ASSURE DEFI®
THE VERIFICATION GOLD STANDARD

Security Assessment:

Spark Starter TOKEN

February 11, 2025

- Audit Status: **Pass**
- 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	0%
Sale Tax	0%
Cannot Buy	Pass
Cannot Sale	Pass
Max Tax	4%
Modify Tax	Yes
Fee Check	Pass
Is Honeypot?	Not Detected
Trading Cooldown	Not-Detected
Can Pause Trade?	Pass
Pause Transfer?	Not-Detected
Max Tx?	Pass
Is Anti Whale?	Not-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	No
Self Destruct?	Not Detected
External Call?	Not-Detected
Other?	Not Detected
Holders	1
Auditor Confidence	Medium
KYC Present	Yes
KYC URL	

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

SolID	File Sha-1	FileName
SparkStarter	1bccbe4d2e4eee82e9782cfa430896b587518c35	SparkStarter.sol

Token Summary

Parameter	Result	
SparkStarter Address		.sol
SparkStarter Name	Spark Starter	.sol
SparkStarter Token Tracker	Spark Starter (Spark)	.sol
SparkStarter Decimals	18	.sol
Supply		
Platform	ETHEREUM	
compiler	v0.8.25+commit.b61c2a91	
Contract Name	SparkStarter	
Optimization	Yes with 200 runs	
LicenseType	MIT	
Language	Solidity	
Codebase	https://etherscan.io/address/#code	
Payment Tx	Corporate	

