3

**Security Assessment**

[Disclaimer 3](#_TOC_250009)

[Description 5](#_TOC_250008)

Engagement 5

Project Engagement 5

Contract Link 6

[Risk Level Classification 7](#_TOC_250006)

[Methodology 8](#_TOC_250005)

Used Code from other Frameworks / Smart Contracts (Imports) 9

[Description 10](#_TOC_250004)

[Scope of Work 12](#_TOC_250003)

[Inheritance Graph 13](#_TOC_250002)

Verify Claim 14

Overall Checkup 18

[Write Functions of Contract 19](#_TOC_250001)

SWC Attack 20

Audit Result 24

[Audit Comments 25](#_TOC_250000)

# Disclaimer

**ContractWolf.io** audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

**ContractWolf** does not provide any warranty on its released reports.

**ContractWolf** should not be used as a decision to invest into an audited project and is not affiliated nor partners to its audited contract projects.

**ContractWolf** provides transparent report to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within it’s **SMART CONTRACT**.

**ContractWolf** presence is to analyze, audit and assess the client's smart contract's code.

Each company or projects should be liable to its security flaws and functionalities.

# Network

Binance Smart Chain (BEP20)

# Website

https://vaquita.site/

# Telegram

https://t.me/vaquita

https://t.me/Vaquitaannouncements

# Twitter

https://twitter.com/vaquita

# Discord

https://discord.com/invite/vrTaTHw7Z3

# Description

Saving the aquatic life P2E with NFTs on blockchain.

# ContractWolf Engagements

12th of August 12, 2022 , **Vaquita** engaged and agrees to audit their smart contract's code by **ContractWolf**. The goal of this engagement was to identify if there is a possibility of security flaws in the implementation of the contract or system.

**ContractWolf** will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, whitepaper and repository which has been provided by **Vaquita.**

