Assure DeFi®

THE VERIFICATION **GOLD STANDARD**

Security Assessment

SparkStarter

Date: 22/02/2025

Audit Status: PASS

Audit Edition: Advanced

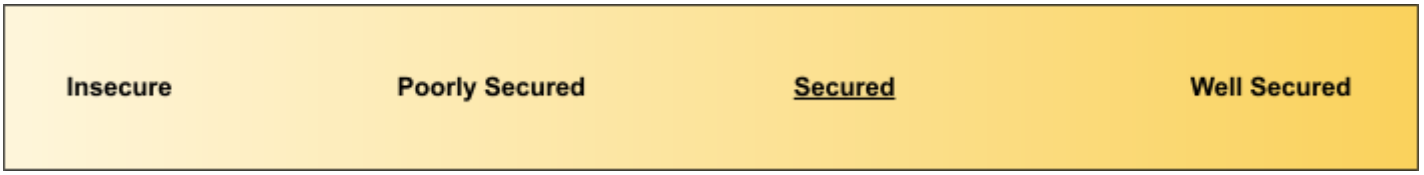
Risk Analysis

Vulnerability summary

Classification	Description
High	High-level vulnerabilities can result in the loss of assets or manipulation of data.
Medium	Medium-level vulnerabilities can be challenging to exploit, but they still have a considerable impact on smart contract execution, such as allowing public access to critical functions.
Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.
Informational	Informational vulnerabilities, code style violations, and informational statements do not affect smart contract execution and can typically be disregarded.

Executive Summary

According to the Assure assessment, the Customer's smart contract is **Secured.**



Scope

Target Code And Revision

For this audit, we performed research, investigation, and review of the SparkStarter contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	SparkStarter.sol [SHA256] 1db10b63a7159f1c12e67c606924194a35419592dd594b67fe5972dd44a9a718 SparkStarterfix.sol [SHA256] e2506462171261070d1df68da3a408ec8345bf7ebc2b2607c86ed2bd72ef775f
Audit Methodology	Static, Manual

Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

Category	Item
Code review & Functional Review	• Compiler warnings. • Race conditions and Reentrancy. Cross-function race conditions. • Possible delays in data delivery. • Oracle calls. • Front running. • Timestamp dependence. • Integer Overflow and Underflow. • DoS with Revert. • DoS with block gas limit. • Methods execution permissions. • Economy model. • Private user data leaks. • Malicious Event log. • Scoping and Declarations. • Uninitialized storage pointers. • Arithmetic accuracy. • Design Logic. • Cross-function race conditions. • Safe Zeppelin module. • Fallback function security. • Overpowered functions / Owner privileges

AUDIT OVERVIEW

HIGH

---

1. Dynamic Tax Misconfiguration Bug [Fixed ✅]

Issue: In the function that updates the tax parameters over time—setInternalTaxes—both the buy and sell taxes are set using the _buyTaxes array from tokenInfo rather than using the corresponding _sellTaxes array for sell transactions. This logic error means that, as time passes, sell tax rates will follow the buy tax schedule rather than their intended schedule.

```
if(timeSinceLaunch >= 15 minutes){
    dynamicTaxOn = false;
    buyTax = tokenInfo._buyTaxes[4];
    sellTax = tokenInfo._buyTaxes[4]; // ❌ Should be tokenInfo._sellTaxes[4]
    maxWallet = uint128(totalSupply());
    limited = false;
} else if(timeSinceLaunch >= 10 minutes){
    buyTax = tokenInfo._buyTaxes[3];
    sellTax = tokenInfo._buyTaxes[3]; // ❌ Should be tokenInfo._sellTaxes[3]
    maxWallet = uint128(totalSupply() * tokenInfo._maxWallets[3] / FEE_DIVISOR);
} else if(timeSinceLaunch >= 5 minutes){
    buyTax = tokenInfo._buyTaxes[2];
    sellTax = tokenInfo._buyTaxes[2]; // ❌ Should be tokenInfo._sellTaxes[2]
    maxWallet = uint128(totalSupply() * tokenInfo._maxWallets[2] / FEE_DIVISOR);
}
```

Recommendation:
Update the setInternalTaxes function so that the sell tax is drawn from the _sellTaxes array.

2. Low-Level ETH Transfers Without Proper Revert Handling [Acknowledge ✅]

Issue: In convertTaxes, after swapping tokens for ETH, the contract sends ETH to various addresses (incubator, platform, tax wallets) using low-level call but only captures the return value in a local variable (success) without checking it. If any of these transfers fail (e.g., because the recipient's fallback function reverts), the failure is silently ignored and may lead to ETH being locked in the contract or funds being misallocated.

