InterFi NETWORK

@INTERFINETWORK

SMART CONTRACT SECURITY AUDIT

UNICOIN VESTING CONTRACTS

UNICOIN

SMART CONTRACT AUDIT | TEAM KYC | PROJECT EVALUATION

RELENTLESSLY SECURING THE PUBLIC BLOCKCHAIN | MADE IN CANADA 🇨🇦

Summary

Auditing Firm	InterFi Network
Architecture	InterFi “Echelon” Auditing Standard
Smart Contract Audit Approved By	Chris | Blockchain Specialist at InterFi Network
Project Overview Approved BY	Albert | Project Specialist at InterFi Network
Platform	Solidity
Audit Check (Mandatory)	Static, Software, Auto Intelligent & Manual Analysis
Project Check (Optional)	KYC Analysis
Consultation Request Date	October 07, 2021
Report Date	October 11, 2021

Audit Summary

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

- **UNICOIN'S vesting smart contract source codes have LOW RISK SEVERITY.**
- **UNICOIN has successfully PASSED the smart contract audit.**
- **UNICOIN'S token contract has successfully PASSED the smart contract audit. Check token audit here**
- **UNICOIN has successfully PASSED the owner's KYC verification. Check KYC here**

For a detailed understanding of risk severity, source code vulnerability, and functional test, kindly refer to the audit. At the time of the audit, the token contract is not deployed on any blockchain, the contract can be modified/altered before the deployment.

Table Of Contents

Project Information

InterFi "Echelon" Audit Standard

Smart Contract Risk Assessment

Report Summary

Legal Advisory

Project Overview

InterFi was consulted by UNICOIN on October 07, 2021, to conduct a smart contract security audit of their token source code.

The UNICOIN utility token will facilitate fast, hassle-free and cost-effective cross-network transactions within a single wallet or application without the need for constant network switching. Initially allowing the transfer of UNCOIN between the Binance Smart Chain, the Ethereum (ETH), POLYGON (MATIC) and Bitcoin network.

UNICOIN ensures that users can quickly and cost-effectively move assets from one network to another with just a few clicks in one APP.

Project	UNICOIN
Blockchain	Not Deployed (Binance & Ethereum Smart Chain Planned)
Language	Solidity
Contracts	Not Deployed

Public logo

Solidity Source Code On UNICOIN GitHub

https://github.com/UnicoinOfficial/vesting-contract

Solidity Source Code On InterFi GitHub

https://github.com/interfinetwork/audited-codes/blob/main/unicoinvesting.sol

GitHub Commits

Solidity source code committed at: 48fb176c4b1cf4d8ecfdc25af0ddc1a4a5fa7e86

Files Under Scope (Solidity Multiple Files Format)

- MisBlockBase.sol
- DevelopmentFundContract.sol
- FarmingRewardContract.sol
- InfluencerContract.sol
- ManualBurningContract.sol
- MarketingContract.sol
- PresaleContract.sol
- StakingContract.sol
- TeamVestingContract.sol

Audit Scope & Methodology

The scope of this report is to audit the smart contract source codes of UNICOIN'S Vesting Contracts. The source code can be viewed in its entirety on

https://github.com/interfinetwork/audited-codes/blob/main/unicoinvesting.sol

InterFi has scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category	
Smart Contract Vulnerabilities	❖ Re-entrancy (RE) ❖ Unhandled Exceptions (UE) ❖ Transaction Order Dependency (TO) ❖ Integer Overflow (IO) ❖ Unrestricted Action (UA)
Source Code Review	❖ Ownership Takeover ❖ Gas Limit and Loops ❖ Deployment Consistency ❖ Repository Consistency ❖ Data Consistency ❖ Token Supply Manipulation
Functional Assessment	❖ Access Control and Authorization ❖ Operations Trail and Event Generation ❖ Assets Manipulation ❖ Liquidity Access

InterFi's Echelon Audit Standard

The aim of InterFi's "Echelon" standard is to analyze the smart contract and identify the vulnerabilities and the hacks in the smart contract. Mentioned are the steps used by ECHELON-1 to assess the smart contract:

1. Solidity smart contract source code reviewal:
   - Review of the specifications, sources, and instructions provided to InterFi to make sure we understand the size, scope, and functionality of the smart contract.
   - Manual review of code, which is the process of reading source code line-by-line to identify potential vulnerabilities.
2. Static, Manual, and Automated AI analysis:
   - Test coverage analysis, which is the process of determining whether the test cases are covering the code and how much code is exercised when we run those test cases.
   - Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts

Automated 3P frameworks used to assess the smart contract vulnerabilities

- Slither
- Consensys MythX
- Consensys Surya
- Open Zeppelin Code Analyzer
- Solidity Code Complier

InterFi's Risk Classification

Smart contracts are generally designed to manipulate and hold funds denominated in ETH/BNB. This makes them very tempting attack targets, as a successful attack may allow the attacker to directly steal funds from the contract. Below are the typical risk levels of a smart contract:

Vulnerable: A contract is vulnerable if it has been flagged by a static analysis tool as such. As we will see later, this means that some contracts may be vulnerable because of a false positive.

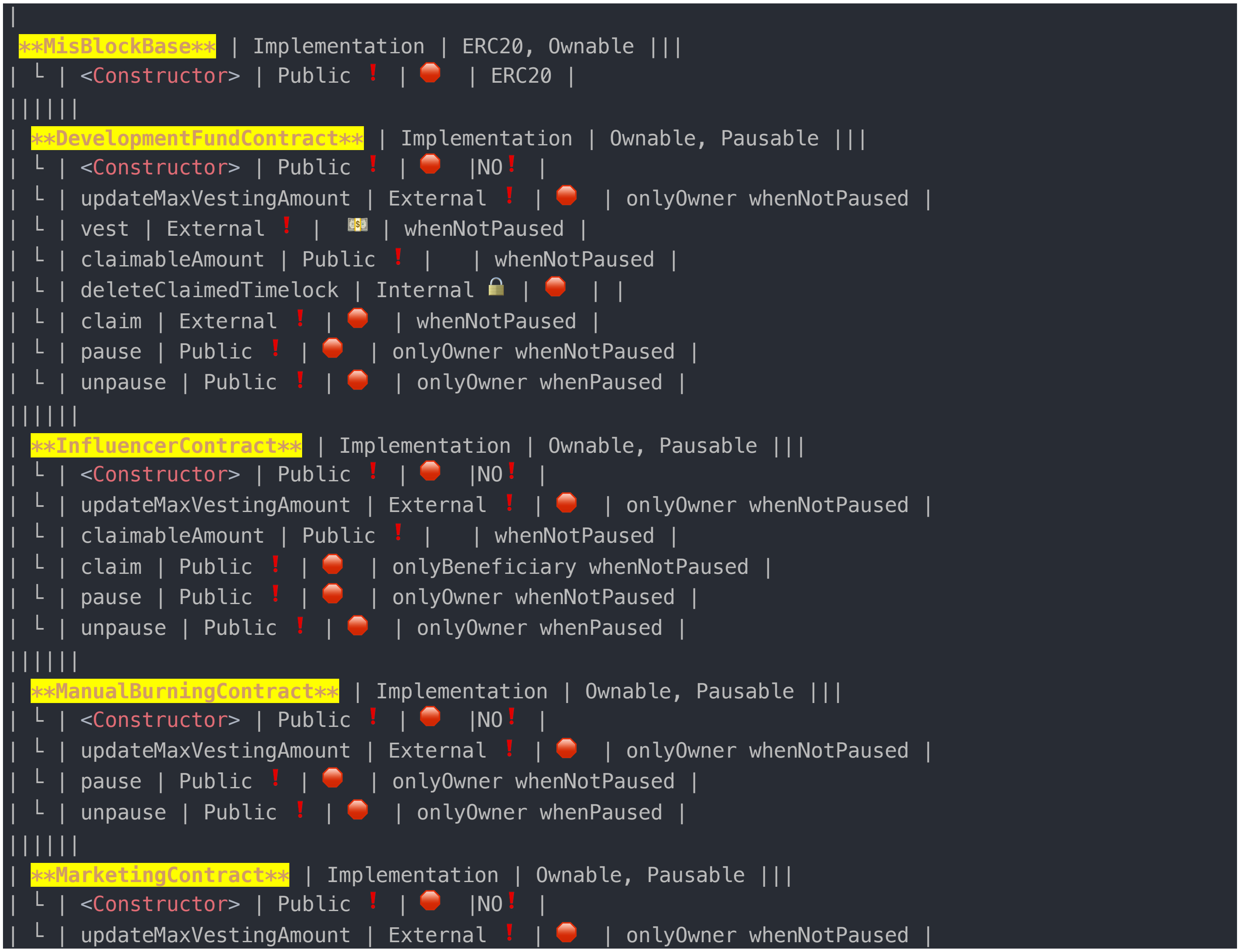
Exploitable: A contract is exploitable if it is vulnerable and the vulnerability could be exploited by an external attacker. For example, if the "vulnerability" flagged by a tool is in a function that requires to own the contract, it would be vulnerable but not exploitable.

