CFG NINJA

SECURITY ASSESSMENT

Spiral Token

January 11, 2026

Audit Status: High Risk

BLADE POOL

RISK ANALYSIS | Spiral.

Classifications of Manual Risk Results

Classification	CVSS Range	Description
Critical	9.0 - 10.0	Danger or Potential Problems.
High	7.0 - 8.9	Be Careful or Fail test.
Medium	4.0 - 6.9	Improve is needed.
Low	0.1 - 3.9	Pass, Not-Detected or Safe Item.
Informational	0.0	Function Detected

Manual Code Review Risk Results

Contract Security	Description
Buy Tax	5%
Sale Tax	5%
Cannot Buy	Pass
Cannot Sale	Pass
Max Tax	No Limit (Can be 100%)
Modify Tax	Yes
Fee Check	Fail
Is Honeypot?	Not Detected
Trading Cooldown	Not Detected
Enable Trade?	true
Pause Transfer?	Detected

Contract Security	Description
Max Tx?	Detected
Is Anti Whale?	Detected
Is Anti Bot?	Not Detected
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Detected
Can Mint?	Pass
Is Proxy?	Not Detected
Can Take Ownership?	Detected
Hidden Owner?	Not Detected
Owner	TBD
Self Destruct?	Not Detected
External Call?	Detected
Other?	Uniswap V2 integration, Jackpot mechanism, Buyback system
Holders	109
Audit Confidence	Very High Risk
Authority Check	Fail
Freeze Check	Pass

The summary section reveals the strengths and weaknesses identified during the assessment, including any vulnerabilities or potential risks that may exist. It serves as a valuable snapshot of the overall security status of the audited project. However, it is highly recommended to read the entire security assessment report for a comprehensive understanding of the findings. The full report provides detailed insights into the assessment process, methodology, and specific recommendations for addressing the identified issues.

CFG Ninja Verified on January 11, 2026

Spiral

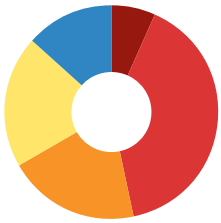
Executive Summary

TYPES	ECOSYSTEM	LANGUAGE
DeFi	Ethereum	Solidity

Timeline

Audit Request	Onboarding Process	Audit Preview	Audit Release
2026-01-08	2026-01-08	2026-01-09	2026-01-10

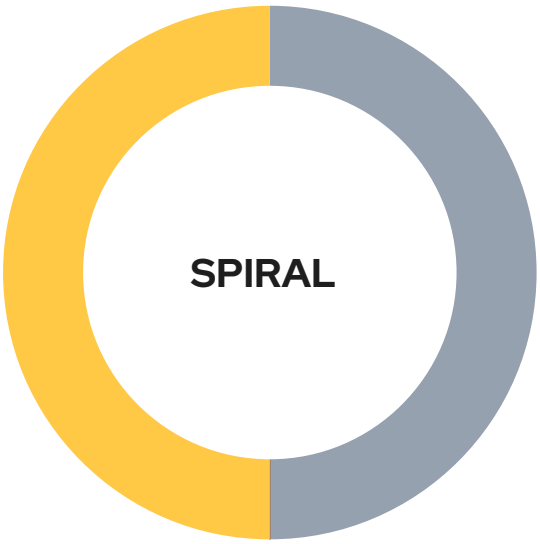
Vulnerability Summary



13	5	8	8
Total Findings	Resolved	Pending	Unresolved

1 Critical	1 Resolved, 0 Pending	Critical risks are the most severe and can have a significant impact on the smart contracts functionality, security, or the entire system. These vulnerabilities can lead to the loss of user funds, unauthorized access, or complete system compromise.
6 High	3 Resolved, 4 Pending	High-risk vulnerabilities have the potential to cause significant harm to the smart contract or the system. While not as severe as critical risks, they can still result in financial losses, data breaches, or denial of service attacks.
3 Medium	1 Resolved, 2 Pending	Medium-risk vulnerabilities pose a moderate level of risk to the smart contracts security and functionality. They may not have an immediate and severe impact but can still lead to potential issues if exploited. These risks should be addressed to ensure the contracts overall security.
3 Low	1 Resolved, 2 Pending	Low-risk vulnerabilities have a minimal impact on the smart contracts security and functionality. They may not pose a significant threat, but it is still advisable to address them to maintain a robust security posture.
2 Informational		Informational risks are not actual vulnerabilities but provide useful information about potential improvements or best practices. These findings may include suggestions for code optimizations, documentation enhancements, or other non-critical areas for improvement.