Recommendation:

Check the success flag immediately after each low-level call, and revert the transaction if the transfer fails.

3. Liquidity Risk from Contract Minting [Acknowledge ✅]

Issue: When the contract mints a large quantity of tokens to its own address, a situation may arise where the convertTaxes() function calls swapTokensForETH and attempts to swap an amount that exceeds the available liquidity. If insufficient liquidity is available, this swap will revert, potentially disrupting trading when a wallet sends tokens.

Recommendation: Instead of minting the tokens intended for trading directly to the contract address, consider minting them to the owner's address (or a designated distribution wallet).

Alternatively, add pre-swap checks within convertTaxes() to verify that sufficient liquidity exists before attempting the token swap.

4. Potential Reentrancy in Tax Conversion Flow [Acknowledge ✅]

Issue: The function convertTaxes is triggered during a token transfer when the contract's token balance exceeds a threshold. It calls an external DEX router to swap tokens for ETH and then distributes ETH via several low-level calls. No reentrancy guard is present, and although a block check (lastSwapBackBlock) is used to limit multiple swaps per block, the external calls (especially the ETH transfers) could potentially be exploited by a malicious recipient.

Recommendation: Incorporate a reentrancy protection mechanism.

MEDIUM

1. Excessively High Tax Values [Acknowledge ✅]

Issue: The constructor does not enforce an upper bound on the values provided in the _buyTaxes and _sellTaxes arrays. This absence of limits may allow setting extremely high tax rates, which could disrupt normal trading behavior or discourage participation.

Recommendation: Introduce require() statements to validate that each element in the _buyTaxes and _sellTaxes arrays is below a predefined maximum threshold.

2. Use of tx.origin in LP Minting [Fixed ✅]

Issue: In the addLp function, the contract uses tx.origin as the recipient when minting liquidity pool (LP) tokens. The use of tx.origin is generally discouraged because it may lead to unintended behavior when the transaction originates from another contract.

Recommendation: Use msg.sender instead of tx.origin.

LOW

1. Unbounded Array Iteration Risk [Acknowledge ✅]

Issue: The whitelistWallets() function processes an input array of wallet addresses without checking its size. If the array is excessively large, the transaction might run out of gas, causing the function to fail.

Recommendation: Add a require() statement to limit the maximum allowed size of the input array.

INFORMATIONAL

1. Lack of Descriptive Revert Messages for Array Length [Acknowledge ✅]

Issue: Although the constructor checks that the arrays _buyTaxes, _sellTaxes, and _maxWallets have the expected length (5), it currently does not provide clear error messages if these checks fail. This absence makes it more difficult to diagnose issues during deployment.

Recommendation: Enhance each require() check with a descriptive error message.

2. Lack of Event Emission for Dynamic Tax Changes [Acknowledge ✅]

Issue: The setInternalTaxes() function updates tax rates and max wallet limits over time without emitting any events. This lack of on-chain logging makes it harder to track tax parameter changes, reducing transparency.

Recommendation: Emit events whenever tax rates or wallet limits change, e.g., after each update in setInternalTaxes().

3. Redundant ABIEncoderV2 Pragma [Acknowledge ✅]

Issue: The contract includes pragma experimental ABIEncoderV2; even though ABIEncoderV2 is the default and no longer experimental in Solidity 0.8.25.

Recommendation: Remove the redundant experimental pragma to clean up the code and avoid confusion.

Testing coverage

During the testing phase, custom use cases were written to cover all the logic of contracts. *Check “Annexes” to see the testing code.*