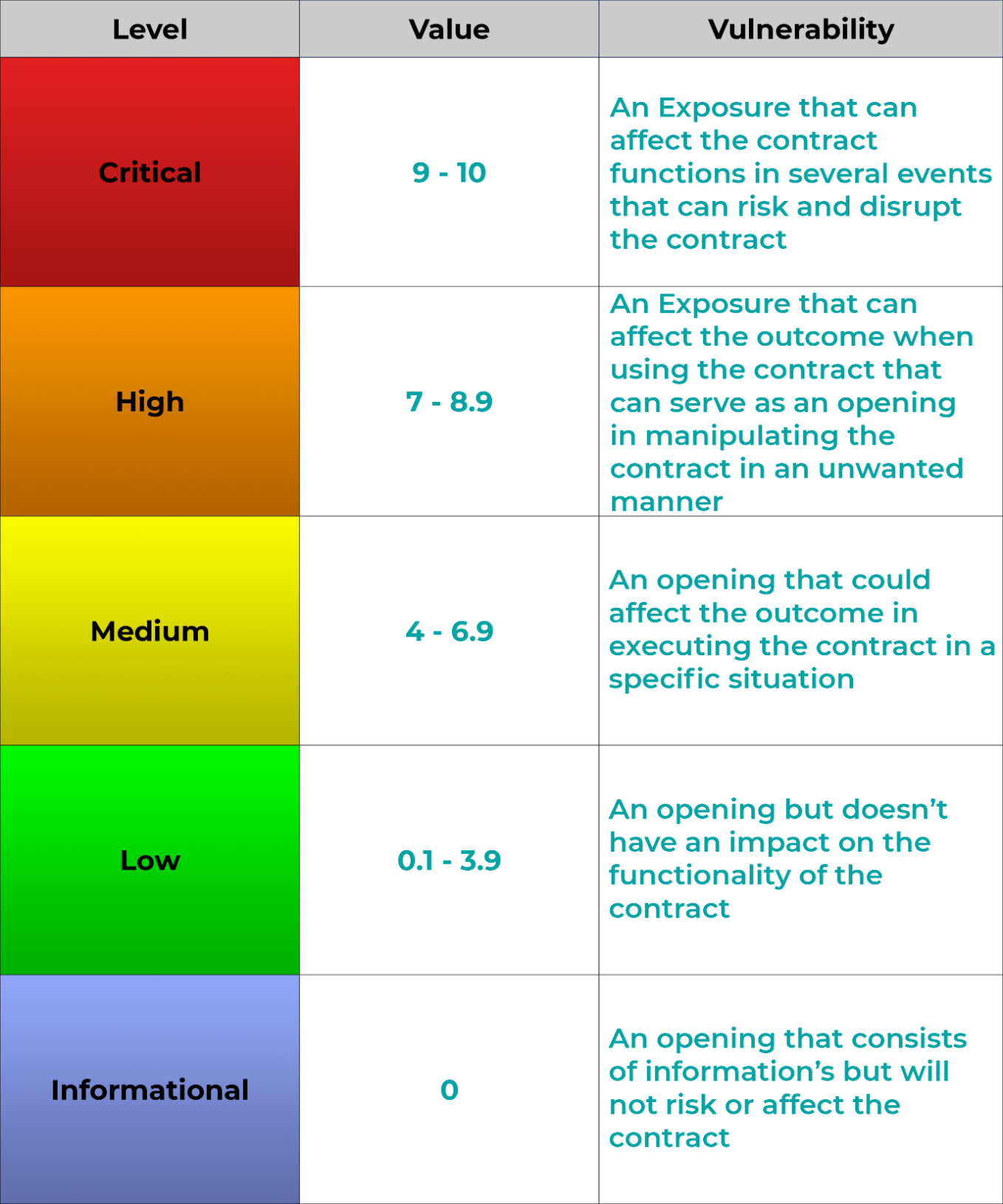
# 

Contract link: https://bscscan.com/address/0x72E8ba612db6ABab6BB11eD77039C71EcB3dDd7c

# Risk Level Classification

Risk Level represents the classification or the probability that a certain function or threat that can exploit vulnerability and have an impact within the system or contract.

Risk Level is computed based on CVSS Version 3.0



# Auditing Approach

Every line of code along with its functionalities will undergo manual review to check its security issues, quality, and contract scope of inheritance. The manual review will be done by our team that will document any issues that there were discovered.

# Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
   * Review of the specifications, sources, and instructions provided to ContractWolf to make sure we understand the size, scope, and functionality of the smart contract.
   * Manual review of code, our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities and security flaws.
2. Testing and automated analysis that includes:
   * Testing the smart contract functions with common test cases and scenarios, to ensure that it returns the expected results.
3. Best practices review, the team will review the contract with the aim to improve efficiency, effectiveness, clarifications, maintainability, security, and control within the smart contract.
4. Recommendations to help the project take steps to secure the smart contract.

# Used Code from other Frameworks/Smart

**Contracts (Direct Imports)**

Imported Packages

* + IBEP20
  + SafeMath
  + Ownable
  + Context
  + IUniswapV2Factory
  + IUniswapV2Pair
  + IUniswapV2Router01
  + IUniswapV2Router02
  + VirtualAds

# Description

Optimization enabled: Yes Version: v0.8.4

Decimal: 18

# Capabilities

**Components**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Contracts** | **Libraries** | **Interfaces** | **Abstract** |
| **1.0** | **1** | **2** | **5** | **2** |

**Exposed Functions**

|  |  |  |
| --- | --- | --- |
| **Version** | **Public** | **Private** |
| **1.0** | **28** | **24** |

|  |  |  |
| --- | --- | --- |
| **Version** | **External** | **Internal** |
| **1.0** | **72** | **25** |

**State Variables**

|  |  |  |
| --- | --- | --- |
| **Version** | **Total** | **Public** |
| **1.0** | **29** | **11** |

**Capabilities**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Solidity Versions**  **Observed** | **Experimental Features** | **Can Receive**  **Funds** | **Uses Assembly** | **Has Destroyable**  **Contracts** |
| **1.0** | **^0.8.4** |  | **Yes** | **Yes (2asm**  **blocks)** | **No** |

# Scope of Work

**Vaquita** team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract.

# Inheritance Graph

**Verify Claims**

**Correct implementation of Token Standard**

|  |  |
| --- | --- |
| **Tested** | **Verified** |
| ✓ | ✘ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Description** | **Exist** | **Tested** | **Verified** |
| TotalSupply | Information about the total  coin or token supply | ✓ | ✓ | ✓ |
| BalanceOf | Details on the account balance from a specified  address | ✓ | ✓ | ✓ |
| Transfer | An action that transfers a specified amount of coin or  token to a specified address | ✓ | ✓ | ✓ |
| TransferFrom | An action that transfers a specified amount of coin or token from a specified  address | ✓ | ✓ | ✓ |
| Approve | Provides permission to withdraw specified number of coin or token from a  specified address | ✓ | ✓ | ✓ |