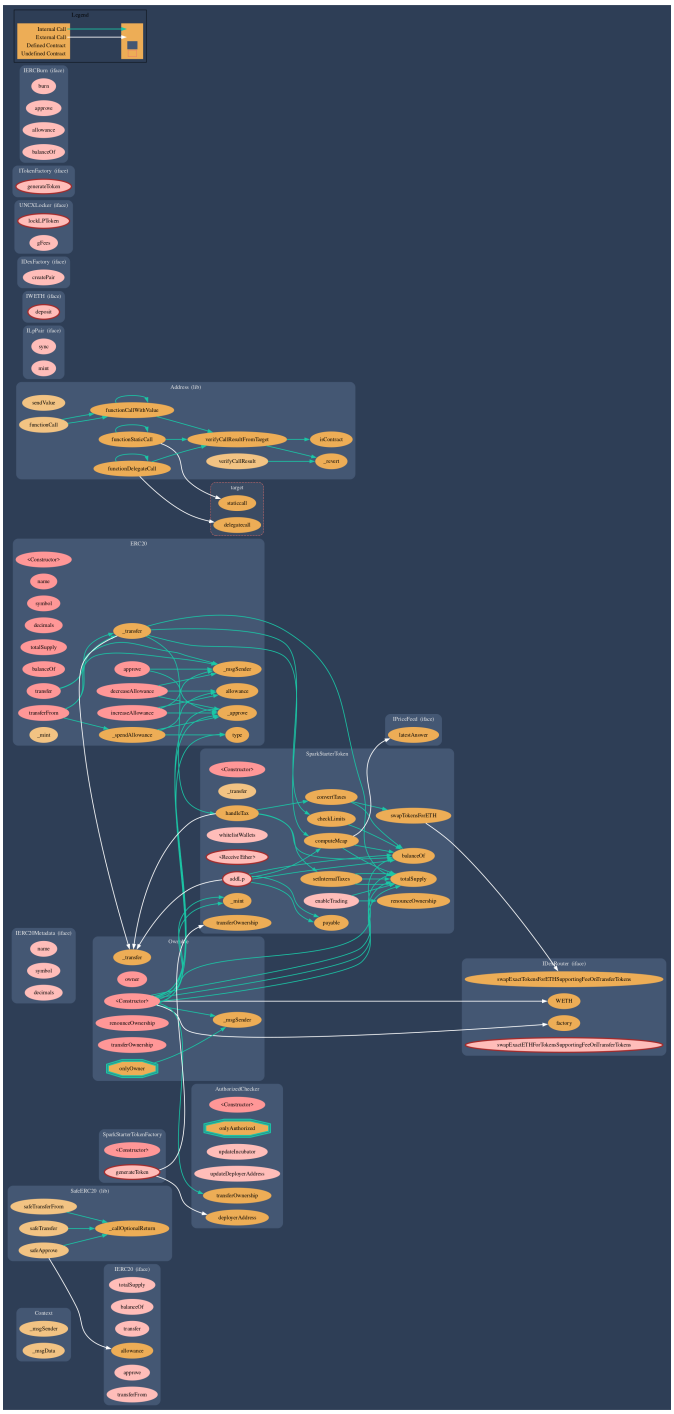
MainNet Contract was Not Assessed

TestNet Contract was Not Assessed

Solidity Code Provided

Call Graph

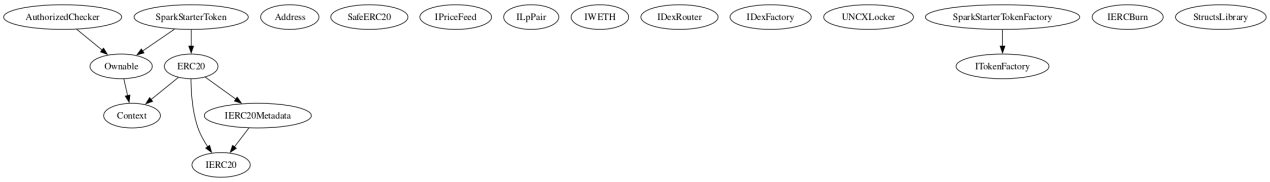
The contract for Spark Starter has the following call graph structure.



Inheritance

The contract for Spark Starter has the following inheritance structure.

The Project has a Total Supply of



Privileged Functions (onlyOwner)

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

Function Name	Parameters	Visibility
renounceOwnership		Public
transferOwnership	address newOwner	Public
enableTrading		External
whitelistWallets		External
addLp		External
updateIncubator		External
updateDeployerAddress		External

Spark-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	Low	SparkStarter.sol: L: 403 C: 12, L: 941 C: 12, L: 976 C: 12, L: 1107 C: 12, L: 1112 C: 12, L: 1116 C: 12	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..

Spark-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	Low	SparkStarter.sol: L: 926 C: 12, L: 941 C: 12, L: 976 C: 12, L: 1107 C: 12, L: 1112 C: 12, L: 1116 C: 12	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.

Spark-09 | Third Party Dependencies.

Category	Severity	Location	Status
Volatile Code	High	SparkStarter.sol: L: 0 C: 0	Detected

Description

The contract is serving as the underlying entity to interact with third party
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 Spark Starter requires interaction with
, 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.

Spark-18 | Stop Transactions by using Enable Trade.

Category	Severity	Location	Status
Logical Issue	Critical	SparkStarter.sol: L: C: 14	Detected

Description

Enable Trade is presend on the following contract and when combined with Exclude from fees it can be considered a whitelist process, this will allow anyone to trade before others and can represent and issue for the holders.

Remediation

We recommend the project owner to carefully review this function and avoid problems when performing both actions.

Project Action

Spark-19 | Centralization Privileges of.

Category	Severity	Location	Status
	Medium	SparkStarter.sol: L: 0 C: 14	Detected

Description

Centralized Privileges are found on the following functions.

Remediation

Inheriting from Ownable and calling its constructor on yours ensures that the address deploying your contract is registered as the owner. The onlyOwner modifier makes a function revert if not called by the address registered as the owner.

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	1	1	0
High	0	1	0
Medium	1	1	0
Low	3	2	0
Informational	0	0	0
Total	5	5	0

Social Media Checks

Social Media	URL	Result
Twitter	https://x.com/sparkstarter_io	Pass
Other	https://whop.com/sparkstarter/	Pass
Website	https://sparkstarter.com/	Pass
Telegram	https://t.me/sparkstarterdeployments	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:

Assessment Results

Score Results

Review	Score
Overall Score	86/100
Auditor Score	85/100
Review by Section	**Score**
Manual Scan Score	31
Auto Scan Score	37
Advance Check Score	18

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 84 Points for a higher standard, if a project does not attain 85% is an automatic failure. Read our notes and final assessment below.

Audit Passed

Assessment Results

Important Notes:

- Contract Structure: Uses multiple interfaces and libraries for modularity. Inherits from ERC20, Ownable, and uses SafeERC20.ı
- Owner Controls: Owner can enable trading, whitelist wallets, and add liquidity. Ownership is renounced after enabling trading, reducing centralization risk.ı
- Dynamic Taxation: Taxes adjust based on time since launch; ensure logic is correct to prevent misuse.ı
- Whitelist Mechanism: Initial whitelist for trading; ensure proper management to prevent unauthorized access.ı
- Liquidity Management: Liquidity is added via addLp function; relies on external locker for locking.ı
- Allowance Management: Implements increaseAllowance and decreaseAllowance to mitigate race conditions.ı
- External Dependencies: Uses UNCXLocker, IPriceFeed, and other interfaces; ensure these are trusted.ı
- Gas Optimization: Consider reviewing for potential gas optimizations, especially in loops and frequent operations.ı
- Security Best Practices: Follows many best practices, but external calls and ETH handling require careful review.ı
- Testing and Verification: Ensure thorough testing, especially around dynamic tax changes and liquidity functions.

Auditor Score =85
Audit Passed

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.