SparkStarter contract tests:

```
tests/test_spark_starter.py::test_constructor RUNNING
Transaction sent: 0x39e60746b9c2e540adef0b00ec586757810fa861628f6cfa8aadba71cebe935a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  WETH9.constructor confirmed   Block: 1   Gas used: 476546 (3.97%)
  WETH9 deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x7bae2624e4397b7e8bf657828e1e07435f469ad759872b75091dcfb92e64b42d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  UniswapV2Factory.constructor confirmed   Block: 2   Gas used: 2412730 (20.11%)
  UniswapV2Factory deployed at: 0x602C71e4DAC47a042Ee7f46E0aee17F94A3bA0B6

Transaction sent: 0xfe6097b03e1a1a43331440c45f0d3c463a215b9ce3bb912ceff4521bea2b5fc7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  UniswapV2Router02.constructor confirmed   Block: 3   Gas used: 3895430 (32.46%)
  UniswapV2Router02 deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0xd7a69c87169bc658e56b4a210510d251531482bb153821b8283795c7fc9d08dd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  PriceFeedMock.constructor confirmed   Block: 4   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0x6951b5Bd815043E3F842c1b026b0Fa888Cc2DD85

Transaction sent: 0x2032d2e3331da8548c4e59ae9e9d039c1d8abd0c15392b341b06b4ff051bca3f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  SparkStarterToken.constructor confirmed (Cannot mint 100% to team wallet)   Block: 5   Gas used: 414731 (3.46%)

Transaction sent: 0x144e5e0b28aca8983c587e6dc826f9939cae4e2b9a89ecb148bcd5163c55be50
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  PriceFeedMock.constructor confirmed   Block: 6   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0x6b4BDe1086912A6Cb24ce3dB43b3466e6c72AFd3

Transaction sent: 0xf7bbd8a5fc263f2552c2aa21421d6b9b0d22de1d02e3bd5c0e5999bb6cd3b6f6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  SparkStarterToken.constructor confirmed (Cannot increase buy tax over time)   Block: 7   Gas used: 952021 (7.93%)

Transaction sent: 0x455b6b747c2adbebd544b1ab11e010360bed7867c04048450b933878c3d5c022
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  PriceFeedMock.constructor confirmed   Block: 8   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0xcCB53c9429d32594F404d01fbe9E65ED1DCda8D9

Transaction sent: 0xf0833e9705b0415256ae7cc8ef6b2cd3e9fcf75716b54e41a6cd49cc266aba33
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  SparkStarterToken.constructor confirmed (Cannot increase sell tax over time)   Block: 9   Gas used: 954309 (7.95%)

Transaction sent: 0xa9da7010873db423eaab4fd4c890e3efbfcde33c6f8ae67d8d11cf4c56d6d337
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  PriceFeedMock.constructor confirmed   Block: 10   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0xa3B53dDCd2E3fC28e8E130288F2aBD8d5EE37472

Transaction sent: 0xb668373a0bea35b269d235d10d44e654490d68e672a153d3d1535fc76e200bc8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  SparkStarterToken.constructor confirmed (Cannot decrease max wallet over time)   Block: 11   Gas used: 975622 (8.13%)

Transaction sent: 0x65e26c721cde94f6a5a869c61971041b3ddefbd9ed7c7e28e34358916e0932e3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  PriceFeedMock.constructor confirmed   Block: 12   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0x7a3d735ee6873f17Dbdcab1d51B604928dc10d92

Transaction sent: 0xd3086df7ed83c4598c57fb0feefb8891653d3cb313c83ed3f2c95180ac68c56c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  SparkStarterToken.constructor confirmed (Cannot exceed 100% for tax split)   Block: 13   Gas used: 977620 (8.15%)

Transaction sent: 0xb3c4047d98a2831e7a57b46755b51b7522770b1272f9396965f0661aabe99f7f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  PriceFeedMock.constructor confirmed   Block: 14   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0xe692Cf21B12e0B2717C4bF647F9768Fa58861c8b

Transaction sent: 0x04dc899b53a35d84fbf37f0eb5af2a22ee57de508b8fbed103dd38f10420dd56
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  SparkStarterToken.constructor confirmed   Block: 15   Gas used: 5388293 (44.90%)
  SparkStarterToken deployed at: 0xe65A7a341978d59d40d30FC23F5014FACB4f575A

tests/test_spark_starter.py::test_constructor PASSED
```

```
tests/test_spark_starter.py::test_constructor PASSED
tests/test_spark_starter.py::test_enable_trading RUNNING
Transaction sent: 0x9f469bb019da26fe2815a606cc98ae6582e673979541ae2a20f12d04b104d9f1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  WETH9.constructor confirmed   Block: 16   Gas used: 476546 (3.97%)
  WETH9 deployed at: 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d

Transaction sent: 0x7acf4a9e3edf83daa9409d625c9b50011742893d8fe7efa289c238c0b73a6cd4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  UniswapV2Factory.constructor confirmed   Block: 17   Gas used: 2412730 (20.11%)
