760e01b1480610527575060016001600160e01b031982166 3a219a02560e01b145b92915050565b6060600380546105 3c906118a1565b80601f0160208091040260200160405190810

© 2025 Electro-Pact. All rights

160405280929190818152602001828054610568906118a1565b80156105b55780601f1061058a576101008083540402835291602001916105b5565b820191906000526020600020905b81548152906001019060200180831161059857829003601f168201915b505050505050905090565b60006105c96110e5565b6105d38383611109565b9392505050565b60006105e46110e5565b6105ef848484611121565b94935050505050565b6105ff611145565b610607611172565b306001600160a01b0384160361066e5760405162461bcd60e51b815260206004820152602160248201527f4e454f563a2043616e6e6f74207769746864726177204e454f5620746f6b656e6044820152607360f81b60648201526084015b60405180910390fd5b6001600160a01b0382166106c45760405162461bcd60e51b815260206004820152601f60248201527f4e454f563a20496e76616c696420726563697069656e742061646472657373006044820152606401610665565b600081116106e45760405162461bcd60e51b8152600401610665906118db565b60405163a9059cbb60e01b81526001600160a01b0383811660048301526024820183905284169063a9059cbb906044016020604051808303816000875af1158015610733573d6000803e3d6000fd5b505050506040513d601f19601f82011682018060405250810190610757919061191e565b50816001600160a01b0316836001600160a01b03167ff24ef89f38eadc1bde50701ad6e4d6d11a2dc24f7cf834a486991f38833285048360405161079d91815260200200190565b60405180910390a36107af6001600655565b505050565b6107bc611145565b6107c461119c565b565b6107ce6110e5565b600081116107ee5760405162461bcd60e51b815260040161066590611940565b6107f7816111ee565b50565b610802611145565b6107c460006111f8565b6108146110e5565b600081116108345760405162461bcd60e51b815260040161066590611940565b6001600160a01b03821661088a5760405162461bcd60e51b815260206004820152601c60248201527f4e454f563a204275726e2066726f6d207a65726f2061646472657373000000006044820152606401610665565b610894828261124a565b5050565b6108a0611172565b6108a86110e5565b816108f55760405162461bcd60e51b815260206004820152601d60248201527f4e454f563a204e6f20726563697069656e7473207370656369666965640000006044820152606401610665565b60648211156109425760405162461bcd60e51b81526020600482015260196024820152784e454f563a20546f6f206d616e7920726563697069656e747360381b6044820152606401610665565b600081116109625760405162461bcd60e51b8152600401610665906118db565b600061096e838361199e565b336000908152602081905260409020549091508111156109d05760405162461bcd60e51b815260206004820152601a60248201527f4e454f563a20496e73756666696369656e742062616c616e6365000000000000006044820152606401610665565b60005b83811015610b1b5760008585838181106106109ef576109ef6119b5565b9050602002016020810190610a049190611701565b6001600160a01b031603610a5a5760405162461bcd60e51b815260206004820152601e60248201527f4e454f563a205472616e7366657220746f207a65726f206164647265737300006044820152606401610665565b33858583818110610a6d57610a6d6119b5565b9050602002016020810190610a829190611701565b6001600160a01b031603610ad85760405162461bcd60e51b815260206004820152601d60248201527f4e454f563a2043616e6e6f74207472616e7366657220746f2073656c6600000006044820152606401610665565b610b0933868684818110610aee57610aee6119b5565b9050602002016020810190610b039190611701565b8561125f565b80610b13816119cb565b9150506106d3565b50506107af6001600655565b610b2f611145565b6107c46112be565b60606004805