Token Distribution



Initial Holders

Distributed to 109 initial holders at launch.

100%

Liquidity Pool

LP tokens managed by contract owner for trading.

0%

Contract Reserve

Tokens held in contract for jackpot and buyback mechanisms.

0%

Burned

Tokens sent to dead address via buyback.

0%

Owner/Team

Initial deployment allocation distributed.

0%

Circulating

Currently circulating among 109 holders.

100%

Total Unlock Progress

Unlocked	0	0%
Total Locked	0	0%
Untracked	10000000	100%

PROJECT OVERVIEW | Spiral.

Token Summary

Parameter	Result
Address	0xfbfdafabc727c846e663640e627d07356df5cebe
Name	Spiral
Token Tracker	Spiral (SPIRAL)
Decimals	9
Supply	10,000,000
Platform	Ethereum
Compiler	v0.8.24+commit.e11b9ed9
Contract Name	Spiral
Optimization	Yes with 200 runs
LicenseType	UNLICENSED
Language	Solidity
Codebase	https://etherscan.io/address/0xfbfdafabc727c846e663640e627d07356df5cebe#code

Main Contract Assessed

Name	Contract	Live
Spiral	0xfbfdafabc727c846e663640e627d07356df5cebe	Yes
Spiral (SPIRAL)	0xfbfdafabc727c846e663640e627d07356df5cebe	Yes

