

Assure DeFi[®]

THE VERIFICATION **GOLD STANDARD**



Security Assessment

Drops

Date: 03/04/2024

Audit Status: PASS

Audit Edition: Advanced



ASSURE DEFI[®]
THE VERIFICATION **GOLD STANDARD**

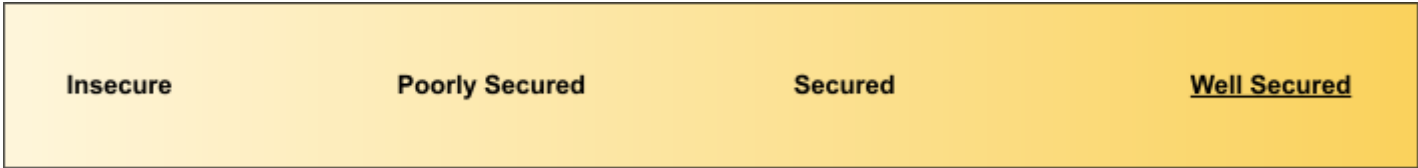
Risk Analysis

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 Low	Low-level vulnerabilities are primarily associated with outdated or unused code snippets that generally do not significantly impact execution, sometimes they can be ignored.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 **Well Secured**.



Scope

Target Code And Revision

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

Target Code And Revision

Project	Assure
Language	Solidity
Codebase	https://github.com/DropsERC/Drops/blob/main/contracts/UNCXMarketplace/DropsUnicryptMarketplace.sol - [Commit] 21f148879b2f4766850895dc316ac6e6c27d60c C Deployed at [Eth Network] 0x76389Ea0Ce243b3404f692De4856A40C0BcB31A7 https://github.com/DropsERC/Drops/tree/main/contracts/PinkLocker - [Commit] 21f148879b2f4766850895dc316ac6e6c27d60c C Deployed at [BSC Network] https://bscscan.com/address/0xA562912e1328eEA987E04c2650EfB5703757850C#code https://github.com/DropsERC/Drops/tree/main/contracts/BSC - [Commit] 21f148879b2f4766850895dc316ac6e6c27d60c C
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	<ul style="list-style-type: none">• 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



1. Incorrect identifier Case Sensitivity in DropsPinksaleMarketplace Functions [FIXED ✓]

Function: activateListing, withdrawListing

Issue: In both functions within the DropsPinksaleMarketplace, the identifier 'lockId' is used when calling pinkLock.getLockById(lockId). However, the correct identifier should be 'lockID' that currently is causing an incorrect contract behaviour.

Fix: Update the identifier from 'lockId' to 'lockID' in the calls to pinkLock.getLockById within both the activateListing and withdrawListing functions to ensure proper functionality.



No medium severity issues were found.



No low severity issues were found.



No Informational severity issues were found.

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.

Drops contracts tests:

Coverages:

```
contract: DropsLockMarketplace - 59.5%
  DropsLockMarketplace.changePriceInDrops - 100.0%
  DropsLockMarketplace.changePriceInETH - 100.0%
  Ownable._checkOwner - 100.0%
  DropsLockMarketplace.activateListing - 93.8%
  DropsLockMarketplace.redactListing - 87.5%
  DropsLockMarketplace.buyLockWithDrops - 81.2%
  DropsLockMarketplace.initiateListing - 81.2%
  DropsLockMarketplace.buyLockWithETH - 77.7%
  ReentrancyGuard._nonReentrantBefore - 75.0%
  DropsLockMarketplace.withdrawListing - 73.2%
  DropsLockMarketplace._getIndex - 53.6%
  DropsLockMarketplace._getIndexForUserLock - 33.9%
```