461053c906118a1565b610b4e611145565b610b56611172565b6001600160a01b038216610bac5760405162461bcd60e51b815260206004820152601f60248201527f4e454f563a20496e76616c696420726563697069656e742061646472657373006044820152606401610665565b60008111610bcc5760405162461bcd60e51b8152600401610665906118db565b80471015610c1c5760405162461bcd60e51b815260206004820152601e60248201527f4e454f563a20496e73756666696369656e7420424e422062616c616e636500006044820152606401610665565b6000826001600160a01b03168260405160006040518083038185875af1925050503d8060008114610c69576040519150601f19603f3d011682016040523d82523d6000602084013e610c6e565b606091505b5050905080610cbf5760405162461bcd60e51b815260206004820152601b60248201527f4e454f563a20424e42207769746864726177616c206661696c65640000000000006044820152606401610665565b506108946001600655565b6000610cd46110e5565b6105d383836112fb565b610ce6611172565b610cee6110e5565b828114610d3d5760405162461bcd60e51b815260206004820152601c60248201527f4e454f563a204172726179732066656e677468206d69736d61746368000000006044820152606401610665565b82610d8a5760405162461bcd60e51b815260206004820152601d60248201527f4e454f563a204e6f20726563697069656e7473207370656369666965640000006044820152606401610665565b6064831115610dd75760405162461bcd60e51b81526020600482015260196024820152784e454f563a20546f6f206d616e7920726563697069656e747360381b6044820152606401610665565b6000805b82811015610e53576000848483818110610df757610df76119b5565b9050602002013511610e1b5760405162461bcd60e51b8152600401610665906118db565b838382818110610e2d57610e2d6119b5565b9050602002013582610e3f91906119e4565b915080610e4b816119cb565b915050610ddb565b5033600090815260208190526040902054811115610eb35760405162461bcd60e51b815260206004820152601a60248201527f4e454f563a20496e73756666696369656e742062616c616e6365000000000000006044820152606401610665565b60005b8481101561098576000868683818110610ed257610ed26119b5565b9050602002016020810190610ee79190611701565b6001600160a01b031603610f3d5760405162461bcd60e51b815260206004820152601e60248201527f4e454f563a205472616e7366657220746f207a65726f206164647265737300006044820152606401610665565b33868683818110610f5057610f506119b5565b9050602002016020810190610f659190611701565b6001600160a01b031603610fbb5760405162461bcd60e51b815260206004820152601c60248201527f4e454f563a2043616e6e6f7420666f727761726420746f2073656c66000000006044820152606401610665565b61100433878784818110610fd157610fd16119b5565b9050602002016020810190610fe69190611701565b868685818110610ff857610ff86119b5565b9050602002013561125f565b858582818110611016576110166119b5565b905060200201602081019061102b9190611701565b6001600160a01b0316337fce1420785bf132685eca483be91a0cf0736cb25ab630933d2281380739b30ae6868685818110611068576110686119b5565b9050602002013560405161107e91815260200190565b60405180910390a380611090816119cb565b915050610eb6565b50506110a46001600655565b50505050565b6110b2611145565b6001600160a01b0381166110dc57604051631e4fbdf760e01b815260006004820152602401610665565b6107f7816111f8565b60075460ff16156107c45760405163d93c066560e01b815260040160405180910390fd5b600033611117818585611309565b506001939250505050565b60003361112f858285611316565b61113a85858561125f565b5060019493505050505050565b6005546001600160a01b031633146107c45760405163118cdaa760e01b8152336004820152602401610665565b60026006540361119557604051633ee5aeb560e01