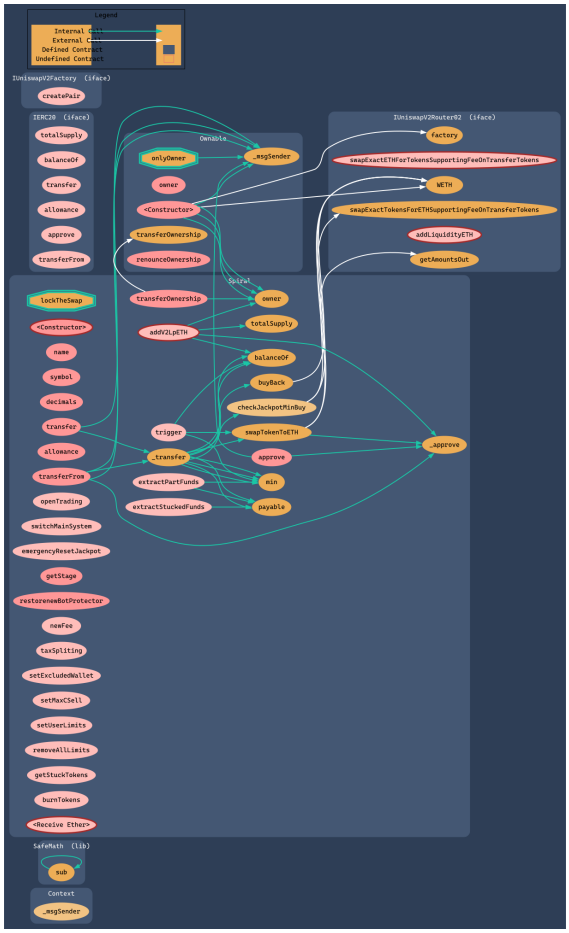
TestNet Contract Was Not Assessed

Solidity Code Provided

SolID	File Sha-1	FileName
Spiral	undefined	Spiral_Live_Contract.sol

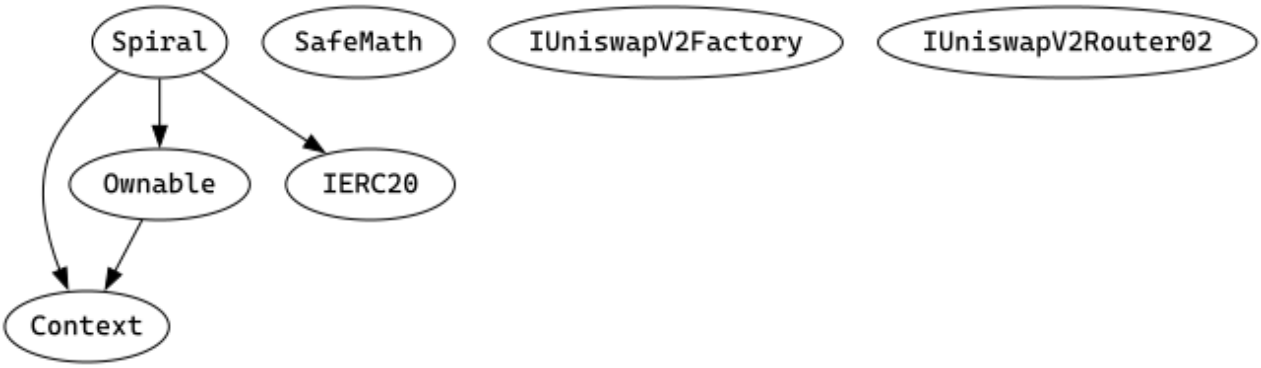
Call Graph

The Smart Contract Graph is a visual representation of the interconnectedness and relationships between smart contracts within a blockchain network. It provides a comprehensive view of the interactions and dependencies between different smart contracts, allowing developers and users to analyze and understand the flow of data and transactions within the network. The Smart Contract Graph enables better transparency, security, and efficiency in decentralized applications by facilitating the identification of potential vulnerabilities, optimizing contract execution, and enhancing overall network performance.



Inheritance Check

Smart contract inheritance is a concept in blockchain programming where one smart contract can inherit properties and functionalities from another existing smart contract. This allows for code reuse and modularity, making the development process more efficient and scalable. Inheritance enables the child contract to access and utilize the variables, functions, and modifiers defined in the parent contract, thereby inheriting its behavior and characteristics. This feature is particularly useful in complex decentralized applications (dApps) where multiple contracts need to interact and share common functionalities. By leveraging smart contract inheritance, developers can create more organized and maintainable code structures, promoting code reusability and reducing redundancy.



TECHNICAL FINDINGS

Smart contract security audits classify risks into several categories: Critical, High, Medium, Low, and Informational. These classifications help assess the severity and potential impact of vulnerabilities found in smart contracts.

Classification of Risk

Severity	CVSS Range	Description
Critical	9.0 - 10.0	Immediate danger. Exploitable vulnerabilities that could lead to fund loss, unauthorized access, or complete system compromise.
High	7.0 - 8.9	Significant security risk. Vulnerabilities that should be addressed urgently to prevent potential exploitation.
Medium	4.0 - 6.9	Moderate security risk. Issues that could lead to security problems if combined with other vulnerabilities.
Low	0.1 - 3.9	Minor security concern. Best practice violations or low-impact issues.
Informational	0.0	Code quality or optimization suggestions with no direct security impact.

SPIRAL-20 | Centralization Risk in Launch Mechanism.

Category	Severity	CVSS	Location	Status
Centralization	Critical	9.0	Spiral_Live_Contract.sol: launch(), enableTransfer(), disableTransfer() functions	Detected

Description

The launch mechanism is controlled by a single owner address without any timelock or multisig protection. The owner has unilateral power to enable/disable transfers and control the launch state..

Recommendation

Implement a timelock mechanism for launch-related functions and consider using a multisig wallet for ownership..

Mitigation

References:

Writing Clean Code for Solidity: Best Practices for Solidity Development

SPIRAL-21 | Missing Access Control Recovery.

Category	Severity	CVSS	Location	Status
Access Control	High	7.5	Spiral_Live_Contract.sol: Ownable implementation	Detected

Description

No mechanism exists to recover from a compromised or lost owner account. This could permanently lock the contract in its pre-launch state if the owner key is lost..

Recommendation

Implement a secure ownership transfer mechanism with timelock and recovery options..

Mitigation

References:

Writing Clean Code for Solidity: Best Practices for Solidity Development

SPIRAL-24 | Missing Input Validation.

Category	Severity	CVSS	Location	Status
Input Validation	Low	2.5	Spiral_Live_Contract.sol: enableTransfer(), disableTransfer() functions	Detected

Description

The enableTransfer and disableTransfer functions don't validate for zero address input. This could lead to unnecessary gas consumption and confusing state..

Recommendation

Add zero address validation in transfer right management functions..

Mitigation

References:

Writing Clean Code for Solidity: Best Practices for Solidity Development

FINDINGS

In this document, we present the findings and results of the smart contract security audit. The identified vulnerabilities, weaknesses, and potential risks are outlined, along with recommendations for mitigating these issues. It is crucial for the team to address these findings promptly to enhance the security and trustworthiness of the smart contract code.