Testing:

```
Transaction sent: 0xfaSeed8be4a97ad5359be3faf02c33b7e9706c2a083be9a29c428348cb2a8158
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 11
Drops.transfer confirmed Block: 15 Gas used: 43252 (0.36%)

Transaction sent: 0xb7d39a5d54d5a7e10497da27c732ea03baea584eb20133f092227db34bd3dec5
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 3
Drops.setSellBuyTax confirmed (Ownable: caller is not the owner) Block: 16 Gas used: 22642 (0.19%)

Transaction sent: 0x1bb53a3bdb9e389d7380664e48d9ca01b338cd6b0f437f60e3993a9e9f60e509
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 12
Drops.setSellBuyTax confirmed (You can only lower fees) Block: 17 Gas used: 23631 (0.20%)

Transaction sent: 0xfb3e63c4ae40e0ea0a5491b26fcac9957535628c2f79d02158ee3b8256304ae1
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 13
Drops.setSellBuyTax confirmed Block: 18 Gas used: 27119 (0.23%)

tests/test_drops.py::test_transfer PASSED
tests/test_drops_marketplace.py::test_listing RUNNING
Transaction sent: 0x91e26fef8ec6fef4d405152ad461077313340749026cd2d5ef67d706a977b557
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 14
UniswapV2Factory.constructor confirmed Block: 19 Gas used: 2412730 (20.11%)
UniswapV2Factory deployed at: 0xe65A7a341978d59d40d30FC23F5014FACB4f575A

Transaction sent: 0x64f92e98cc38958bde48b00da891d4e84fc36b324fff92728f3af2f38c49557e
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 15
UniswapV2Locker.constructor confirmed Block: 20 Gas used: 2656179 (22.13%)
UniswapV2Locker deployed at: 0x303758532345801c88c2AD12541b09E9AA53A93d

Transaction sent: 0xf8b52f01f59bfe56acab0854acf7ff63b4ec61da711d71b8b08f9f26f73c64da8
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 16
WETH9.constructor confirmed Block: 21 Gas used: 476546 (3.97%)
WETH9 deployed at: 0x26f153358B1C6a4C08660eDd694a0555A9F1c3e3

Transaction sent: 0x4cd7c10e952908d347be0eff68d3d2bb04dce1fa648db5e8558aa6a0bc9f3e0
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 17
UniswapV2Router02.constructor confirmed Block: 22 Gas used: 3895430 (32.46%)
UniswapV2Router02 deployed at: 0xFb0588c72B438faD4Cf7cD879c8F730Faa2130a0

Transaction sent: 0x4c6ed9dfa466bb589c45f67eca5f53105357c0222a727f6ca874429c3a09b197
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 18
UniswapV2AddLiquidity.constructor confirmed Block: 23 Gas used: 449853 (3.75%)
UniswapV2AddLiquidity deployed at: 0xed00238F9A0F7b4d93842033cdF56cC832C781c2

Transaction sent: 0xc7f42017eda30812e5b76915eda2c72c3789535609e1c18a28dd788453ba997b
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 19
Drops.constructor confirmed Block: 24 Gas used: 4146942 (34.56%)
Drops deployed at: 0xDae02e4fE488952cF88c95177154D188647a0146

Transaction sent: 0x0d5230fdbb243a7e36650a14c55c5ef5ef901cf5338e01e59fee06f772ec4c70
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 20
DropsLockMarketplace.constructor confirmed Block: 25 Gas used: 2757893 (22.98%)
DropsLockMarketplace deployed at: 0xdCF93F11ef216cEC9C07fd31d0801c9b2b39Afb4

Transaction sent: 0x14960dc52b8c0c81ebfab9a36a48064681cbf25243b50dc575137aba3b941f4c
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 4
WETH9.deposit confirmed Block: 26 Gas used: 43706 (0.36%)

Transaction sent: 0xf3a5a9d484e4d41dc5315e92a01d345d25d39d4fe3a3182860242b767dc27865
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 21
Drops.setTradingOpen confirmed Block: 27 Gas used: 44880 (0.37%)

Transaction sent: 0xc9af84bd2e23c3541c145d20a15c2b53b773366c9dadd9e3af7eb18502f472d
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 22
Drops.transfer confirmed Block: 28 Gas used: 59942 (0.50%)
```

Transaction sent: 0x8c3e7f6dfa8c83a16601f3891ed9f6cc4ee07cbc83cddb6e8caadd38bbb4b698
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 5
WETH9.approve confirmed Block: 29 Gas used: 43986 (0.37%)

Transaction sent: 0xdd668fab873b15840ead9d705306d679455e2233db021698ea4dbade8ee97b79
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 6
Drops.approve confirmed Block: 30 Gas used: 44200 (0.37%)

Transaction sent: 0x05c9f25bd774e9ba4df5e04bc4b75383dd92fdf37b7d84271797c7468712aa2a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 7
UniswapV2AddLiquidity.addLiquidity confirmed Block: 31 Gas used: 314140 (2.62%)

Transaction sent: 0x99d64943edb04a10437429052665bfccdd6995be2adc881d4afaf87e8f73cbc41
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 8
Transaction confirmed Block: 32 Gas used: 44042 (0.37%)

Transaction sent: 0x576c190a680e351b8750fa0c8908701faf04045570a420524bbfa3ccb6dab901
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
Transaction confirmed Block: 33 Gas used: 36020 (0.30%)

Transaction sent: 0xf05fa19b2096da2a771dbc9c6387ff2549213faed1f101b88e8049f689da82e5
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 9
UniswapV2Locker.lockLPToken confirmed Block: 34 Gas used: 385420 (3.21%)

Transaction sent: 0x1045c9a546c046d80852bcb24427e297efe46184a06a830c919569e98fall7c9
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 0
DropsLockMarketplace.initiateListing confirmed (You dont own that lock.) Block: 35 Gas used: 31461 (0.26%)

Transaction sent: 0x8c191ad90ec8dbcb84494e79ea7eb73662dbe3ca8324e5ea36686d7436180c4a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 10
DropsLockMarketplace.initiateListing confirmed (You must set a price in Drops or ETH) Block: 36 Gas used: 31416 (0.26%)