  UniswapV2Factory deployed at: 0x26f15335BB1C6a4C0B660eDd694a0555A9F1cce3

Transaction sent: 0xa0d0a9219504d08c048293ea7cc4e6a229a899a6b8e3f5945351863173be3708
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  UniswapV2Router02.constructor confirmed   Block: 18   Gas used: 3895430 (32.46%)
  UniswapV2Router02 deployed at: 0xFbD588c72B438faD4Cf7cD879c8F730Faa213Da0

Transaction sent: 0x21068ecfed01e21a099d521c330f27b8f936151f7cdf2492f3c400d70913a71a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 18
  PriceFeedMock.constructor confirmed   Block: 19   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0xed00238F9A0F7b4d93842033cdF56cCB32C781c2

Transaction sent: 0xedfeaa37d85d3166fdbee11335dd6d11ee7e5306d3c75aeaf40185be7c7b657e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 19
  SparkStarterToken.constructor confirmed   Block: 20   Gas used: 5388293 (44.90%)
  SparkStarterToken deployed at: 0xDae02e4fE488952cFB8c95177154D188647a0146

Transaction sent: 0x77506344f41e8472b90b513e690e05eefe06dfcd8695209e80db9c7f5f278d71
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  SparkStarterToken.enableTrading confirmed (Ownable: caller is not the owner)   Block: 21   Gas used: 22241 (0.19%)

Transaction sent: 0xb2e5c381b755247d545b75296be40712b1478e5c92cdaae938713a62bc9bb378
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 20
  SparkStarterToken.enableTrading confirmed   Block: 22   Gas used: 69277 (0.58%)

Transaction sent: 0x81488970305370033ef26d178a3db4760182eb218337e95f6421f32892ddff36
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  SparkStarterToken.enableTrading confirmed (Trading already enabled)   Block: 23   Gas used: 23049 (0.19%)

tests/test_spark_starter.py::test_enable_trading PASSED
```

```
tests/test_spark_starter.py::test_transfer RUNNING
Transaction sent: 0x2b938135887e99ff66900749a69e12af0f230b2ee8513c292cf92fdcb11533bc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 21
  WETH9.constructor confirmed   Block: 24   Gas used: 476546 (3.97%)
  WETH9 deployed at: 0xBcb61491F1859f53438918F1A5aFCA542Af9D397

Transaction sent: 0x4025830a1bff8e2ffa9c389bbbab94519b0b8cf25f9ce2855f5be0d506ae075c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 22
  UniswapV2Factory.constructor confirmed   Block: 25   Gas used: 2412730 (20.11%)
  UniswapV2Factory deployed at: 0xD22363efee93190f82b52FCD62B7Dbcb920eF658

Transaction sent: 0xd31537db5fa662d7fae898896bbe236df2693a39a72267d029aeafb3318836f7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 23
  UniswapV2Router02.constructor confirmed   Block: 26   Gas used: 3895430 (32.46%)
  UniswapV2Router02 deployed at: 0x4D1B781ce59B8C184F63B99D39d6719A522f46B5

Transaction sent: 0x1ebc5ad7100fc996352b8495fd954d16db5ae52d03b525055674503cea7f0ccd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 24
  PriceFeedMock.constructor confirmed   Block: 27   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0xf9C8Cf55f2E520B08d869df7bc76aa3d3ddDF913

Transaction sent: 0xe6ad1ee42faef7bff47b5ecc1db3e2083819ff9091b7932948e59c7ffd5dc431
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 25
  SparkStarterToken.constructor confirmed   Block: 28   Gas used: 5388295 (44.90%)
  SparkStarterToken deployed at: 0x654f70d8442EA18904FA1AD79114f7250F7E9336

Transaction sent: 0x01d4456aaca688102427557edfd50550cd7173aa09646832f07d7341d041b595
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  SparkStarterToken.transfer confirmed (Trading not active)   Block: 29   Gas used: 24883 (0.21%)

Transaction sent: 0xdf3e55eacd7ef214ac14627a28caca19df7910a45ab582d47a02acf2064875cc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  SparkStarterToken.transfer confirmed   Block: 30   Gas used: 57758 (0.48%)

Transaction sent: 0x893dfd23de87a83f66cb654fff8350ff1d9ele9359b23b77322ba552a5064e4f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 26
  SparkStarterToken.transfer confirmed   Block: 31   Gas used: 56857 (0.47%)

Transaction sent: 0x7c14eee59a3f0b1210f01aba71761f47ed2cece12abb44f72aaa65af4b49ec71
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 27
  SparkStarterToken.enableTrading confirmed   Block: 32   Gas used: 69277 (0.58%)

Transaction sent: 0x8a762064bfd1b6befcffde43eff57f5758e5bc5ebd22b129adc53d59b4492b01
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  SparkStarterToken.transfer confirmed   Block: 33   Gas used: 72902 (0.61%)

Transaction sent: 0xfe8c7a7c7eff67ed5e1b1581b6877a0106fb9e45d05a231831f6ff42cf434e42
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  SparkStarterToken.transfer confirmed   Block: 34   Gas used: 83161 (0.69%)

Transaction sent: 0x563d2cb4be7b47fcea5f6aec85eadb489b641c1249b28fa88c5fecde343f4380
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  SparkStarterToken.transfer confirmed   Block: 35   Gas used: 52649 (0.44%)

tests/test_spark_starter.py::test_transfer PASSED