© 2025 Electro-Pact. All rights

b8152600401604051809103 90fd5b6002600655565b6111a461138f565b6007805460ff191690557f5db9ee0a495bf2e6ff
9c91a7834c1ba4fdd244a5e8aa4e537bd38aeae4b073aa335b6040516001600160a01b039091168152602001604051809
10390a1565b6107f733826113b2565b60058054600160016 00160a01b038381166001600160a01b0319831681179093556040 5
19116919082907f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e090600090a35050565b61
125582338361131 6565b61089482826113b2565b6001600160a01b038316611289576040516 34b637e8f60e11b8152600
0600482015260240161066556 5b6001600160a01b0382166112b35760405163ec442f0560e01b8152600060048201526 02
401610665565b6107af8383836113e4565b6112c66110e5565b6007805460ff191660011790557f62e78cea01bee320cd4
e420270b5ea74000d11b0c9f74754ebdbfc544b05a2586111d13390565b60003361111781858561125f565b6107af83838
3600161150e565b6001600160a01b038381166000908152600160209081526040808320938616835292905220540600019
8110156110a4578181101561138057604051637dc7a0d960e11b8152600160016 0a01b038416600482015260248101829
052604481018390526064016106655 65b6110a484848484036000611 50e565b60075460ff166107c457604051638dfc202
b60e01b8152600401604051809103 90fd5b6001600160a01b0382166113dc57604051634b637e8f60e11b8152600060048
201526024016106655 65b610894826000835b6001600160a01b0383166114 0f57806002600082825461140491906119e45
65b909155506114819050565b6001600160a01b038316600090815260208190526040902054818110156114625760405 1
63391434e360e21b81526001600160a01b038516600482015260248101829052604481018390526064016106655 65b600
1600160a01b038416600090815260208190526040902090828900390555b6001600160a01b03821661149d576002805482
900390556114bc565b6001600160a01b038216600090815260208190526040902080548201905 55b816001600160a01b0
316836001600160a01b03167fddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef8360405161
150191815260200190565b60405180910390a3505050565b6001600160a01b038416611538576040516 3e602df0560e01b
8152600060048201526024016106655 65b6001600160a01b038316611562576040516 34a1406b160e11b8152600060048
201526024016106655 65b6001600160a01b038085166000908152600160209081526040808320938716835292905220 82
905580156110a457826001600160a01b031684600160016 0a01b03167f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0
f7b2291e5b200ac8c7c3b925846040516115d591815260200190565b60405180910390a350505050565b60006020828403
12156115f557600080fd5b81356001600160e01b031981168114610 5d357600080fd5b60008151808452600 5b81811015
6116335760208185018101518683018201520161161756 5b506000602082860101526020601f19601f8301168501019150
5092915050565b602081526000610 5d360208301846116 0d565b6001600160a01b038116814610 7f757600080fd5b6000
806040838503121561168e57600080fd5b8235611699816116116665 65b946020939093013593505050565b60008060006060
84860312156116bc57600080fd5b83356116c781611666565b9250602084013561 16d781611666565b9295929450505060
409190910135905 65b60006020828403121561 56116fa57600080fd5b5035919050565b60006020828403121561171357600 0
80fd5b81356105d381611666565b60a0815260006117 3160a083018861160d565b828103602084015261174381886116 0
d565b60ff969096166040840152505060608101929092526080909101529291505 0565b60008083601f840112611776576
00080fd5b50813567ffffffffffffffff81111561178e57600080fd5b6020830191508360208260051b85010111156117a95760
0080fd5b9250929050565b60008060006040848603121561 17c557600080fd5b833567ffffffffffffffff8111156117dc57600
080fd5b6117e886828701611764565b909790965060209590950135949350505050565b60008060008060840858703121 56
1181257600080fd5b843567ffffffffffffffff8082111561182a57600080fd5b6118368883890161176456 5b90965094506020
870135915080821115618418 4f57600080fd5b5061185c878288016117 64565b959894975095505050565b600080604083
8503121561187b57600080fd5b8235611886816116116665 65b9150602083013561189681611666565b809150509250929050
565b600181811c908216806118b557607f821691505b6020821081036118d557634e487b7160e01b600052602260045260
246000fd5b50919050565b6020808252602390820152 7f4e454f563a20416d6f756e74206d7573742065206772656174
6572207468616040820152 6206e20360ec1b6060820152 6080019056 5b60006020828403121561193057600080fd5b8151
80151581146105d357600080fd5b6020808252602890820152 7f4e454f563a204275726e20616d6f756e74206d75737420
626520677265617465604082015267072207468616e20360c41b6060820152 6080019056 5b634e487b7160e01b6000526
011600452602460 00fd5b808202811582820484141761052757610527611988565b634e487b7160e01b600052603260045
260246000fd5b600060018201611 9dd576119dd611988565b5060010190565b808201808211156105275761052761198 8
56fea2646970667358221220aa97ba4723cd0dddb9c24174fb809d5af67d544ae9e5ece6f8dcc9c9ade1a47e64736f6c63
430008140033

Conclusion

Electro-Pact had conducted a security audit for NEO VAULT (NEOV) staking functions. Total 0 issues werefound, but none of these issues represented actual bugs or security problems. These issues thenwere accepted by the NEO VAULT (NEOV) team.