Exploited: A contract is exploited if it received a transaction on the main network which triggered one of its vulnerabilities. Therefore, a contract can be vulnerable or even exploitable without having been exploited.

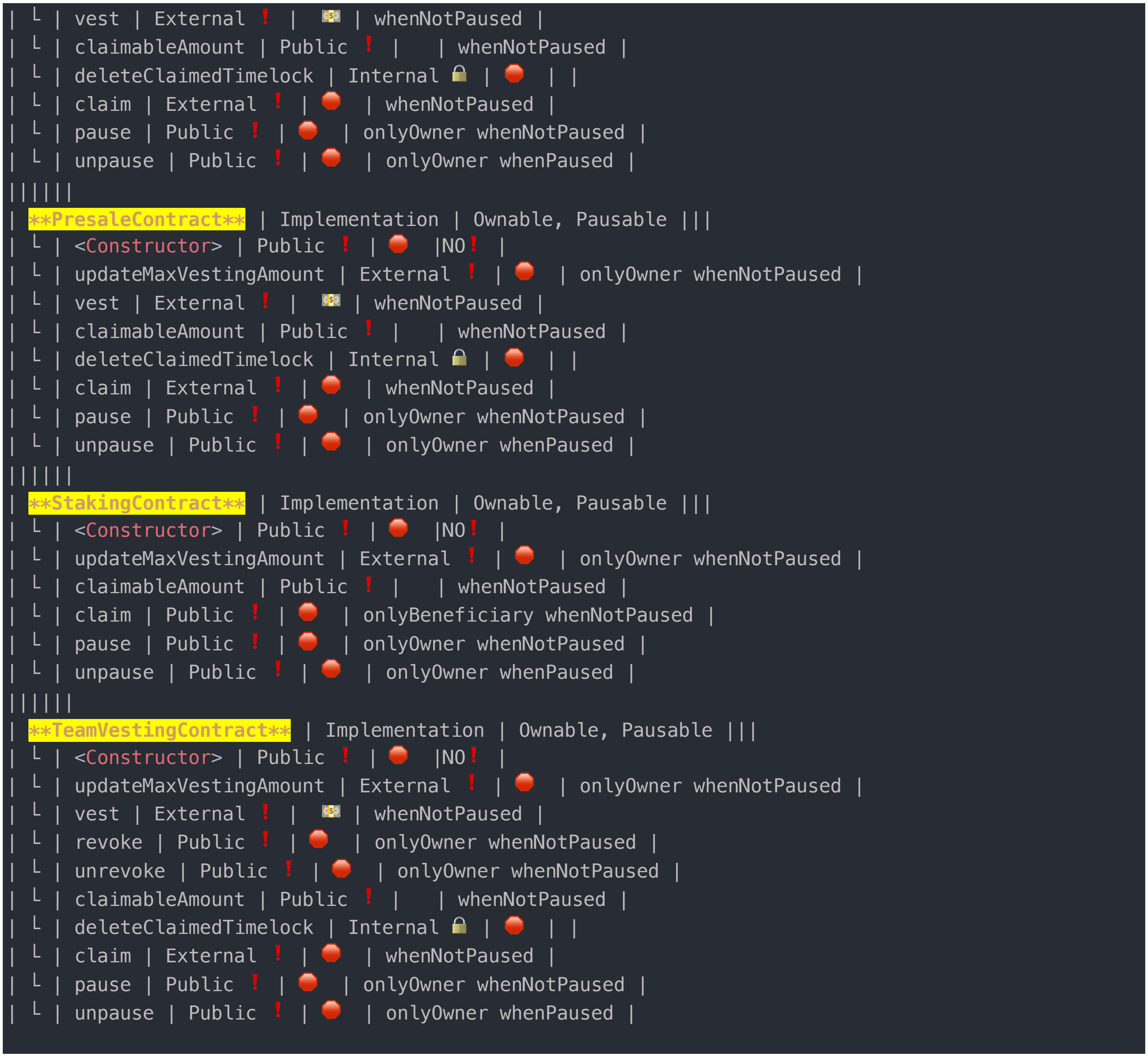
Risk severity	Meaning
! Critical	This level of vulnerability could be exploited easily and can lead to asset loss, data loss, asset manipulation, or data manipulation. They should be fixed right away.
! High	These vulnerabilities are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to critical risk severity
! Medium	These vulnerabilities are should be fixed, as they carry an inherent risk of future exploits, and hacks that may or may not impact the smart contract execution.
! Low	These vulnerabilities can be ignored. They are code style violations, and informational statements in the code. They may not affect the smart contract execution