Transaction sent: 0xde3967bb86d868043f27a6fa0689f79b8f22fbf93cc4a6666f9b5c57e0cbba31
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 11
DropsLockMarketplace.initiateListing confirmed Block: 37 Gas used: 281563 (2.35%)

Transaction sent: 0x235b1e20c8897f13045d8a788255469cb2ed0f9ac71e725f99ba8007746be2b1
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
DropsLockMarketplace.activateListing confirmed (Lock doesnt belong to you.) Block: 38 Gas used: 29894 (0.25%)

Transaction sent: 0x11bb58227e987c871e5881717f5d00e13beed705d11c124651690281c175d1cf
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 12
DropsLockMarketplace.activateListing confirmed (Lock ownership not yet transferred.) Block: 39 Gas used: 38492 (0.32%)

Transaction sent: 0x2dc3cd087ae348a4a429ced1d0a9a671633166e9be064d51b78d622285279c73
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 13
UniswapV2Locker.transferLockOwnership confirmed Block: 40 Gas used: 87746 (0.73%)

Transaction sent: 0xd634b987263bc6eebf49fb162662f2827b6de05755c664b4ce24f84a0edbc6d0
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 14
DropsLockMarketplace.activateListing confirmed Block: 41 Gas used: 67198 (0.56%)

Transaction sent: 0xdf075d1cde53bbbe2055e8e7b0b3906c2d6592d104b04cf6e2fe2d3036dd863a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 15
DropsLockMarketplace.activateListing confirmed (Listing already active.) Block: 42 Gas used: 29923 (0.25%)

Transaction sent: 0x07a8a20511f26ca914732138bc9c9c20c8de19ece6499acf187122d9bcac0c86
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 2
DropsLockMarketplace.buyLockWithETH confirmed (Listing must be active.) Block: 43 Gas used: 35431 (0.30%)