To improve the quality for this report, and for Electro-Pact's Smart Contract Audit report in general, wegreatly appreciate any constructive feedback or suggestions, on our methodology, audit findings, orpotential gaps in scope/coverage.

© 2025 Electro-Pact. All rights

Appendices

Appendix A – Security Issue Status Definitions

Status	Definition
Open	The issue has been reported and currently being review by the smart contract developers/issuer.
Unresolved	The issue is acknowledged and planned to be addressed in future. At the time of the corresponding report version, the issue has not been fixed.
Resolved	The issue is acknowledged and has been fully fixed by the smart contract developers/issuer.
Rejected	The issue is considered to have no security implications or to make only little security impacts, so it is not planned to be addressed and won't be fixed.

© 2025 Electro-Pact. All rights

Appendix B – Severity Explanation

Severity	Definition
CRITICAL	Issues, considered as critical, are straightforwardly exploitable bugs and security vulnerabilities. It is advised to immediately resolve these issues in order to prevent major problems or a full failure during contract system operation.
MAJOR	Major issues are bugs and vulnerabilities, which cannot be exploited directly without certain conditions. It is advised to patch the codebase of the smart contract as soon as possible, since these issues, with a high degree of probability, can cause certain problems for operation of the smart contract or severe security impacts on the system in some way.
MEDIUM	In terms of medium issues, bugs and vulnerabilities exist but cannot be exploited without extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
MINOR	Minor issues are generally objective in nature but do not represent actual bugs or security problems. It is advised to address these issues, unless there is a clear reason not to.
INFO	Issues, regarded as informational (info), possibly relate to “guides for the best practices” or “readability”. Generally, these issues are not actual bugs or vulnerabilities. It is recommended to address these issues, if it make effective and secure improvements to the smart contract codebase.

© 2025 Electro-Pact. All rights

Appendix C – Smart Contract Weakness Classification Registry (SWC Registry)

ID	Name	Description
	Coding Specification Issues	
SWC-100	Function Default Visibility	It is recommended to make a conscious decision on which visibility type (*external*, *public*, *internal* or *private*) is appropriate for a function. By default, functions without concrete specifiers are *public*.
SWC-102	Outdated Compiler Version	It is recommended to use a recent version of the Solidity compiler to avoid publicly disclosed bugs and issues in outdated versions.
SWC-103	Floating Pragma	It is recommended to lock the pragma to ensure that contracts do not accidentally get deployed using.
SWC-108	State Variable Default Visibility	Variables can be specified as being *public*, *internal* or *private*. Explicitly define visibility for all state variables.
SWC-111	Use of Deprecated Solidity Functions	Solidity provides alternatives to the deprecated constructions, the use of which might reduce code quality. Most of them are aliases, thus replacing old constructions will not break current behavior.
SWC-118	Incorrect Constructor Name	It is therefore recommended to upgrade the contract to a recent version of the Solidity compiler and change to the new constructor declaration (the keyword *constructor*).
	Design Defect Issues	
SWC-113	DoS with Failed Call	External calls can fail accidentally or deliberately, which can cause a DoS condition in the contract. It is better to isolate each external call into its own transaction and implement the contract logic to handle failed calls.