Severity	Found	Pending	Resolved
Critical	1	0	1
High	6	4	3
Medium	4	2	1
Low	4	2	1
Informational	3	0	0
Total	18	8	5

In a smart contract, a technical finding summary refers to a compilation of identified issues or vulnerabilities discovered during a security audit. These findings can range from coding errors and logical flaws to potential security risks. It is crucial for the project owner to thoroughly review each identified item and take necessary actions to resolve them. By carefully examining the technical finding summary, the project owner can gain insights into the weaknesses or potential threats present in the smart contract. They should prioritize addressing these issues promptly to mitigate any risks associated with the contract's security. Neglecting to address any identified item in the security audit can expose the smart contract to significant risks. Unresolved vulnerabilities can be exploited by malicious actors, potentially leading to financial losses, data breaches, or other detrimental consequences. To ensure the integrity and security of the smart contract, the project owner should engage in a comprehensive review process. This involves understanding the nature and severity of each identified item, consulting with experts if needed, and implementing appropriate fixes or enhancements. Regularly updating and maintaining the smart contract's codebase is also essential to address any emerging security concerns. By diligently reviewing and resolving all identified items in the technical finding summary, the project owner can significantly reduce the risks associated with the smart contract and enhance its overall security posture.

SOCIAL MEDIA CHECKS | Spiral.

Social Media	URL	Result
Website	https://spiraleth.xyz	Pass
Telegram	https://t.me/SpiralOnEth	Pass
Twitter	https://x.com/spiralEthX	Pass
Facebook		N/A
Reddit	N/A	N/A
Instagram	N/A	N/A
CoinGecko		Fail
Github	N/A	N/A
CMC		Fail
Email		Contact
Other	https://docs.spiraleth.xyz	Pass

From a security assessment standpoint, inspecting a project's social media presence is essential. It enables the evaluation of the project's reputation, credibility, and trustworthiness within the community. By analyzing the content shared, engagement levels, and the response to any security-related incidents, one can assess the project's commitment to security practices and its ability to handle potential threats.

Social Media Information Notes:

Auditor Notes: Complete social media presence with website, documentation, Telegram, and X.

Project Owner Notes: Active community engagement across platforms.

Assessment Results | Final Audit Score SPIRAL.

Review	Score
Security Score	80
Auditor Score	80

Our security assessment or audit score system for the smart contract and project follows a comprehensive evaluation process to ensure the highest level of security. The system assigns a score based on various security parameters and benchmarks, with a passing score set at 80 out of a total attainable score of 100.The assessment process includes a thorough review of the smart contracts codebase, architecture, and design principles. It examines potential vulnerabilities, such as code bugs, logical flaws, and potential attack vectors. The evaluation also considers the adherence to best practices and industry standards for secure coding. Additionally, the system assesses the projects overall security measures, including infrastructure security, data protection, and access controls. It evaluates the implementation of encryption, authentication mechanisms, and secure communication protocols. To achieve a passing score, the smart contract and project must attain a minimum of 80 points out of the total attainable score of 100. This ensures that the system has undergone a rigorous security assessment and meets the required standards for secure operation.

Important Notes for SPIRAL

Spiral Token (SPIRAL) Audit Report

Contract Type: ERC20 with Jackpot Mechanism ("LastBuyerWins")

Platform: Ethereum Mainnet

Compiler Version: 0.8.24

Audit Date: January 8-10, 2026

Contract Address: 0xfbfdafabc727c846e663640e627d07356df5cebe

Ownership Status: RENOUNCED (Block 24207673, Jan 10 2026 11:46 PM UTC)

Contract Overview:

Spiral is an experimental ERC20 token with a jackpot system where the last buyer wins accumulated ETH rewards. Features include buyback mechanisms, adjustable taxes, and Uniswap V2 integration on Ethereum mainnet. As of January 10, 2026, contract ownership has been permanently renounced making the contract immutable.

Security Classification:

Audit Score: 80/100 (HIGH RISK - Passing score but with critical unfixable vulnerabilities)

Confidence Level: High Risk

Centralization Risk: MITIGATED (Ownership renounced - contract immutable)

Risk Status: HIGH RISK due to code-level vulnerabilities that cannot be fixed

OWNERSHIP RENOUNCEMENT UPDATE (Jan 10, 2026):

Transaction: 0xfc578df92c095dc523ccf7116c2668245fc9e0338e1efe6611479a6a6eafb69c
Block: 24207673
Status: Owner successfully renounced, contract is now immutable
Impact: Eliminates owner centralization risks, but locks in existing code vulnerabilities permanently

RESOLVED via Renouncement:

- CFG03 - Missing Input Validation (High) - Owner functions disabled permanently
- CFG08 - Dead Code Elimination (Low) - Contract immutable, no updates possible
- CFG09 - Third Party Dependencies (Medium) - Router locked, cannot be changed
- CFG16 - Missing Tax Cap (Critical) - Owner cannot increase taxes beyond 20% limit