Transaction sent: 0xb2b4a9d01cb71889889d5a34692925eb8bfb2652cef6d3c50cbe039765aa3ab
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 3
DropsLockMarketplace.buyLockWithETH confirmed (Incorrect amount of ETH.) Block: 44 Gas used: 35728 (0.30%)


```
Transaction sent: 0x609b17761bf451c321871a956337e7235b52998e79be55b2b589258fe899b99a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 4
DropsLockMarketplace.buyLockWithETH confirmed Block: 45 Gas used: 129758 (1.08%)

Transaction sent: 0x421c303e9a023d81e85e6e91c8205cbd16b6d6903d730f4ecdb4119625b58c0b
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 16
DropsLockMarketplace.redactListing confirmed (Ownable: caller is not the owner) Block: 46 Gas used: 22887 (0.19%)

Transaction sent: 0xa18e2fbcd393b4bb84e1cf101ab9cbcd212ed5c06cfba7f4701dfe0930aefe5f
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 23
DropsLockMarketplace.redactListing confirmed (Listing does not exist.) Block: 47 Gas used: 23650 (0.20%)

Transaction sent: 0x410f1f098953f6d34628aa32091clab64alda5e38c46b629db3bd3e05fa71fba
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 24
DropsLockMarketplace.redactListing confirmed (Lock not found.) Block: 48 Gas used: 34990 (0.29%)

tests/test_drops_marketplace.py::test_listing PASSED
tests/test_drops_marketplace.py::test_listing_with_drops RUNNING
Transaction sent: 0x2777f20dd3bc257d5e86e6bf7258c21201682567d7c12cc830860243ced574bb
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 25
UniswapV2Factory.constructor confirmed Block: 49 Gas used: 2412730 (20.11%)
UniswapV2Factory deployed at: 0x654f70d8442EA18904FAD79114f7250F7E9336

Transaction sent: 0xfced2d85587471220fa005ba26b38de10d81c6db4d8e0aaeabaf5c92185e2d7
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 26
UniswapV2Locker.constructor confirmed Block: 50 Gas used: 2656179 (22.13%)
UniswapV2Locker deployed at: 0xADE061D42dE86f9058386D1D0d739d20C7eAfC43

Transaction sent: 0x8648c20cc1678a3f5bc688f117bb2f629c8bc2dd99bb8bed7570350e93d75048
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 27
WETH9.constructor confirmed Block: 51 Gas used: 476546 (3.97%)
WETH9 deployed at: 0x8326980aec363C9A7aB036C224Af5821280b3AC6

Transaction sent: 0x98e7b4fab595c2f846194c48332775093316ad7c01c5e705c4d6db6e93506dd7
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 28
UniswapV2Router02.constructor confirmed Block: 52 Gas used: 3895430 (32.46%)
UniswapV2Router02 deployed at: 0xA95916C3D979400C7443961330b3092510a2298a

Transaction sent: 0x011dfce2250581fc1c17c4128174077550c3a621379407ea409921e07e5f551f
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 29
UniswapV2AddLiquidity.constructor confirmed Block: 53 Gas used: 449853 (3.75%)
UniswapV2AddLiquidity deployed at: 0x42E8D004c84E6B58ad559D3b5CE7947AADb9E0bc

Transaction sent: 0xf619bd146b6a772cc0a89fda485282e02b087ea45ab84ad656564e1c625f94f9
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 30
Drops.constructor confirmed Block: 54 Gas used: 4146942 (34.56%)
Drops deployed at: 0xF06D5f58fFFC86a52c84cfec03AD35637728E73

Transaction sent: 0xde1f6f1678283b962dc0afcfaf730100d3687c619e4b34abf622e5f6df175ca7
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 31
DropsLockMarketplace.constructor confirmed Block: 55 Gas used: 2757893 (22.98%)
DropsLockMarketplace deployed at: 0x82c83b7f88aef2eD99d4869D547b6ED28e69C8df

Transaction sent: 0xe39976adaf27834466065398d65d304a102bf8574ee07d9f7b082e2c2fab4ab1
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 17
WETH9.deposit confirmed Block: 56 Gas used: 43706 (0.36%)

Transaction sent: 0x910a11e3945555a7d29e7f6c26e49ebf38a514c9c961b6091c3d36be0edce72f
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 32
Drops.setTradingOpen confirmed Block: 57 Gas used: 44880 (0.37%)

Transaction sent: 0x859740d01ccb9239fb443ffae3b16e6e40df06c18192a4862992f0dea5b9dc94
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 33
Drops.transfer confirmed Block: 58 Gas used: 59942 (0.50%)
```

```
Transaction sent: 0xce3ddb5867ac7d99d511ca199d1e6ab0c9034c250304d3d55c22e54cba927610
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 18
WETH9.approve confirmed Block: 59 Gas used: 43998 (0.37%)

Transaction sent: 0x109855c2b710c57ee3956b651f37fa526408b8e1088d2f027e1d2809390c08fd
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 19
Drops.approve confirmed Block: 60 Gas used: 44212 (0.37%)

Transaction sent: 0x89412b916deecae745e47323263f614f6555352cb688aae79add84595b27803
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 20
UniswapV2AddLiquidity.addLiquidity confirmed Block: 61 Gas used: 314140 (2.62%)

Transaction sent: 0x92623961ada4670e23040dbfd9f24f034f4c0ce4b6821998f52e0f34d6b275bf
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 21
Transaction confirmed Block: 62 Gas used: 44042 (0.37%)

Transaction sent: 0x4ac57e1ce17b4ea804c1acbea83cb583a1f3a5d5283c33fcae685973d7f269b
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
Transaction confirmed Block: 63 Gas used: 36020 (0.30%)

Transaction sent: 0xecf73a055a0250ace3387e58c60d86b62a83af918b446f4413eebf78f2bb665d
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 22
UniswapV2Locker.lockLPToken confirmed Block: 64 Gas used: 385420 (3.21%)

Transaction sent: 0x8d6f0ad29de6629eb62c85e4c66e2cd31296bf746fbf2f52db090784eb12f9fa
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 23
DropsLockMarketplace.initiateListing confirmed Block: 65 Gas used: 281563 (2.35%)

Transaction sent: 0xb2746187df73c6b67711ca2764a46d911e90cf36288f50c601b1bdf71b9d3d88
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 24
UniswapV2Locker.transferLockOwnership confirmed Block: 66 Gas used: 87746 (0.73%)

Transaction sent: 0xdeafad30ae8babe31cde4e321a9ca91119b0b40b2f57d1ae0fcdcf8181ca8cde
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 25
DropsLockMarketplace.activateListing confirmed Block: 67 Gas used: 67198 (0.56%)

Transaction sent: 0xa713fe49a0319afe2e10171b2b824dcad16abb288cbdef6bc965449c474d3b59
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 5
