ASSURE DEFI®
THE VERIFICATION GOLD STANDARD

Security Assessment:

Guardian Network TOKEN

May 26, 2024

GUARDIAN NETWORK

- Audit Status: **Fail**
- Audit Edition: **Advance**

Risk Analysis

Classifications of Manual Risk Results

Classification	Description
Critical	Danger or Potential Problems.
High	Be Careful or Fail test.
Medium	Pass, Not-Detected or Safe Item.
Low	Function Detected

Manual Code Review Risk Results

Contract Privilege	Description
Buy Tax	35%
Sale Tax	40%
Cannot Buy	Pass
Cannot Sale	Pass
Max Tax	50%
Modify Tax	Yes
Fee Check	Pass
Is Honeypot?	Detected
Trading Cooldown	Not Detected
Can Pause Trade?	Pass
Pause Transfer?	Not Detected
Max Tx?	Fail
Is Anti Whale?	Detected
Is Anti Bot?	Not Detected

Contract Privilege	Description
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Detected
Can Mint?	Pass
Is Proxy?	Not Detected
Can Take Ownership?	Not Detected
Hidden Owner?	Not Detected
Owner	0x
Self Destruct?	Not Detected
External Call?	Detected
Other?	Not Detected
Holders	0
Auditor Confidence	Medium Risk
KYC Present	No
KYC URL	N/A

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Project Overview

Token Summary

Parameter	Result
Address	0x
Name	Guardian Network
Token Tracker	Guardian Network (GRDN)
Decimals	18
Supply	100,000,000
Platform	ETHEREUM
compiler	v0.8.20+commit.a1b79de6
Contract Name	GuardianNetwork
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	
Payment Tx	Corporate