**Optional implementation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Description** | **Exist** | **Tested** | **Verified** |
| renounceOwnership | Owner renounce ownership for more  trust | ✓ | ✓ | ✓ |

**Deployer cannot mint after initial deployment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Statement** | **Exist** | **Tested** | **Verified** | **File** |
| Deployer cannot mint | ✓ | ✓ | ✓ | Main |

**Deployer cannot block user**

|  |  |  |  |
| --- | --- | --- | --- |
| **Statement** | **Exist** | **Tested** | **Verified** |
| Deployer cannot block  user | ✓ | ✓ | ✓ |

**Deployer cannot burn**

|  |  |  |  |
| --- | --- | --- | --- |
| **Statement** | **Exist** | **Tested** | **Verified** |
| Deployer cannot burn | ✓ | ✓ | ✓ |

**Deployer can pause contract**

|  |  |  |  |
| --- | --- | --- | --- |
| **Statement** | **Exist** | **Tested** | **Verified** |
| Deployer can pause | ✓ | ✓ | ✓ |

# Overall Checkup (Smart Contract Security)

|  |  |
| --- | --- |
| **Tested** | **Verified** |
| ✓ | ✓ |

Legend

|  |  |
| --- | --- |
| **Attribute** | **Symbol** |
| Verified / Checked | ✓ |
| Partly Verified | ✘ |
| Unverified / Not checked | 🏴 |
| Not Available | **–** |

# Graphical user interface Description automatically generatedWrite Functions of Contract

**SWC Attacks**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Title** | **Relationships** | **Status** |
| [SWC-136](https://swcregistry.io/docs/SWC-136) | Unencrypted Private Data  On-Chain | [CWE-767: Access to Critical](https://cwe.mitre.org/data/definitions/767.html) [Private Variable via Public](https://cwe.mitre.org/data/definitions/767.html)  [Method](https://cwe.mitre.org/data/definitions/767.html) | **PASSED** |
| [SWC-135](https://swcregistry.io/docs/SWC-135) | Code With No  Effects | [CWE-1164: Irrelevant Code](https://cwe.mitre.org/data/definitions/1164.html) | **PASSED** |
| [SWC-134](https://swcregistry.io/docs/SWC-134) | Message call with hardcoded gas  amount | [CWE-655: Improper](https://cwe.mitre.org/data/definitions/665.html) [Initialization](https://cwe.mitre.org/data/definitions/665.html) | **PASSED** |
| [SWC-133](https://swcregistry.io/docs/SWC-133) | Hash Collisions with Multiple Variable Length  Arguments | [CWE-294: Authentication](https://cwe.mitre.org/data/definitions/294.html) [Bypass by Capture-replay](https://cwe.mitre.org/data/definitions/294.html) | **PASSED** |
| [SWC-132](https://swcregistry.io/docs/SWC-132) | Unexpected Ether  balance | CWE-667:  Improper Locking | **PASSED** |
| [SWC-131](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-131) | Presence of  unused variables | [CWE-1164: Irrelevant Code](https://cwe.mitre.org/data/definitions/1164.html) | **PASSED** |
| [SWC-130](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-130) | Right-To Left Override control character (U+202E) | [CWE-451: User Interface](http://cwe.mitre.org/data/definitions/451.html) [(UI)](http://cwe.mitre.org/data/definitions/451.html)  [Misrepresentation of](http://cwe.mitre.org/data/definitions/451.html) [Critical](http://cwe.mitre.org/data/definitions/451.html)  [Information](http://cwe.mitre.org/data/definitions/451.html) | **PASSED** |
| [SWC-129](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-129) | Typographical  Error | [CWE-480: Use of Incorrect](https://cwe.mitre.org/data/definitions/480.html)  [Operator](https://cwe.mitre.org/data/definitions/480.html) | **PASSED** |
| [SWC-128](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-128) | DoS With Block  Gas Limit | [CWE-400: Uncontrolled](https://cwe.mitre.org/data/definitions/400.html)  [Resource Consumption](https://cwe.mitre.org/data/definitions/400.html) | **PASSED** |