```

```
tests/test_authorized_checker.py::test_update_incubator RUNNING
Transaction sent: 0x140100e0ce01811985917e4d4bc65bdb16c330c0b99ddde18a54f637f206db87
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  AuthorizedChecker.constructor confirmed   Block: 1   Gas used: 406782 (3.39%)
  AuthorizedChecker deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x1f63911511bd7e82c5d5f565cae155fc49b4f4ff2b06e4bd38c86692c163ddf8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  AuthorizedChecker.updateIncubator confirmed (Ownable: caller is not the owner)   Block: 2   Gas used: 22965 (0.19%)

Transaction sent: 0xc365e07eb835c4fe798eca6cca02cbacce62b3e2d3eeca58055b725db6fa197f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  AuthorizedChecker.updateIncubator confirmed   Block: 3   Gas used: 43772 (0.36%)

tests/test_authorized_checker.py::test_update_incubator PASSED
tests/test_authorized_checker.py::test_update_deployer_addr RUNNING
Transaction sent: 0xbfbcdbc5787fb1c3935abc28c1ac762fa44d9166a94090dfe8cd96b293976ab7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  AuthorizedChecker.constructor confirmed   Block: 4   Gas used: 406782 (3.39%)
  AuthorizedChecker deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0x574da35fddbfc9d28d5af409cf804e6e59ef21573feec2fbea9e929dea78a29f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  AuthorizedChecker.updateDeployerAddress confirmed (Not Authorized)   Block: 5   Gas used: 22928 (0.19%)

Transaction sent: 0xe2a0672e470ddb557cf007942d0e17c2a099385a898a2ba3ee4c202346470d1d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  AuthorizedChecker.updateDeployerAddress confirmed   Block: 6   Gas used: 65601 (0.55%)

Transaction sent: 0x1d69c18069475f14ee23ddc4c56b11ab56d37511c567e59e8a9916d09a4700ff
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  AuthorizedChecker.updateIncubator confirmed   Block: 7   Gas used: 43760 (0.36%)

Transaction sent: 0x3f6634730a844da50ff8c61f573d04335d760a267fedbd3a7abbae709c37a1b6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  AuthorizedChecker.updateDeployerAddress confirmed (reverted)   Block: 8   Gas used: 24635 (0.21%)

tests/test_authorized_checker.py::test_update_deployer_addr PASSED
```

```
tests/test_spark_starter_factory.py::test_generate_token RUNNING
Transaction sent: 0x9bd5dac114ceb6b30cd466a994ba44594068090343b5a1fc5779fb7d4b6b3ac1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  WETH9.constructor confirmed   Block: 9   Gas used: 476546 (3.97%)
  WETH9 deployed at: 0x6b4BDe1086912A6Cb24ce3dB43b3466e6c72AFd3

Transaction sent: 0xb407585473e3a7bb1172c7d8c764a9883cdc97e12f4798d657afd34c07d7a7e4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  UniswapV2Factory.constructor confirmed   Block: 10   Gas used: 2412730 (20.11%)
  UniswapV2Factory deployed at: 0x9E4c14403d7d9A8A782044E86a93CAE09D7B2ac9

Transaction sent: 0x8ad16371ec92e6bb38641ab7caa448e2094a9767d4a9b8415915d3741d91d844
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  UniswapV2Router02.constructor confirmed   Block: 11   Gas used: 3895430 (32.46%)
  UniswapV2Router02 deployed at: 0xcCB53c9429d32594F404d01fbe9E65ED1DCda8D9

Transaction sent: 0x88481cfd5273e0e8569c4783bf542ea7d38beecd9510aa704a8a84899edbb5d1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  PriceFeedMock.constructor confirmed   Block: 12   Gas used: 79123 (0.66%)
  PriceFeedMock deployed at: 0x420b1099B9eF5baba6D92029594eF45E19A04A4A

Transaction sent: 0xda4f9e9415f48b1272dc5805b76a429fe2bce5f5cf2348dc45c548578caceaf9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  AuthorizedChecker.constructor confirmed   Block: 13   Gas used: 406782 (3.39%)
  AuthorizedChecker deployed at: 0xa3B53dDCd2E3fC28e8E130288F2aBD8d5EE37472

Transaction sent: 0x9b5b10702c09db8818fddd3c7bbdab0c1a5fff741995b210ca94b55023fd7ee9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  SparkStarterTokenFactory.constructor confirmed   Block: 14   Gas used: 4313206 (35.94%)
  SparkStarterTokenFactory deployed at: 0xb6286fAFd0451320ad6A8143089b216C2152c025

Transaction sent: 0xfcd3b0315b3c6052ae13874c552459e942c1a98e4513623d84425e1df8518fb6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  SparkStarterTokenFactory.generateToken confirmed (not a valid deployer)   Block: 15   Gas used: 37626 (0.31%)

Transaction sent: 0x811cc24ea3504c2c8f5253291caddle7dd3fd290d9334299a3eea0636a6e7774
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 11
  SparkStarterTokenFactory.generateToken confirmed   Block: 16   Gas used: 5336873 (44.47%)

Transaction sent: 0x6dae4dc088a552d343b300135d5e33dcd4dec5e2421b7c2ed091c4da06acee42
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  SparkStarterTokenFactory.generateToken confirmed (reverted)   Block: 17   Gas used: 5131365 (42.76%)

tests/test_spark_starter_factory.py::test_generate_token PASSED
```