Smart Contract – Static Analysis

Symbol	Meaning
🛑	Function can be modified
💵	Function is payable
🔒	Function is locked
🔐	Function can be accessed
❗	Important functionality



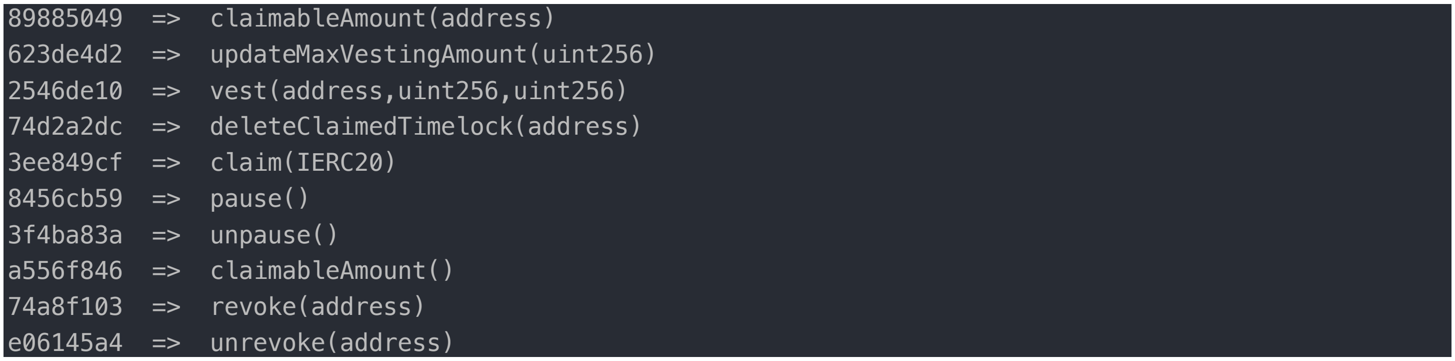
```
|
**MisBlockBase** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗ | 🛑  | ERC20 |
||||||
| **DevelopmentFundContract** | Implementation | Ownable, Pausable |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | updateMaxVestingAmount | External ❗ | 🛑  | onlyOwner whenNotPaused |
| └ | vest | External ❗ |  💵 | whenNotPaused |
| └ | claimableAmount | Public ❗ |   | whenNotPaused |
| └ | deleteClaimedTimelock | Internal 🔒 | 🛑  | |
| └ | claim | External ❗ | 🛑  | whenNotPaused |
| └ | pause | Public ❗ | 🛑  | onlyOwner whenNotPaused |
| └ | unpause | Public ❗ | 🛑  | onlyOwner whenPaused |
||||||
| **InfluencerContract** | Implementation | Ownable, Pausable |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | updateMaxVestingAmount | External ❗ | 🛑  | onlyOwner whenNotPaused |
| └ | claimableAmount | Public ❗ |   | whenNotPaused |
| └ | claim | Public ❗ | 🛑  | onlyBeneficiary whenNotPaused |
| └ | pause | Public ❗ | 🛑  | onlyOwner whenNotPaused |
| └ | unpause | Public ❗ | 🛑  | onlyOwner whenPaused |
||||||
| **ManualBurningContract** | Implementation | Ownable, Pausable |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | updateMaxVestingAmount | External ❗ | 🛑  | onlyOwner whenNotPaused |
| └ | pause | Public ❗ | 🛑  | onlyOwner whenNotPaused |
| └ | unpause | Public ❗ | 🛑  | onlyOwner whenPaused |
||||||
| **MarketingContract** | Implementation | Ownable, Pausable |||
| └ | <Constructor> | Public ❗ | 🛑  |NO❗ |
| └ | updateMaxVestingAmount | External ❗ | 🛑  | onlyOwner whenNotPaused |
```



```
| └ | vest | External ❗️ |  💵 | whenNotPaused |
| └ | claimableAmount | Public ❗️ |   | whenNotPaused |
| └ | deleteClaimedTimelock | Internal 🔒 | 🛑  | |
| └ | claim | External ❗️ | 🛑  | whenNotPaused |
| └ | pause | Public ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | unpause | Public ❗️ | 🛑  | onlyOwner whenPaused |
||||||