© 2025 Electro-Pact. All rights

SWC-119	Shadowing State Variables	Review storage variable layouts for your contract systems carefully and remove any ambiguities. Always check for compiler warnings as they can flag the issue within a single contract.
SWC-125	Incorrect Inheritance Order	When inheriting multiple contracts, especially ifthey have identical functions, a developer should carefully specify inheritance in the correct order (from more /general/ to more /specific/).
SWC-128	DoS With Block Gas Limit	Modifying an array of unknown size, that increases in size over time, can lead to such a Denial of Service condition. Actions that require looping across the entire data structure should be avoided.
	Coding Security Issues	
SWC-101	Integer Overflow and Underflow	It is recommended to use safe math libraries for arithmetic operations throughout the smart contract system to avoid integer overflows and underflows.
SWC-107	Reentrancy	Make sure all internal state changes are performed before the call is executed or use a reentrancy lock.
SWC-112	Delegatecall to Untrusted Callee	Use *delegatecall* with caution and make sure to never call into untrusted contracts. If the target address is derived from user input ensure to check it against a whitelist of trusted contracts.
SWC-117	Signature Malleability	A signature should never be included into a signed message hash to check if previously messages have been processed by the contract.
SWC-121	Missing Protection against Signature Replay Attacks	In order to protect against signature replay attacks, store every message hash that has been processed by the smart contract, include the address of the contract that processes the message and never generate the message hash including the signature.
SWC-122	Lack of Proper Signature Verification	It is not recommended to use alternate verification schemes that do not require proper signature verification through *ecrecover()*.

© 2025 Electro-Pact. All rights

SWC-130	Right-To-Left-Override control character (U+202E)	The character *U+202E* should not appear in the source code of a smart contract.
	Coding Design Issues	
SWC-104	Unchecked Call Return Value	If you choose to use low-level call methods (e.g. *call()*), make sure to handle the possibility that the call fails by checking the return value.
SWC-105	Unprotected Ether Withdrawal	Implement controls so withdrawals can only be triggered by authorized parties or according to the specs of the smart contract system.
SWC-106	Unprotected SELFDESTRUCT Instruction	Consider removing the self-destruct functionality. If absolutely required, it is recommended to implement a multisig scheme so that multiple parties must approve the self-destruct action.
SWC-110	Assert Violation	Consider whether the condition checked in the *assert()* is actually an invariant. If not, replace the *assert()* statement with a *require()* statement.
SWC-116	Block values as a proxy for time	Developers should write smart contracts with the notion that block values are not precise, and the use of them can lead to unexpected effects. Alternatively, they may make use oracles.
SWC-120	Weak Sources of Randomness from Chain Attributes	To avoid weak sources of randomness, use commitment scheme, e.g. RANDAO, external sources of randomness via oracles, e.g. Oraclize, or Bitcoin block hashes.
SWC-123	Requirement Violation	If the required logical condition is too strong, it should be weakened to allow all valid external inputs. Otherwise, make sure no invalid inputs are provided.
SWC-124	Write to Arbitrary Storage Location	As a general advice, given that all data structures share the same storage (address) space, one should make sure that writes to one data structure cannot inadvertently overwrite entries of another data structure.

© 2025 Electro-Pact. All rights

SWC-132	Unexpected Ether balance	Avoid strict equality checks for the Ether balance in a contract.
SWC-133	Hash Collisions With Multiple Variable Length Arguments	When using *abi.encodePacked()*, it's crucial to ensure that a matching signature cannot beachieved using different parameters. Alternatively, you can simply use *abi.encode()* instead. It is also recommended to use replay protection.
	Coding Hidden Dangers	
SWC-109	Uninitialized Storage Pointer	Uninitialized local storage variables can point to unexpected storage locations in the contract. If a local variable is sufficient, mark it with *memory*, else *storage* upon declaration. As of compiler version 0.5.0 and higher this issue has been systematically resolved.
SWC-114	Transaction Order Dependence	A possible way to remedy for race conditions in submission of information in exchange for a reward is called a commit reveal hash scheme. The best fix for the ERC20 race condition is to add a field to the inputs of approve which is the expected current value and to have approve revert or add a safe approve function.
SWC-115	Authorization through tx.origin	*tx.origin* should not be used for authorization. Use *msg.sender* instead.
SWC-126	Insufficient Gas Griefing	Insufficient gas griefing attacks can be performed on contracts which accept data and use it in a sub-call on another contract. To avoid them, only allow trusted users to relay transactions and require that the forwarder provides enough gas.
SWC-127	Arbitrary Jump with Function Type Variable	The use of assembly should be minimal. A developer should not allow a user to assign arbitrary values to function type variables.