Annexes

Testing code:

SparkStarter:

```
from brownie import (
    reverts,
)


from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    get_account,
    get_timestamp,
    get_chain_number,
    increase_timestamp
)


from scripts.deploy import (
    deploy_weth,
    deploy_factory,
    deploy_router,
    deploy_spark_starter_token,
)


'''


struct TokenInfo {
    string _name;
    string _symbol;
```

```
    uint256 _supply;
    uint256 _teamTokenPercent;
    address _teamTokensWallet;
    uint32[] _maxWallets;
    uint24[] _buyTaxes;
    uint24[] _sellTaxes;
    address _incubatorWallet;
    address _taxWallet1;
    uint24 _taxWallet1Split;
    address _taxWallet2;
    bool _isWhitelistLaunch;
    uint8 lpLockDurationInMonths;
    string jsonPayload;
}
'''


def test_constructor(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    team_wallet = get_account(3)
    incubator_addr = get_account(4)
    tax_addr = get_account(5)
    tax_addr_2 = get_account(6)
    platform_addr = get_account(8)


    weth = deploy_weth(owner)
```

```
    factory = deploy_factory(owner, owner)
    router = deploy_router(owner, factory.address, weth.address)


    params = [
        "token", "tkn", 1000000,
        1000, team_wallet,
        [1000, 1000, 1000, 1000, 1000], # _maxWallets
        [1000, 1000, 1000, 1000, 1000], # buyTaxes
        [1000, 1000, 1000, 1000, 1000], # sellTaxes
        incubator_addr, tax_addr, 1000,
        tax_addr_2, False, 0, 'payload'
    ]


    with reverts("Cannot mint 100% to team wallet"):
        deploy_spark_starter_token(owner,
            [
                "token", "tkn", 1000000,
                10000, team_wallet,
                [1000, 1000, 1000, 1000, 1000], # _maxWallets
                [1000, 1000, 1000, 1000, 1000], # buyTaxes
                [1000, 1000, 1000, 1000, 1000], # sellTaxes
                incubator_addr, tax_addr, 1000,
                tax_addr_2, False, 0, 'payload'
            ],
            platform_addr, router.address)


    with reverts("Cannot increase buy tax over time"):
        deploy_spark_starter_token(owner,
```

```
            [
                "token", "tkn", 1000000,
                1000, team_wallet,
                [1000, 1000, 1000, 1000, 1000], # _maxWallets
                [1000, 1000, 1000, 1000, 10000], # buyTaxes
                [1000, 1000, 1000, 1000, 1000], # sellTaxes
                incubator_addr, tax_addr, 1000,
                tax_addr_2, False, 0, 'payload'
            ],
            platform_addr, router.address)

    with reverts("Cannot increase sell tax over time"):
        deploy_spark_starter_token(owner,
            [
                "token", "tkn", 1000000,
                1000, team_wallet,
                [1000, 1000, 1000, 1000, 1000], # _maxWallets
                [1000, 1000, 1000, 1000, 1000], # buyTaxes
                [1000, 1000, 1000, 1000, 10000], # sellTaxes
                incubator_addr, tax_addr, 1000,
                tax_addr_2, False, 0, 'payload'
            ],
            platform_addr, router.address)

    with reverts("Cannot decrease max wallet over time"):
        deploy_spark_starter_token(owner,
            [
                "token", "tkn", 1000000,
```

```
                1000, team_wallet,
                [1000, 10000, 1000, 1000, 1000], # _maxWallets
                [1000, 1000, 1000, 1000, 1000], # buyTaxes
                [1000, 1000, 1000, 1000, 1000], # sellTaxes
                incubator_addr, tax_addr, 1000,
                tax_addr_2, False, 0, 'payload'
            ],
            platform_addr, router.address)

    with reverts("Cannot exceed 100% for tax split"):
        deploy_spark_starter_token(owner,
            [
                "token", "tkn", 1000000,
                1000, team_wallet,
                [1000, 1000, 1000, 1000, 1000], # _maxWallets
                [1000, 1000, 1000, 1000, 1000], # buyTaxes