| **PresaleContract** | Implementation | Ownable, Pausable |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | updateMaxVestingAmount | External ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | vest | External ❗️ |  💵 | whenNotPaused |
| └ | claimableAmount | Public ❗️ |   | whenNotPaused |
| └ | deleteClaimedTimelock | Internal 🔒 | 🛑  | |
| └ | claim | External ❗️ | 🛑  | whenNotPaused |
| └ | pause | Public ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | unpause | Public ❗️ | 🛑  | onlyOwner whenPaused |
||||||
| **StakingContract** | Implementation | Ownable, Pausable |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | updateMaxVestingAmount | External ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | claimableAmount | Public ❗️ |   | whenNotPaused |
| └ | claim | Public ❗️ | 🛑  | onlyBeneficiary whenNotPaused |
| └ | pause | Public ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | unpause | Public ❗️ | 🛑  | onlyOwner whenPaused |
||||||
| **TeamVestingContract** | Implementation | Ownable, Pausable |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | updateMaxVestingAmount | External ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | vest | External ❗️ |  💵 | whenNotPaused |
| └ | revoke | Public ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | unrevoke | Public ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | claimableAmount | Public ❗️ |   | whenNotPaused |
| └ | deleteClaimedTimelock | Internal 🔒 | 🛑  | |
| └ | claim | External ❗️ | 🛑  | whenNotPaused |
| └ | pause | Public ❗️ | 🛑  | onlyOwner whenNotPaused |
| └ | unpause | Public ❗️ | 🛑  | onlyOwner whenPaused |
```

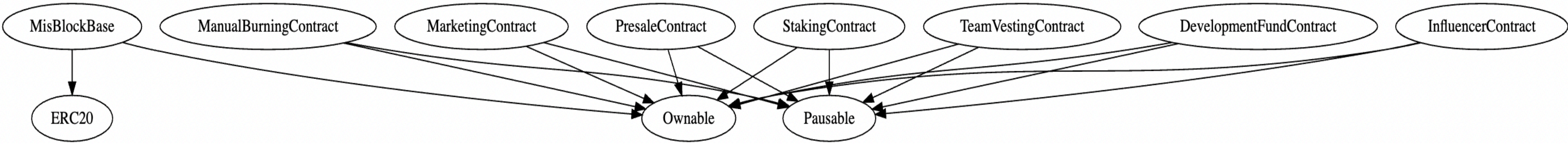
Smart Contract – Software Analysis

Callout function Signatures



```
89885049  =>  claimableAmount(address)
623de4d2  =>  updateMaxVestingAmount(uint256)
2546de10  =>  vest(address,uint256,uint256)
74d2a2dc  =>  deleteClaimedTimelock(address)
3ee849cf  =>  claim(IERC20)
8456cb59  =>  pause()
3f4ba83a  =>  unpause()
a556f846  =>  claimableAmount()
74a8f103  =>  revoke(address)
e06145a4  =>  unrevoke(address)
```