|  |  |  |  |
| --- | --- | --- | --- |
| [SWC-127](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-127) | Arbitrary Jump with Function  Type Variable | [CWE-695: Use of Low-Level](https://cwe.mitre.org/data/definitions/695.html) [Functionality](https://cwe.mitre.org/data/definitions/695.html) | **PASSED** |
| [SWC-126](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-128) | Insufficient Gas  Griefing | CWE-691: Insufficient  Control Flow Management | **PASSED** |
| [SWC-125](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-125) | Incorrect  Inheritance Order | [CWE-696: Incorrect](https://cwe.mitre.org/data/definitions/696.html)  [Behavior Order](https://cwe.mitre.org/data/definitions/696.html) | **PASSED** |
| [SWC-124](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-124) | Write to Arbitrary  Storage Location | [CWE-123: Write-what-](https://cwe.mitre.org/data/definitions/123.html) [where Condition](https://cwe.mitre.org/data/definitions/123.html) | **PASSED** |
| [SWC-123](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-123) | Requirement Violation | [CWE-573: Improper](https://cwe.mitre.org/data/definitions/573.html) [Following of Specification](https://cwe.mitre.org/data/definitions/573.html)  [by Caller](https://cwe.mitre.org/data/definitions/573.html) | **PASSED** |
| [SWC-122](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-122) | Lack of Proper Signature  Verification | [CWE-345: Insufficient](https://cwe.mitre.org/data/definitions/345.html) [Verification of Data](https://cwe.mitre.org/data/definitions/345.html)  [Authenticity](https://cwe.mitre.org/data/definitions/345.html) | **PASSED** |
| [SWC-121](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-121) | Missing Protection against Signature Replay Attacks | CWE-347: Improper Verification of Cryptographic  Signature | **PASSED** |
| [SWC-120](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-120) | Weak Sources of Randomness from  Chain Attributes | [CWE-330: Use of](https://cwe.mitre.org/data/definitions/330.html) [Insufficiently](https://cwe.mitre.org/data/definitions/330.html)  [Random Values](https://cwe.mitre.org/data/definitions/330.html) | **PASSED** |
| [SWC-119](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-119) | Shadowing State Variables | [CWE-710: Improper](http://cwe.mitre.org/data/definitions/710.html) [Adherence to Coding](http://cwe.mitre.org/data/definitions/710.html)  [Standards](http://cwe.mitre.org/data/definitions/710.html) | **PASSED** |
| [SWC-118](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-118) | Incorrect Constructor Name | [CWE-665: Improper](http://cwe.mitre.org/data/definitions/665.html) I[nitialization](http://cwe.mitre.org/data/definitions/665.html) | **PASSED** |



|  |  |  |  |
| --- | --- | --- | --- |
| [SWC-117](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-117) | Signature Malleability | [CWE-347: Improper](https://cwe.mitre.org/data/definitions/347.html) [Verification of](https://cwe.mitre.org/data/definitions/347.html) [Cryptographic Signature](https://cwe.mitre.org/data/definitions/347.html) | **PASSED** |
| [SWC-116](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-116) | Timestamp Dependence | [CWE-829: Inclusion of](https://cwe.mitre.org/data/definitions/829.html) [Functionality from](https://cwe.mitre.org/data/definitions/829.html)  [Untrusted Control Sphere](https://cwe.mitre.org/data/definitions/829.html) | **PASSED** |
| [SWC-115](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-115) | Authorization  through tx.origin | [CWE-477: Use of Obsolete](https://cwe.mitre.org/data/definitions/477.html)  [Function](https://cwe.mitre.org/data/definitions/477.html) | **PASSED** |
| [SWC-114](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-114) | Transaction Order Dependence | [CWE-362: Concurrent](https://cwe.mitre.org/data/definitions/362.html) [Execution using Shared](https://cwe.mitre.org/data/definitions/362.html) [Resource with Improper](https://cwe.mitre.org/data/definitions/362.html) [Synchronization ('Race](https://cwe.mitre.org/data/definitions/362.html)  [Condition')](https://cwe.mitre.org/data/definitions/362.html) | **PASSED** |
| [SWC-113](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-113) | DoS with Failed Call | [CWE-703: Improper Check](https://cwe.mitre.org/data/definitions/703.html) [or Handling of Exceptional](https://cwe.mitre.org/data/definitions/703.html)  [Conditions](https://cwe.mitre.org/data/definitions/703.html) | **PASSED** |
| [SWC-112](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-112) | Delegate call to Untrusted Callee | [CWE-829: Inclusion of](https://cwe.mitre.org/data/definitions/829.html) [Functionality from](https://cwe.mitre.org/data/definitions/829.html) [Untrusted](https://cwe.mitre.org/data/definitions/829.html)  [Control Sphere](https://cwe.mitre.org/data/definitions/829.html) | **PASSED** |
| [SWC-111](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-110) | Use of Deprecated  Solidity Functions | CWE-477: Use of Obsolete Function | **PASSED** |
| [SWC-110](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-110) | Assert Violation | [CWE-670: Always-](https://cwe.mitre.org/data/definitions/670.html) [Incorrect Control Flow](https://cwe.mitre.org/data/definitions/670.html)  [Implementation](https://cwe.mitre.org/data/definitions/670.html) | **PASSED** |
| [SWC-109](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-109) | Uninitialized  Storage Pointer | [CWE-824: Access of](https://cwe.mitre.org/data/definitions/824.html)  [Uninitialized Pointer](https://cwe.mitre.org/data/definitions/824.html) | **PASSED** |