                [1000, 1000, 1000, 1000, 1000], # sellTaxes
                incubator_addr, tax_addr, 100000,
                tax_addr_2, False, 0, 'payload'
            ],
            platform_addr, router.address)

    token = deploy_spark_starter_token(owner, params, platform_addr, router.address)

def test_enable_trading(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
```

```
    extra = get_account(2)
    team_wallet = get_account(3)
    incubator_addr = get_account(4)
    tax_addr = get_account(5)
    tax_addr_2 = get_account(6)
    platform_addr = get_account(8)


    weth = deploy_weth(owner)
    factory = deploy_factory(owner, owner)
    router = deploy_router(owner, factory.address, weth.address)


    params = [
        "token", "tkn", 1000000,
        1000, team_wallet,
        [1000, 1000, 1000, 1000, 1000], # _maxWallets
        [1000, 1000, 1000, 1000, 1000], # buyTaxes
        [1000, 1000, 1000, 1000, 1000], # sellTaxes
        incubator_addr, tax_addr, 1000,
        tax_addr_2, False, 0, 'payload'
    ]


    token = deploy_spark_starter_token(owner, params, platform_addr, router.address)
    with reverts("Ownable: caller is not the owner"):
        token.enableTrading({"from": other})


    tx = token.enableTrading({"from": owner})
    assert tx.events['OwnershipTransferred'][0]['previousOwner'] == owner
    assert tx.events['OwnershipTransferred'][0]['newOwner'] == ZERO_ADDRESS
```

```
    with reverts("Trading already enabled"):
        token.enableTrading({"from": ZERO_ADDRESS})


def test_transfer(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    team_wallet = get_account(3)
    incubator_addr = get_account(4)
    tax_addr = get_account(5)
    tax_addr_2 = get_account(6)
    platform_addr = get_account(8)

    weth = deploy_weth(owner)
    factory = deploy_factory(owner, owner)
    router = deploy_router(owner, factory.address, weth.address)

    params = [
        "token", "tkn", 1000000,
        1000, team_wallet,
        [1000, 1000, 1000, 1000, 1000], # _maxWallets
        [1000, 1000, 1000, 1000, 1000], # buyTaxes
        [1000, 1000, 1000, 1000, 1000], # sellTaxes
        incubator_addr, tax_addr, 1000,
        tax_addr_2, False, 0, 'payload'
    ]
```

```
token = deploy_spark_starter_token(owner, params, platform_addr, router.address)

with reverts("Trading not active"):

    token.transfer(other, 5e18, {"from": team_wallet})


tx = token.transfer(owner, 15e18, {"from": team_wallet})

assert tx.events['Transfer'][0]['from'] == team_wallet

assert tx.events['Transfer'][0]['to'] == owner

assert tx.events['Transfer'][0]['value'] == 15e18


tx = token.transfer(other, 5e18, {"from": owner})

assert tx.events['Transfer'][0]['from'] == owner

assert tx.events['Transfer'][0]['to'] == other

assert tx.events['Transfer'][0]['value'] == 5e18


token.enableTrading({"from": owner})


tx = token.transfer(extra, 1e18, {"from": other})

assert tx.events['Transfer'][0]['from'] == other

assert tx.events['Transfer'][0]['to'] == extra

assert tx.events['Transfer'][0]['value'] == 1e18


lp_addr = token.lpPair()

tx = token.transfer(lp_addr, 1e18, {"from": other})

assert tx.events['Transfer'][0]['from'] == other

assert tx.events['Transfer'][0]['to'] == token.address

assert tx.events['Transfer'][0]['value'] == 0.1e18

assert tx.events['Transfer'][1]['from'] == other
```

```
    assert tx.events['Transfer'][1]['to'] == lp_addr

    assert tx.events['Transfer'][1]['value'] == 0.9e18


    tx = token.transfer(extra, 0.9e18, {"from": lp_addr})

    assert tx.events['Transfer'][0]['from'] == lp_addr

    assert tx.events['Transfer'][0]['to'] == token.address

    assert tx.events['Transfer'][0]['value'] == 0.09e18

    assert tx.events['Transfer'][1]['from'] == lp_addr

    assert tx.events['Transfer'][1]['to'] == extra

    assert tx.events['Transfer'][1]['value'] == 0.81e18
```