Inheritance Graph



Smart Contract – Manual Analysis

Function	Description	Tested	Verdict
Vest	provides information about the vesting	Yes	**Passed**
Claim	provides balance of claimable tokens	Yes	**Passed**
Pause	pauses the specific function or the contract	Yes	**Passed**
Unpause	unpauses the specific function or the contract	Yes	**Passed**
Revoke	revokes wallet access to the specific function or the contract	Yes	**Passed**
Unrevoke	unrevokes wallet access to the specific function or the contract	Yes	**Passed**

Verified

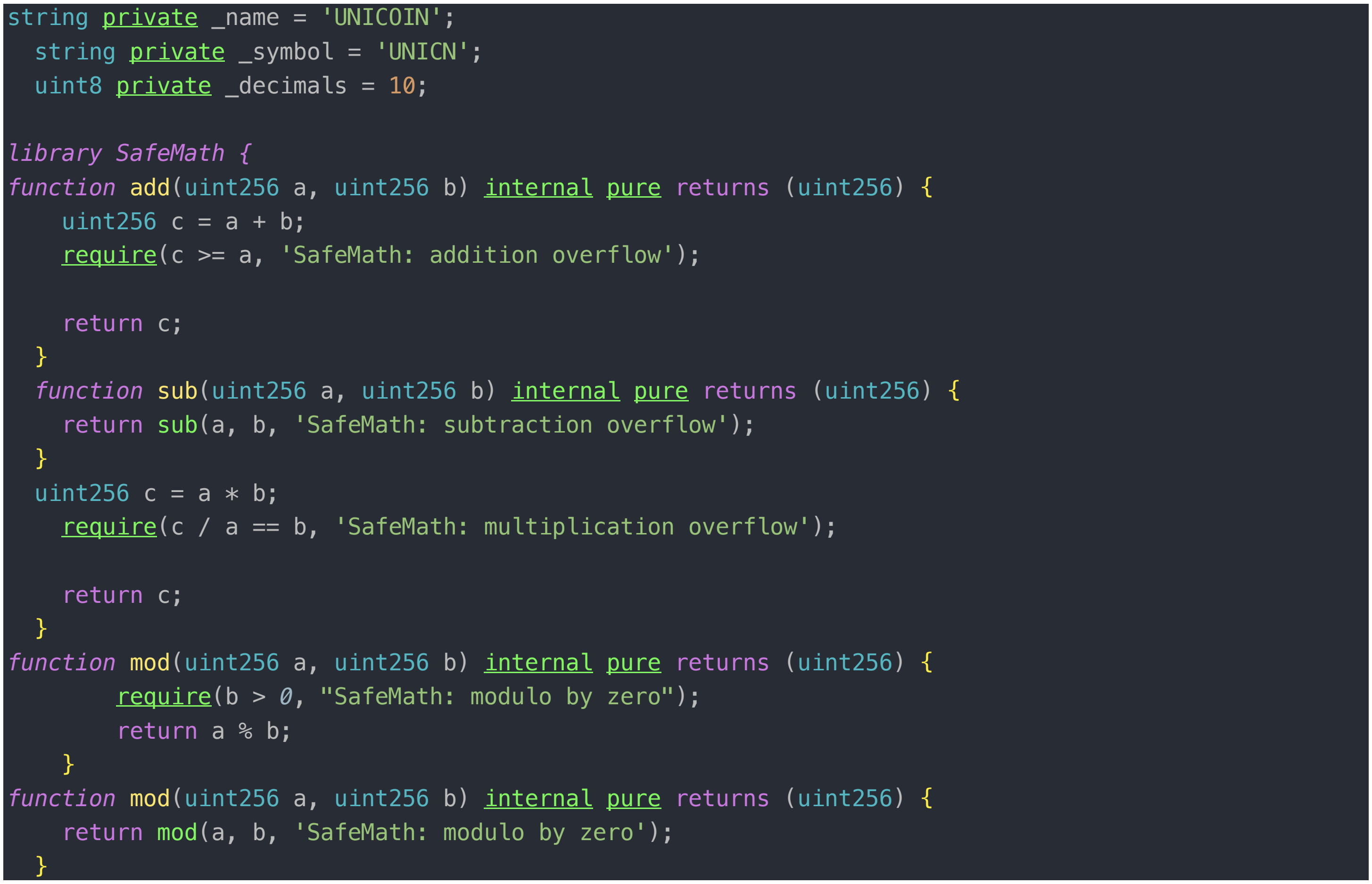
Active smart contract owner privileges constitute an elevated impact to smart contract's safety and security.