Main Contract Assessed Contract Name

Name	Contract	Live
Guardian Network	0x	Yes

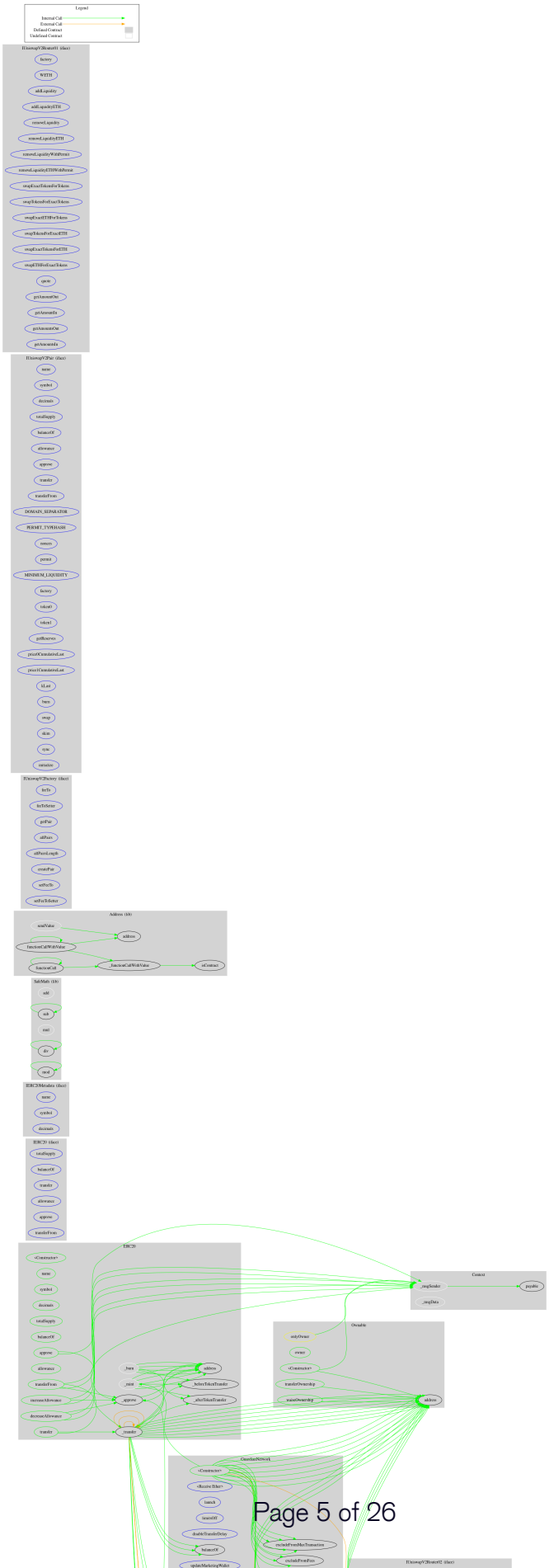
TestNet Contract was Not Assessed

Solidity Code Provided

SolID	File Sha-1	FileName
GC	c65dfc37f208d6638f04d6a5eb1eb300f58548ef	Guardian.sol
GC		
GC		
GC		
GC		
GC		

Call Graph

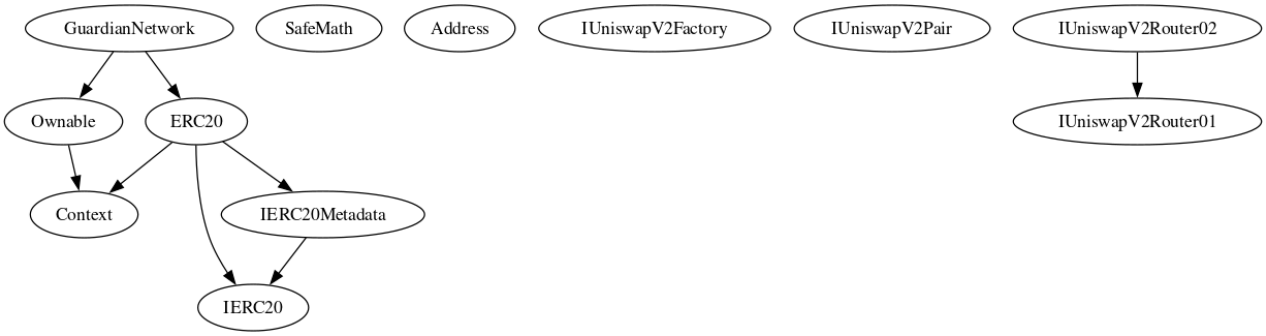
The contract for Guardian Network has the following call graph structure.



Inheritance

The contract for Guardian Network has the following inheritance structure.

The Project has a Total Supply of 100,000,000



Privileged Functions (onlyOwner)

Please Note if the contract is Renounced none of this functions can be executed.

Function Name	Parameters	Visibility
launch		public
limitsOff		public
disableTransferDelay		public
excludeFromMaxTransaction	address updAds, bool isEx	public
excludeFromFees	address account, bool excluded	public
setAutomatedMarketMakerPair	address pair, bool value	public
updateMarketingWallet	address newWallet	public
updateTeamWallet	address newWallet	public
changeSwapTokensAtAmount	uint256 newValue	public
changeMaxTransaction	uint256 newValue	public
changeMaxWallet	uint256 newValue	public
updateSwapEnabled	bool _isEnable	public
updateBuyFee	uint256 _newFee	public
updateSellFee	uint256 _newFee	public

GRDN-01 | Potential Sandwich Attacks.

Category	Severity	Location	Status
Security	Low	Guardian.sol: L: 1102	Unresolved

Description

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (before the transaction being attacked) a transaction to purchase one of the assets and make profits by back running (after the transaction being attacked) a transaction to sell the asset. The following functions are called without setting restrictions on slippage or minimum output amount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

- swapExactTokensForETHSupportingFeeOnTransferTokens()
- addLiquidityETH()

Remediation

We recommend setting reasonable minimum output amounts, instead of 0, based on token prices when calling the aforementioned functions.

Referrences:

What Are Sandwich Attacks in DeFi — and How Can You Avoid Them?.

GRDN-02 | Function Visibility Optimization.