Spark Starter Factory:

```
from brownie import (

    reverts,

)


from scripts.helpful_scripts import (

    ZERO_ADDRESS,

    get_account,

    get_timestamp,

    get_chain_number,

    increase_timestamp

)


from scripts.deploy import (

    deploy_weth,

    deploy_factory,
```

```
    deploy_router,
    deploy_price_feed,
    deploy_authorized_checker,
    deploy_spark_starter_factory
)

'''

struct TokenInfo {
    string _name;
    string _symbol;
    uint256 _supply;
    uint256 _teamTokenPercent;
    address _teamTokensWallet;
    uint32[] _maxWallets;
    uint24[] _buyTaxes;
    uint24[] _sellTaxes;
    address _incubatorWallet;
    address _taxWallet1;
    uint24 _taxWallet1Split;
    address _taxWallet2;
    bool _isWhitelistLaunch;
    uint8 lpLockDurationInMonths;
    string jsonPayload;
}
'''

def test_generate_token(only_local):
```

```
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    team_wallet = get_account(3)
    incubator_addr = get_account(4)
    tax_addr = get_account(5)
    tax_addr_2 = get_account(6)
    platform_addr = get_account(8)


    weth = deploy_weth(owner)
    factory = deploy_factory(owner, owner)
    router = deploy_router(owner, factory.address, weth.address)
    mock_price_feed = deploy_price_feed(owner)
    checker = deploy_authorized_checker(owner)
    factory = deploy_spark_starter_factory(owner, platform_addr, checker.address)



    params = [
        "token", "tkn", 1000000,
        1000, team_wallet,
        [1000, 1000, 1000, 1000, 1000], # _maxWallets
        [1000, 1000, 1000, 1000, 1000], # buyTaxes
        [1000, 1000, 1000, 1000, 1000], # sellTaxes
        incubator_addr, tax_addr, 1000,
        tax_addr_2, False, 0, 'payload'
    ]
```

```
    with reverts("not a valid deployer"):
        factory.generateToken(
            params, router.address, mock_price_feed.address, {"from": other})

    tx = factory.generateToken(
        params, router.address, mock_price_feed.address, {"from": owner, "value": 1e18})
    assert tx.events['OwnershipTransferred'][0]['previousOwner'] == ZERO_ADDRESS
    assert tx.events['OwnershipTransferred'][0]['newOwner'] == factory.address
    assert tx.events['OwnershipTransferred'][1]['previousOwner'] == factory.address
    assert tx.events['OwnershipTransferred'][1]['newOwner'] == owner
    assert tx.events['NewTokenCreated'][0]['newToken'] is not None

    with reverts(): # no value
        factory.generateToken(
            params, router.address, mock_price_feed.address, {"from": owner})
```

Authorized Checker:

```
from brownie import (
    reverts,
)

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    DAY_TIMESTAMP,
    get_account,
    get_timestamp,
    get_chain_number,
```

```
    increase_timestamp
)


from scripts.deploy import (
    deploy_authorized_checker
)


def test_update_incubator(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    checker = deploy_authorized_checker(owner)
    with reverts("Ownable: caller is not the owner"):
        checker.updateIncubator(other, True, {"from": other})

    assert checker.incubatorAddress(other) == False
    checker.updateIncubator(other, True, {"from": owner})
    assert checker.incubatorAddress(other) == True


def test_update_deployer_addr(only_local):
    # Arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)

    checker = deploy_authorized_checker(owner)
```

```
    with reverts("Not Authorized"):

        checker.updateDeployerAddress(other, True, {"from": other})


    assert checker.deployersIncubatorAddress(other) == ZERO_ADDRESS

    assert checker.deployerAddress(other) == False

    checker.updateDeployerAddress(other, True, {"from": owner})

    assert checker.deployersIncubatorAddress(other) == owner

    assert checker.deployerAddress(other) == True


    checker.updateIncubator(extra, True, {"from": owner})

    with reverts():

        checker.updateDeployerAddress(other, False, {"from": extra})
```

Technical Findings Summary

Findings

Vulnerability Level	Total	Pending	Not Apply	Acknowledged	Partially Fixed	Fixed
High	4			3		1
Medium	2			1		1
Low	1			1		
Informational	3			3		

Assessment Results

Score Results

Review	Score
Global Score	**90/100**
Assure KYC	Not completed
Audit Score	90/100

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 must 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. The Global Score is a combination of the evaluations obtained between having or not having KYC and the type of contract audited together with its manual audit.

Audit PASS

Following our comprehensive security audit of the token contract for the SparkStarter project, the project did meet the necessary criteria required to pass the security audit.

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 advocating for the Project, and users relying on this report should not consider this as having any merit for financial adSparkStarter 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 adSparkStarter, nor should it be used to signal that any person 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 audit 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 serSparkStarters 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 serSparkStarters, 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 serSparkStarters may access, and depend upon, multiple layers of third parties.