© 2025 Electro-Pact. All rights

SWC-129	Typographical Error	The weakness can be avoided by performing pre-condition checks on any math operation or using a vetted library for arithmetic calculations such as SafeMath developed by OpenZeppelin.
SWC-131	Presence of unused variables	Remove all unused variables from the code base.
SWC-134	Message call with hardcoded gas amount	Avoid the use of *transfer()* and *send()* and do not otherwise specify a fixed amount of gas when performing calls. Use *.call.value(...)(“”)* instead.
SWC-135	Code With No Effects	It’s important to carefully ensure that your contract works as intended. Write unit tests to verify correct behaviour of the code.
SWC-136	Unencrypted Private Data On-Chain	Any private data should either be stored off-chain, or carefully encrypted.

© 2025 Electro-Pact. All rights

Appendix D – Related Common Weakness Enumeration (CWE)

The SWC Registry loosely aligned to the terminologies and structure used in the CWE while overlaying a wide range of weakness variants that are specific to smart contracts.

CWE IDs *, to which SWC Registry is related, are listed in the following table:

CWE ID	Name	Related SWC IDs
CWE-284	**Improper Access Control**	SWC-105, SWC-106
CWE-294	Authentication Bypass by Capture-replay	SWC-133
CWE-664	**Improper Control of a Resource Through its Lifetime**	SWC-103
CWE-123	Write-what-where Condition	SWC-124
CWE-400	Uncontrolled Resource Consumption	SWC-128
CWE-451	User Interface (UI) Misrepresentation of Critical Information	SWC-130
CWE-665	Improper Initialization	SWC-118, SWC-134
CWE-767	Access to Critical Private Variable via Public Method	SWC-136
CWE-824	Access of Uninitialized Pointer	SWC-109
CWE-829	Inclusion of Functionality from Untrusted Control Sphere	SWC-112, SWC-116
CWE-682	**Incorrect Calculation**	SWC-101
CWE-691	**Insufficient Control Flow Management**	SWC-126
CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization (“Race Condition”)	SWC-114
CWE-480	Use of Incorrect Operator	SWC-129
CWE-667	Improper Locking	SWC-132
CWE-670	Always-Incorrect Control Flow Implementation	SWC-110
CWE-696	Incorrect Behavior Order	SWC-125
CWE-841	Improper Enforcement of Behavioral Workflow	SWC-107
CWE-693	**Protection Mechanism Failure**	

© 2025 Electro-Pact. All rights

CWE-330	Use of Insufficiently Random Values	SWC-120
CWE-345	Insufficient Verification of Data Authenticity	SWC-122
CWE-347	Improper Verification of Cryptographic Signature	SWC-117, SWC-121
CWE-703	**Improper Check or Handling of Exceptional Conditions**	SWC-113
CWE-252	Unchecked Return Value	SWC-104
CWE-710	**Improper Adherence to Coding Standards**	SWC-100, SWC-108, SWC-119
CWE-477	Use of Obsolete Function	SWC-111, SWC-115
CWE-573	Improper Following of Specification by Caller	SWC-123
CWE-695	Use of Low-Level Functionality	SWC-127
CWE-1164	Irrelevant Code	SWC-131, SWC-135
CWE-937	**Using Components with Known Vulnerabilities**	SWC-102

** CWE IDs, which are presented in bold, are the greatest parent nodes of those nodes following it.*

All IDs in the CWE list above are relevant to the view "Research Concepts" (CWE-1000), except for CWE-937, which is relevant to the "Weaknesses in OWASP Top Ten (2013)" (CWE-928).

© 2025 Electro-Pact. All rights