DropsLockMarketplace.buyLockWithDrops confirmed (Listing must be active.) Block: 68 Gas used: 35431 (0.30%)

Transaction sent: 0x24b952b96aa6bab9af77a9fb5f5f36711720376296562bd77485cca5ff26belc
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 6
DropsLockMarketplace.buyLockWithDrops confirmed (Insufficient drops.) Block: 69 Gas used: 38801 (0.32%)

Transaction sent: 0xe9dd66c7b78057bb1f6cd517043c4e96f53769ee2ba336b8b5bfcd8fc631f255
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 34
Drops.transfer confirmed Block: 70 Gas used: 59930 (0.50%)

Transaction sent: 0xdd09973aec4c97aea10c7cc39ccc21b8591f7127af752106d2db058df1327404
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 7
Drops.approve confirmed Block: 71 Gas used: 44200 (0.37%)

Transaction sent: 0xe02d5e4a301a49bfbec3bc6b214b5466164fa46cc551ee9f828adfce7677ac41
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 8
DropsLockMarketplace.buyLockWithDrops confirmed Block: 72 Gas used: 122232 (1.02%)

Transaction sent: 0x21f4113a5fc6041ac20e1eedb6b35165de4a52480b5ad61bb6be22e6434c5756
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 9
DropsLockMarketplace.withdrawListing confirmed (This listing does not belong to you.) Block: 73 Gas used: 35715 (0.30%)

Transaction sent: 0xbd0f30c5c8a9a44c32261fd95deba9e2665dc91f9e4582e380de16262de59991
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 26
DropsLockMarketplace.withdrawListing confirmed (Marketplace does not own your lock) Block: 74 Gas used: 44210 (0.37%)
```

```
Transaction sent: 0x98f73ef2f76694df5465660822a80d79547e756f4b23bfae6aa801646633d496
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 10
UniswapV2Locker.transferLockOwnership confirmed Block: 75 Gas used: 87746 (0.73%)

Transaction sent: 0x71f98789549b9a3e85312bc2c685500f6b1b4e6b1fde5cc4d22cc1dall11839a6
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 27
DropsLockMarketplace.withdrawListing confirmed Block: 76 Gas used: 113761 (0.95%)

tests/test_drops_marketplace.py::test_listing_with_drops PASSED
tests/test_drops_marketplace.py::test_redact_listing RUNNING
Transaction sent: 0x88bcc5ec517fc6d00a705470aa0a513ea0540221260b2fb0052b506871d46dbc
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 35
UniswapV2Factory.constructor confirmed Block: 77 Gas used: 2412730 (20.11%)
UniswapV2Factory deployed at: 0xbc8eCcb89650c3E796e803C80098F9b898C8359

Transaction sent: 0x988263f94cc162ccd9eb557ff3f946ee364bbf5132327bec921b76dd202a2ed4
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 36
UniswapV2Locker.constructor confirmed Block: 78 Gas used: 2656179 (22.13%)
UniswapV2Locker deployed at: 0x741e3E1f81041c62C2A97d0b6E567Aca809A6232

Transaction sent: 0xecbabe8a7b5d619b2aa2d435d8468e8e780abdb5389498a0427789a42012e6fc
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 37
WETH9.constructor confirmed Block: 79 Gas used: 476546 (3.97%)
WETH9 deployed at: 0x480FccF53589c1F185835db88b8315a0b8F9a3e0

Transaction sent: 0xfcc3c971eld16d34a6674bc19b76e887043194f02b0eafe45766c9c26741b93e
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 38
UniswapV2Router02.constructor confirmed Block: 80 Gas used: 3895430 (32.46%)
UniswapV2Router02 deployed at: 0xFE0F4Cf81B5c0a6Fd65a610FD0488F33aE9095cB

Transaction sent: 0x260136ddd240dd545086167507749078c196cca12be3b56c48d9c21a0566a33b
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 39
UniswapV2AddLiquidity.constructor confirmed Block: 81 Gas used: 449853 (3.75%)
UniswapV2AddLiquidity deployed at: 0x4e3E7dC9D84dA78E8f017f4C36153A61341736d4

Transaction sent: 0x851d73a6124bd05ba4f4755e13e0cc331298de54056b68f616cae851b4dfa2f7
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 40
Drops.constructor confirmed Block: 82 Gas used: 4146932 (34.56%)
Drops deployed at: 0x0AC45e945A008D3fc19da8f591be8601C1F63130

Transaction sent: 0xa8503b2e9bc3e40e0fb48e74859ac1972090b9f1fb02f6df4792227d34c2ce75
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 41
DropsLockMarketplace.constructor confirmed Block: 83 Gas used: 2757881 (22.98%)
DropsLockMarketplace deployed at: 0x5847798CE8c89e3Fff59AE5fA308EC0d406b5687

Transaction sent: 0xc08738643dc0ea6ced833f7fe1c421e7589f3d27485cf8b8b06bf66a4f59e311
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 28
WETH9.deposit confirmed Block: 84 Gas used: 43706 (0.36%)

Transaction sent: 0xcf527478f29c0aecf8b4624dd77f591d0f4d6743aeb98dc114f94194ae918b6a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 42
Drops.setTradingOpen confirmed Block: 85 Gas used: 44880 (0.37%)

Transaction sent: 0x87286510491e5c7b72b790480332663b80a52688bc4ccf7a62e429e4545f0b17
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 43
Drops.transfer confirmed Block: 86 Gas used: 59942 (0.50%)

Transaction sent: 0x2dff8ac5174477afb57bb5d89cc7d374f3c8a5cfa96e980a9d281c1bedfd6eb
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 29
WETH9.approve confirmed Block: 87 Gas used: 43998 (0.37%)

Transaction sent: 0xb5d30a35b84879e6d251a05c0f3e2974df6ee9d37f45fe5249102a509693bc87
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 30
Drops.approve confirmed Block: 88 Gas used: 44212 (0.37%)
```

```
Transaction sent: 0x98c955684bab581faee4dc21bb7a2897a5db6d0052d7fc02aa760fc8b9522b6
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 31
UniswapV2AddLiquidity.addLiquidity confirmed Block: 89 Gas used: 314088 (2.62%)

Transaction sent: 0x2652c309f35bdbfbbea71c0fcb058e03b8271aa78d8506f5fa29236a5d514a926
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 32
Transaction confirmed Block: 90 Gas used: 44042 (0.37%)

Transaction sent: 0xaa7e4d063fdec1e46d41e016fd09b6db508cf8c95dd8e51845dcf2d485ebdda4
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
Transaction confirmed Block: 91 Gas used: 36020 (0.30%)

Transaction sent: 0x3297721e9306a272a29cccefe8085b4f5e8900c9fc62f52edd321c32d7b752a
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 33
UniswapV2Locker.lockLPToken confirmed Block: 92 Gas used: 385420 (3.21%)

Transaction sent: 0xf15a97eeb06267fcd10ebdb3b96c240777cd533b35880e162d87e9708c11e6a3
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 34