- Owner can pause the vesting smart contracts.

At the time of the audit, the vesting contracts are not deployed on any blockchain, the contract can be modified/altered before the deployment.

Important Information

UNICOIN vesting smart contracts utilize the “SafeMath” to prevent known vulnerabilities.



```
string private _name = 'UNICOIN';
  string private _symbol = 'UNICN';
  uint8 private _decimals = 10;

library SafeMath {
function add(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a + b;
    require(c >= a, 'SafeMath: addition overflow');

    return c;
  }
  function sub(uint256 a, uint256 b) internal pure returns (uint256) {
    return sub(a, b, 'SafeMath: subtraction overflow');
  }
  uint256 c = a * b;
    require(c / a == b, 'SafeMath: multiplication overflow');

    return c;
  }
function mod(uint256 a, uint256 b) internal pure returns (uint256) {
        require(b > 0, "SafeMath: modulo by zero");
        return a % b;
    }
function mod(uint256 a, uint256 b) internal pure returns (uint256) {
    return mod(a, b, 'SafeMath: modulo by zero');
  }
```

UNICOIN smart contract has low severity issues which may not create any functional vulnerability.

"Expected identifier, got 'LParen",

Different compliers e.g., pragma solidity 0.8.4; pragma solidity 0.8.2; pragma solidity 0.8.0; are being used in the workspace.

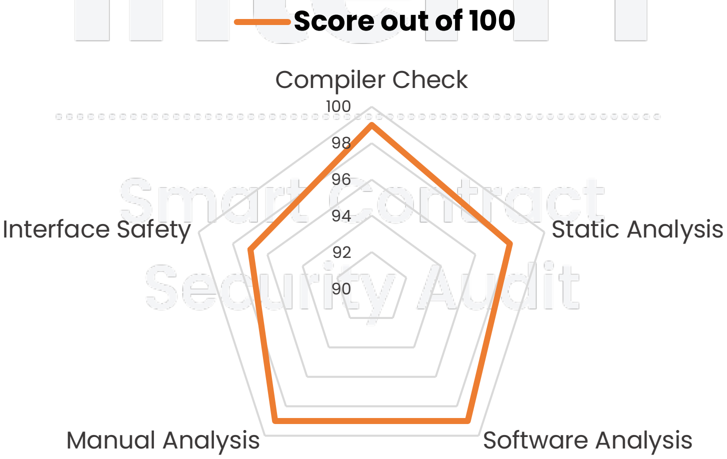
Smart Contract – SWC Attacks

SWC ID	Description	Verdict
SWC-101	Integer Overflow and Underflow	Passed
SWC-102	Outdated Compiler Version	! Low
SWC-103	Floating Pragma	Passed
SWC-104	Unchecked Call Return Value	Passed
SWC-105	Unprotected Ether Withdrawal	Passed
SWC-106	Unprotected SELFDESTRUCT Instruction	Passed
SWC-107	Re-entrancy	Passed
SWC-108	State Variable Default Visibility	Passed
SWC-109	Uninitialized Storage Pointer	Passed
SWC-110	Assert Violation	Passed
SWC-111	Use of Deprecated Solidity Functions	Passed
SWC-112	Delegate Call to Untrusted Callee	Passed
SWC-113	DoS with Failed Call	Passed
SWC-114	Transaction Order Dependence	Passed
SWC-115	Authorization through tx.origin	Passed
SWC-116	Block values as a proxy for time	Passed
SWC-117	Signature Malleability	Passed
SWC-118	Incorrect Constructor Name	Passed