Category	Severity	Location	Status
Gas Optimization	Informational	Guardian.sol: L: 930 C: 14	Detected

Description

The following functions are declared as public and are not invoked in any of the contracts contained within the projects scope:

Function Name	Parameters	Visibility
setAutomatedMarketMakerPair	address pair, bool value	Public

The functions that are never called internally within the contract should have external visibility

Remediation

We advise that the function's visibility specifiers are set to external, and the array-based arguments change their data location from memory to calldata, optimizing the gas cost of the function.

References:

external vs public best practices.

GRDN-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	Low	Guardian.sol: L: 1154 C: 14, L: 952 C: 14, L: 947 C: 14, L: 942 C: 14,L: 925 C: 14, L: 918 C: 14, L: 913 C: 14	Detected

Description

The given input is missing the check for the non-zero address.

The given input is missing the check for the onlyOwners need to be revisited for require..

Remediation

We advise the client to add the check for the passed-in values to prevent unexpected errors as below:

```
...
 require(receiver != address(0), "Receiver is the zero address");
...
...
require(value X limitation, "Your not able to do this function");
...
```

We also recommend customer to review the following function that is missing a required validation. onlyOwners need to be revisited for require..

GRDN-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	Low	Guardian.sol: L: 1163 C: 14, L: 1158 C: 14, L: 1154 C: 14, L: 1145 C: 14, L: 1137 C: 14,L: 1145 C: 14, L: 1120 C: 14, L: 918 C: 14, L: 913 C: 14, L: 907 C: 14, L: 900 C: 14	Detected

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.The linked code does not create an event for the transfer.

Remediation

Emit an event for critical parameter changes. It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

GRDN-08 | Dead Code Elimination.

Category	Severity	Location	Status
Coding Style	Low	Guardian.sol: L: 1082, L: 277, L: 341	Detected

Description

Functions that are not used in the contract, and make the code s size bigger.

SafeMath
Address
min

Remediation

Remove unused functions. dead-code elimination (also known as DCE, dead-code removal, dead-code stripping, or dead-code strip) is a compiler optimization to remove code which does not affect the program results. Removing such code has several benefits: it shrinks program size, an important consideration in some contexts, and it allows the running program to avoid executing irrelevant operations, which reduces its running time. It can also enable further optimizations by simplifying program structure.

https://docs.soliditylang.org/en/latest/cheatsheet.html

GRDN-09 | Third Party Dependencies.

Category	Severity	Location	Status
Volatile Code	High	Guardian.sol: L: 347 C: 14	Detected

Description

The contract is serving as the underlying entity to interact with third party
bytes32 accountHash =
0xc5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470; protocols.
The scope of the audit treats 3rd party entities
as black boxes and assume their functional correctness. However, in the real world, 3rd parties
can be
compromised and this may lead to lost or stolen assets. In addition, upgrades of 3rd parties
can possibly
create severe impacts, such as increasing fees of 3rd parties, migrating to new LP pools, etc.

Remediation

We understand that the business logic of Guardian Network requires interaction with
bytes32 accountHash =
0xc5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470; , etc. We
encourage the team to constantly monitor the
statuses of 3rd parties to mitigate the side effects when unexpected activities are observed.

Project Action

Update Library to latest version.

GRDN-10 | Initial Token Distribution.

Category	Severity	Location	Status
Centralization / Privilege	High	Guardian.sol: L: 894	Detected

Description

All of the Guardian Network tokens are sent to the contract deployer when deploying the contract. This could be a centralization risk as the deployer can distribute tokens without obtaining the consensus of the community.

Remediation

We recommend the team to be transparent regarding the initial token distribution process, and the team shall make enough efforts to restrict the access of the private key.

Project Action

GRDN-11 | Potential Reentrancy in swapBack.

Category	Severity	Location	Status
Optimization	High	Guardian.sol: L: 1102 C: 14	Detected

Description

The swapBack function involves external calls which could be exploited for reentrancy.

Remediation

Use reentrancy guards or check-effects-interactions pattern.