|  |  |  |  |
| --- | --- | --- | --- |
| [SWC-108](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-108) | State Variable  Default Visibility | [CWE-710: Improper](https://cwe.mitre.org/data/definitions/710.html) [Adherence to Coding](https://cwe.mitre.org/data/definitions/710.html)  [Standards](https://cwe.mitre.org/data/definitions/710.html) | **NOT PASSED** |
| [SWC-107](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-107) | Reentrancy | [CWE-841: Improper](https://cwe.mitre.org/data/definitions/841.html) [Enforcement of Behavioral](https://cwe.mitre.org/data/definitions/841.html)  [Workflow](https://cwe.mitre.org/data/definitions/841.html) | **PASSED** |
| [SWC-106](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-106) | Unprotected SELFDESTRUCT  Instruction | [CWE-284: Improper](https://cwe.mitre.org/data/definitions/284.html) [Access Control](https://cwe.mitre.org/data/definitions/284.html) | **PASSED** |
| [SWC-105](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-105) | Unprotected Ether  Withdrawal | [CWE-284: Improper](https://cwe.mitre.org/data/definitions/284.html) [Access Control](https://cwe.mitre.org/data/definitions/284.html) | **PASSED** |
| [SWC-104](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-104) | Unchecked Call  Return Value | [CWE-252: Unchecked](https://cwe.mitre.org/data/definitions/252.html)  [Return Value](https://cwe.mitre.org/data/definitions/252.html) | **PASSED** |
| [SWC-103](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-103) | Floating Pragma | [CWE-664: Improper](https://cwe.mitre.org/data/definitions/664.html) [Control of a Resource](https://cwe.mitre.org/data/definitions/664.html)  [Through its Lifetime](https://cwe.mitre.org/data/definitions/664.html) | **NOT PASSED** |
| [SWC-102](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-102) | Outdated Compiler Version | [CWE-937: Using](http://cwe.mitre.org/data/definitions/937.html) [Components with Known](http://cwe.mitre.org/data/definitions/937.html)  [Vulnerabilities](http://cwe.mitre.org/data/definitions/937.html) | **PASSED** |
| [SWC-101](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-101) | Integer Overflow  and Underflow | [CWE-682: Incorrect](https://cwe.mitre.org/data/definitions/682.html)  [Calculation](https://cwe.mitre.org/data/definitions/682.html) | **PASSED** |
| [SWC-100](https://smartcontractsecurity.github.io/SWC-registry/docs/SWC-100) | Function Default Visibility | [CWE-710: Improper](https://cwe.mitre.org/data/definitions/710.html) [Adherence to Coding](https://cwe.mitre.org/data/definitions/710.html)  [Standards](https://cwe.mitre.org/data/definitions/710.html) | **PASSED** |



**AUDIT PASSED**

**Low Issues**

|  |  |
| --- | --- |
| A floating pragma is set | L: 7 |
| State variable visibility is not set | L: 462 |

# Audit Comments

* Deployer can renounce ownership
* Deployer can transfer ownership
* Deployer can set update fees/rates with an indefinite amount
* Deployer can set transaction limit
* Deployer can pause/lock contract
* Deployer cannot burn
* Deployer cannot mint after initial deployment



3