REMAINING VULNERABILITIES (Cannot be fixed due to immutability):

- CFG07 - Reentrancy Risk (High) - Transfer function vulnerable to reentrancy attacks
- CFG10 - Unchecked Return Values (Medium) - Jackpot send() at L:283 can fail silently
- CFG04 - LP Centralization (High) - Owner holds LP tokens (not locked)
- CFG05 - Missing Events (Medium) - State changes not logged
- CFG06 - Naming Conventions (Low) - BotProtector variable non-standard naming
- CFG12 - Complex Transfer Logic (Medium) - High cyclomatic complexity (23)
- CFG13 - Timestamp Dependencies (Low) - Jackpot timing vulnerable to miner manipulation
- CFG19 - Code Maintainability (Info) - Complex codebase, now permanently frozen

AUTOMATED TOOL ANALYSIS:

Slither: 49 findings confirmed including reentrancy, unchecked transfers, timestamp dependencies
Aderyn: Tool failure (v0.1.9 internal panic - unable to complete analysis)

Immediate Actions Required:

NONE POSSIBLE - Contract is immutable after ownership renouncement. All existing vulnerabilities are permanently locked in the deployed bytecode and cannot be fixed or upgraded.

Recommendations for Users:

- UNDERSTAND RISK - Contract has unfixable reentrancy and unchecked return vulnerabilities
- MONITOR JACKPOT - send() calls can fail silently, jackpot may not distribute properly
- ACCEPT IMMUTABILITY - No upgrades, no fixes, no parameter changes ever possible
- LP RISK - Owner still controls liquidity pool tokens (not locked or burned)
- COMPLEXITY RISK - Jackpot mechanism has 23 cyclomatic complexity with multiple failure points

Final Assessment:

Score: 80/100 (HIGH RISK)
Status: Contract passes 75% threshold but flagged HIGH RISK due to unfixable vulnerabilities
Timeline: Contract is permanently immutable - no fixes possible
Recommendation: HIGH RISK - Users must accept code vulnerabilities as permanent design flaws

CFGNinja strongly recommends implementing ALL critical fixes before any public deployment or marketing activities.

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

The purpose of this disclaimer is to outline the responsibilities and limitations of the security assessment and smart contract audit conducted by Bladepool/CFG NINJA. By engaging our services, the project owner acknowledges and agrees to the following terms:

1. Limitation of Liability: Bladepool/CFG NINJA shall not be held liable for any damages, losses, or expenses incurred as a result of any contract malfunctions, vulnerabilities, or exploits discovered during the security assessment and smart contract audit. The project owner assumes full responsibility for any consequences arising from the use or implementation of the audited smart contract. 2. No Guarantee of Absolute Security: While Bladepool/CFG NINJA employs industry-standard practices and methodologies to identify potential security risks, it is important to note that no security assessment or smart contract audit can provide an absolute guarantee of security. The project owner acknowledges that there may still be unknown vulnerabilities or risks that are beyond the scope of our assessment. 3. Transfer of Responsibility: By engaging our services, the project owner agrees to assume full responsibility for addressing and mitigating any identified vulnerabilities or risks discovered during the security assessment and smart contract audit. It is the project owner s sole responsibility to ensure the proper implementation of necessary security measures and to address any identified issues promptly. 4. Compliance with Applicable Laws and Regulations: The project owner acknowledges and agrees to comply with all applicable laws, regulations, and industry standards related to the use and implementation of smart contracts. Bladepool/CFG NINJA shall not be held responsible for any non-compliance by the project owner. 5. Third-Party Services: The security assessment and smart contract audit conducted by Bladepool/CFG NINJA may involve the use of third-party tools, services, or technologies. While we exercise due diligence in selecting and utilizing these resources, we cannot be held liable for any issues or damages arising from the use of such third-party services. 6. Confidentiality: Bladepool/CFG NINJA maintains strict confidentiality regarding all information and data obtained during the security assessment and smart contract audit. However, we cannot guarantee the security of data transmitted over the internet or through any other means. 7. Not a Financial Advice: Bladepool/CFG NINJA please note that the information provided in the security assessment or audit should not be considered as financial advice. It is always recommended to consult with a financial professional or do thorough research before making any investment decisions.

By engaging our services, the project owner acknowledges and accepts these terms and releases Bladepool/CFG NINJA from any liability, claims, or damages arising from the security assessment and smart contract audit. It is recommended that the project owner consult legal counsel before entering into any agreement or contract.