SWC-119	Shadowing State Variables	**Passed**
SWC-120	Weak Sources of Randomness from Chain Attributes	**Passed**
SWC-121	Missing Protection against Signature Replay Attacks	**Passed**
SWC-122	Lack of Proper Signature Verification	**Passed**
SWC-123	Requirement Violation	**Passed**
SWC-124	Write to Arbitrary Storage Location	**Passed**
SWC-125	Incorrect Inheritance Order	**Passed**
SWC-126	Insufficient Gas Griefing	**Passed**
SWC-127	Arbitrary Jump with Function Type Variable	**Passed**
SWC-128	DoS With Block Gas Limit	**Passed**
SWC-129	Typographical Error	**Passed**
SWC-130	Right-To-Left-Override control character (U+202E)	**Passed**
SWC-131	Presence of unused variables	**Passed**
SWC-132	Unexpected Ether balance	**Passed**
SWC-133	Hash Collisions With Multiple Variable Length Arguments	**Passed**
SWC-134	Message call with hardcoded gas amount	**Passed**
SWC-135	Code With No Effects (Irrelevant/Dead Code)	**Passed**
SWC-136	Unencrypted Private Data On-Chain	**Passed**

Smart Contract - Risk Status & Radar Chart

Risk Severity	Status
! Critical	None critical severity issues identified
! High	None high severity issues identified
! Medium	None medium severity issues identified
! Low	**1 Low severity issue identified**
Passed	**41 functions and instances verified and passed**



Compiler Check	99
Static Analysis	98
Software Analysis	99
Manual Analysis	99
Interface Safety	97

Auditor’s Verdict

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks.

UNICOIN’S vesting smart contract source codes have LOW RISK SEVERITY.

UNICOIN has successfully PASSED the smart contract audit.

UNICOIN’S token contract has successfully PASSED the smart contract audit. Check token audit here

UNICOIN has successfully PASSED the owner's KYC verification. Check KYC here

General Note:

- Be aware that active smart contract owner privileges constitute an elevated impact on smart contract safety and security.
- At the time of the audit, the vesting contracts are not deployed on any blockchain, the contract can be modified/altered before the deployment.
- The project’s liquidity pair isn’t checked and verified due to out of scope.
- The project website is not checked due to out of scope. The website hasn’t been reviewed for SSL and lighthouse reports.

Important Disclaimer

InterFi Network provides contract auditing and project verification services for blockchain projects. The purpose of the audit is to analyze the on-chain smart contract source code and to provide a basic overview of the project. **This report should not be transmitted, disclosed, referred to, or relied upon by any person for any purpose without InterFi's prior written consent.**

InterFi provides the easy-to-understand assessment of the project, and the smart contract (otherwise known as the source code). The audit makes no statements or warranties on the security of the code. It also cannot be considered as enough assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have used all the data at our disposal to provide the transparent analysis, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. **Be aware that smart contracts deployed on a blockchain aren't resistant to external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact on smart contract safety and security. Therefore, InterFi does not guarantee the explicit security of the audited smart contract.**

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

This report should not be considered as an endorsement or disapproval of any project or team. The information provided in this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. Do conduct your due diligence and consult your financial advisor before making any investment decisions.

About InterFi Network

InterFi Network provides intelligent blockchain solutions. InterFi is developing an ecosystem that is seamless and responsive. Some of our services: Blockchain Security, Token Launchpad, NFT Marketplace, etc. **InterFi's mission is to interconnect multiple services like Blockchain Security, DeFi, Gaming, and Marketplace under one ecosystem that is seamless, multi-chain compatible, scalable, secure, fast, responsive, and easy to use.**

InterFi is built by a decentralized team of UI experts, contributors, engineers, and enthusiasts from all over the world. Our team currently consists of 6+ core team members, and 10+ casual contributors. **InterFi provides manual, static, and automatic smart contract analysis, to ensure that project is checked against known attacks and potential vulnerabilities.**

To learn more, visit https://interfi.network

To view our audit portfolio, visit https://github.com/interfinetwork

To book an audit, message https://t.me/interfiaudits

@INTERFINETWORK

RELENTLESSLY SECURING THE PUBLIC BLOCKCHAIN | MADE IN CANADA 🇨🇦