Project Action

GRDN-14 | Unnecessary Use Of SafeMath

Category	Severity	Location	Status
Logical Issue	Medium	Guardian.sol: L: 277 C: 9	Detected

Description

The SafeMath library is used unnecessarily. With Solidity compiler versions 0.8.0 or newer, arithmetic operations
will automatically revert in case of integer overflow or underflow.
library SafeMath {
An implementation of SafeMath library is found.
using SafeMath for uint256;
SafeMath library is used for uint256 type in contract.

Remediation

We advise removing the usage of SafeMath library and using the built-in arithmetic operations provided by the
Solidity programming language

Project Action

GRDN-20 | Transfer Delay Mechanism.

Category	Severity	Location	Status
Optimization	Low	Guardian.sol: L: 989	Detected

Description

The transfer delay mechanism (isTransferDelayActive) can only be disabled and not re-enabled, which might be restrictive.

Remediation

Consider allowing re-enabling of the transfer delay if needed.

Project Action

GRDN-21 | Lack of Fee Cap Enforcement.

Category	Severity	Location	Status
Optimization	Medium	Guardian.sol: L: 1158, L: 1163.	Detected

Description

The contract allows the owner to set fees up to 50% (5000 basis points).

Remediation

Implement a lower maximum fee cap to protect users.

Project Action

Technical Findings Summary

Classification of Risk

Severity	Description
Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
High	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
Low	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
Critical	0	0	0
High	2	2	0
Medium	3	3	0
Low	5	5	0
Informational	1	1	0
Total	11	11	0

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/Guardian_GRDN	Pass
Other		
Website	https://guardiannetwork.io/	Pass
Telegram	https://t.me/Guardian_GRDN	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

Audit Result

Final Audit Score

Review	Score
Security Score	63
Auditor Score	63

The Following Score System Has been Added to this page to help understand the value of the audit, the maximum score is 100, however to attain that value the project most pass and provide all the data needed for the assessment. Our Passing Score has been changed to 85 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below.

Audit Fail

Assessment Results

Important Notes:

- Overall Classification:ı
- Security: Medium Riskı
- Code Quality: Moderateı
- Documentation: Insufficientı
- Key Issues Identified:ı
- Centralization of Control: Owner has significant control over critical functions.ı
- Transfer Delay Mechanism: Can only be disabled, not re-enabled.ı
- Hardcoded Addresses: Uniswap router and initial wallets are hardcoded.ı
- Fee Cap: High maximum fee cap (50%).ı
- Reentrancy Risk: Potential reentrancy in swapBack function.ı
- Event Logging: Insufficient event logging for critical state changes.ı
- Unchecked Arithmetic: Unchecked arithmetic in _transfer function.ı
- Front-Running Risk: Potential for front-running in swapBack.ı
- Validation Checks: Missing zero address validation in updateMarketingWallet.ı

- Use of SafeMath: Redundant in Solidity 0.8.20.ı
- Max Transaction and Wallet Limits: Need careful setting and documentation.ı
- isTradingEnabled Flag: Centralized control and potential misuse.ı
- External OnlyOwner Functions: Missing emit events and validations.ı
- Recommendations:ı
- Implement multi-signature or governance mechanisms.ı
- Allow re-enabling of transfer delay.ı
- Make addresses configurable.ı
- Lower maximum fee cap.ı
- Use reentrancy guards.ı
- Add comprehensive event logging.ı
- Use SafeMath for unchecked arithmetic.ı
- Introduce anti-front-running measures.ı
- Add validation checks for zero addresses.ı
- Remove SafeMath for Solidity 0.8.20.ı
- Document max transaction and wallet limits.ı
- Add emit events and validations for external onlyOwner functions.ı
- Overall Score: 65/100ı
- Summaryı

- The GuardianNetwork contract has several areas for improvement, particularly in security and documentation. Addressing the identified issues will significantly enhance the contract's robustness and user trust.

Auditor Score =63
Audit Fail

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

Assure Defi has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocation for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided ‘as is, and Assure Defi is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will Assure Defi or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by Assure Defi are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.