DropsLockMarketplace.initiateListing confirmed Block: 93 Gas used: 281563 (2.35%)

Transaction sent: 0x560b024db91eab8f1614238bled8e777d74a3b10802b43fd26f027990815b1eb
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 35
UniswapV2Locker.transferLockOwnership confirmed Block: 94 Gas used: 87746 (0.73%)

Transaction sent: 0xf5b8596236fc23ea0a84927e21ca19cab3af930353982fc9122c848a41358b7
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 36
DropsLockMarketplace.activateListing confirmed Block: 95 Gas used: 67198 (0.56%)

Transaction sent: 0x098aa62bef9a5c15c23e99f783330c0b62d531244d87bdccc213929bb894987
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 44
DropsLockMarketplace.redactListing confirmed Block: 96 Gas used: 107002 (0.89%)

tests/test_drops_marketplace.py::test_redact_listing PASSED
tests/test_drops_marketplace.py::test_change_price RUNNING
Transaction sent: 0xdc16e00c878bf9cb9a8e2eca896ed061247282ea7162de8d440b88fcc1cf1617
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 45
UniswapV2Factory.constructor confirmed Block: 97 Gas used: 2412730 (20.11%)
UniswapV2Factory deployed at: 0xa1910C6e0Fbd0E38cfBabAeC5D1C1D539F81CC63

Transaction sent: 0xbccab8bad0e5b088347a89b50c94a6230c58ca74e48cbc2f6d0e4b12dfae800
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 46
UniswapV2Locker.constructor confirmed Block: 98 Gas used: 2656179 (22.13%)
UniswapV2Locker deployed at: 0x88E16B874F47371C42498b499837Ed87D7661E86

Transaction sent: 0x3d411f9e6e8831730200b2a8607a3fa7a78a4bb8223e0d0943189380e40b8eeb
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 47
WETH9.constructor confirmed Block: 99 Gas used: 476546 (3.97%)
WETH9 deployed at: 0x1aa96efd28002541E830CB7f60e473AA24e31F9A

Transaction sent: 0x73dabd716c258797feaf97893931cd531080c3d25f5ed06b064e169df627df3f
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 48
UniswapV2Router02.constructor confirmed Block: 100 Gas used: 3895418 (32.46%)
UniswapV2Router02 deployed at: 0x8b184407240Ce2EE9779B58af841Ec59F545838B

Transaction sent: 0x7594b8ac35a0b1c1debc4a6857be3f00628b7a33df335c4ae0455adcaacb17d
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 49
UniswapV2AddLiquidity.constructor confirmed Block: 101 Gas used: 449853 (3.75%)
UniswapV2AddLiquidity deployed at: 0xc6D563d5c2243b27e7294511063f563ED701EA2C

Transaction sent: 0x3ac36f78c19f7a9772dc5181f8b6a6cb87056d7814da126844fb3b32a59e4008
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 50
Drops.constructor confirmed Block: 102 Gas used: 4146942 (34.56%)
Drops deployed at: 0x0537bf4b795b7D078d5F4bAf7017e3ce836081DE
```



```
Transaction sent: 0x60be063e5cf63a4377a4e5ffa6744d4a891739303cd415db3fc91017fd2e4fe
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 51
DropsLockMarketplace.constructor confirmed Block: 103 Gas used: 2757893 (22.98%)
DropsLockMarketplace deployed at: 0x70bC60873D1100a59a9c49E7485a27B0F605E5db

Transaction sent: 0xa3dfd38c211fa9288cfc0619bebb1f8b25f0e48b12540399418f4d7a49128b1
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 37
WETH9.deposit confirmed Block: 104 Gas used: 43706 (0.36%)

Transaction sent: 0x1a350fa8093b8ce4cdf01961e492f22a45bad14cde357a0cd46df1d1d1c0cc9e
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 52
Drops.setTradingOpen confirmed Block: 105 Gas used: 44880 (0.37%)

Transaction sent: 0x735f105f8fd1855c2f0c8a24c39767f61f1cfa63baedddc544785ba1f71984c
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 53
Drops.transfer confirmed Block: 106 Gas used: 59942 (0.50%)

Transaction sent: 0x3361b69a816a3ca2d10dd55f754143919865ad1a4a0d6a2cf18173476ad7d193
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 38
WETH9.approve confirmed Block: 107 Gas used: 43998 (0.37%)

Transaction sent: 0xea63a567e2a6a74ee0085becb21579d1a5fda8f0ae3b653420248ffe7ad173e
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 39
Drops.approve confirmed Block: 108 Gas used: 44212 (0.37%)

Transaction sent: 0x4c5e9915efcd0358c82e70d31d77cc2f2a915959b8a6ce24f9ff0a57aada0552
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 40
UniswapV2AddLiquidity.addLiquidity confirmed Block: 109 Gas used: 314128 (2.62%)

Transaction sent: 0xce4e6d4b762d30bc19eb1380c222d2ba1387a6a59588950b3700d3849dc31817
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 41
Transaction confirmed Block: 110 Gas used: 44042 (0.37%)

Transaction sent: 0x01e7fe704cbd72c2b6cdef51e34028b47897a0048cb6637374651f0dda1fe31c
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 1
Transaction confirmed Block: 111 Gas used: 36020 (0.30%)

Transaction sent: 0x52d1038a66167eab8eeae1f0c0e8502ed10c7e1f76f7b5d7fdd4153fdac19992
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 42
UniswapV2Locker.lockLPToken confirmed Block: 112 Gas used: 385420 (3.21%)

Transaction sent: 0x10f6ca212bbb33046c9dd0d9e6b95264027b0102d05f285f62f910442527bb26
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 43
DropsLockMarketplace.initiateListing confirmed Block: 113 Gas used: 281563 (2.35%)

Transaction sent: 0x0ab87532126e3d0f0f99361819f3f844d022a30a6cdce42eb160c7875fc1f6a5
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 11
DropsLockMarketplace.changePriceInETH confirmed (This listing does not belong to you.) Block: 114 Gas used: 29148 (0.24%)

Transaction sent: 0xd47bfa999bea2594c7ff07006838a6f36d3970ac9f7add21d9797325d1e024bb0
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 44
DropsLockMarketplace.changePriceInETH confirmed Block: 115 Gas used: 30632 (0.26%)

Transaction sent: 0xba77903bdc76a8842af0ca31e3e01d857f45e314aa603c7cc123e642c7e38d71
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 12
DropsLockMarketplace.changePriceInDrops confirmed (This listing does not belong to you.) Block: 116 Gas used: 29217 (0.24%)

Transaction sent: 0x42cc04871ff8a90bf3b64169f0b1ee864dfac1069086fe30664f42f037061ee7
Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 45
DropsLockMarketplace.changePriceInDrops confirmed Block: 117 Gas used: 30701 (0.26%)

tests/test_drops_marketplace.py::test_change_price PASSED
```

Annexes

Testing code:

UnicryptMarketplace.sol:

```
from brownie import (
    reverts,
    UniswapV2Pair
)

from brownie.network.contract import Contract

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

from scripts.deploy import (
    deploy_weth,
    deploy_router,
    deploy_drops,
    deploy_lock_marketplace,
    deploy_locker,
    deploy_factory,
    deploy_liquidity
)

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

    # Deploy contracts
    factory = deploy_factory(owner, owner)
    locker = deploy_locker(owner, factory.address)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
    liquidity = deploy_liquidity(owner, factory.address, router.address)
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
    market = deploy_lock_marketplace(owner, locker.address, owner, drops.address)

    # Add liquidity
    weth.deposit({"from": other, "value": 10e18})
    drops.setTradingOpen({"from": owner})
    drops.transfer(other, 5e18, {"from": owner})
```

```

weth.approve(liquidity.address, 100e18, {"from": other})
drops.approve(liquidity.address, 100e18, {"from": other})
liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

pair_addr = drops.uniswapV2Pair()
pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
pair.approve(locker.address, 10e18, {"from": other})
pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

# Lock tokens
locker.lockLPToken(pair_addr, 3e10, get_timestamp(), ZERO_ADDRESS, True, other,
{"from": other, "value": 1e18})

with reverts("You dont own that lock."):
    market.initiateListing(pair_addr, 0, 1e16, 1e16, other, {"from": extra})
with reverts("You must set a price in Drops or ETH"):
    market.initiateListing(pair_addr, 0, 0, 0, other, {"from": other})

tx = market.initiateListing(pair_addr, 0, 1e18, 1e18, other, {"from": other})
assert tx.events['ListingInitiated'][0]['lpToken'] == pair_addr
assert tx.events['ListingInitiated'][0]['lockID'] == 0
assert tx.events['ListingInitiated'][0]['seller'] == other

with reverts("Lock doesnt belong to you."):
    market.activateListing(pair_addr, 0, {"from": extra})
with reverts("Lock ownership not yet transferred."):
    market.activateListing(pair_addr, 0, {"from": other})
locker.transferLockOwnership(pair_addr, 0, 0, market.address, {"from": other})
market.activateListing(pair_addr, 0, {"from": other})
with reverts("Listing already active."):
    market.activateListing(pair_addr, 0, {"from": other})

with reverts("Listing must be active."):
    market.buyLockWithETH(ZERO_ADDRESS, 0, {"from": extra})
with reverts("Incorrect amount of ETH."):
    market.buyLockWithETH(pair_addr, 0, {"from": extra, "value": 1e17})
market.buyLockWithETH(pair_addr, 0, {"from": extra, "value": 1e18})

with reverts("Ownable: caller is not the owner"):
    market.redactListing(pair_addr, 0, {"from": other})
with reverts("Listing does not exist."):
    market.redactListing(ZERO_ADDRESS, 0, {"from": owner})
with reverts("Lock not found."):
    market.redactListing(pair_addr, 0, {"from": owner})

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

```

```

extra = get_account(2)

# Deploy contracts
factory = deploy_factory(owner, owner)
locker = deploy_locker(owner, factory.address)
weth = deploy_weth(owner)
router = deploy_router(owner, factory.address, weth.address)
liquidity = deploy_liquidity(owner, factory.address, router.address)
drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
market = deploy_lock_marketplace(owner, locker.address, owner, drops.address)

# Add liquidity
weth.deposit({"from": other, "value": 10e18})
drops.setTradingOpen({"from": owner})
drops.transfer(other, 5e18, {"from": owner})
weth.approve(liquidity.address, 100e18, {"from": other})
drops.approve(liquidity.address, 100e18, {"from": other})
liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

pair_addr = drops.uniswapV2Pair()
pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
pair.approve(locker.address, 10e18, {"from": other})
pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

# Lock tokens
locker.lockLPToken(pair_addr, 3e10, get_timestamp(), ZERO_ADDRESS, True, other, {"from": other, "value": 1e18})
tx = market.initiateListing(pair_addr, 0, 1e18, 1e18, other, {"from": other})
assert tx.events['ListingInitiated'][0]['lpToken'] == pair_addr
assert tx.events['ListingInitiated'][0]['lockID'] == 0
assert tx.events['ListingInitiated'][0]['seller'] == other

locker.transferLockOwnership(pair_addr, 0, 0, market.address, {"from": other})
market.activateListing(pair_addr, 0, {"from": other})

with reverts("Listing must be active."):
    market.buyLockWithDrops(ZERO_ADDRESS, 0, {"from": extra})
with reverts("Insufficient drops."):
    market.buyLockWithDrops(pair_addr, 0, {"from": extra})
drops.transfer(extra, 5e18, {"from": owner})
drops.approve(market.address, 1e18, {"from": extra})
tx = market.buyLockWithDrops(pair_addr, 0, {"from": extra})
assert tx.events['LockPurchasedWithDrops'][0]['lpToken'] == pair_addr
assert tx.events['LockPurchasedWithDrops'][0]['lockID'] == 0
assert tx.events['LockPurchasedWithDrops'][0]['profitInDrops'] == 1e18

with reverts("This listing does not belong to you."):
    market.withdrawListing(pair_addr, 0, {"from": extra})
with reverts("Marketplace does not own your lock"):

```



```

        market.withdrawListing(pair_addr, 0, {"from": other})
        locker.transferLockOwnership(pair_addr, 0, 0, market.address, {"from": extra})
        tx = market.withdrawListing(pair_addr, 0, {"from": other})
        assert tx.events['ListingWithdrawn'][0]['lpToken'] == pair_addr
        assert tx.events['ListingWithdrawn'][0]['lockID'] == 0

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

    # Deploy contracts
    factory = deploy_factory(owner, owner)
    locker = deploy_locker(owner, factory.address)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
    liquidity = deploy_liquidity(owner, factory.address, router.address)
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
    market = deploy_lock_marketplace(owner, locker.address, owner, drops.address)

    # Add liquidity
    weth.deposit({"from": other, "value": 10e18})
    drops.setTradingOpen({"from": owner})
    drops.transfer(other, 5e18, {"from": owner})
    weth.approve(liquidity.address, 100e18, {"from": other})
    drops.approve(liquidity.address, 100e18, {"from": other})
    liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

    pair_addr = drops.uniswapV2Pair()
    pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
    pair.approve(locker.address, 10e18, {"from": other})
    pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

    # Lock tokens
    locker.lockLPToken(pair_addr, 3e10, get_timestamp(), ZERO_ADDRESS, True, other,
{"from": other, "value": 1e18})
    market.initiateListing(pair_addr, 0, 1e18, 1e18, other, {"from": other})
    locker.transferLockOwnership(pair_addr, 0, 0, market.address, {"from": other})
    market.activateListing(pair_addr, 0, {"from": other})
    tx = market.redactListing(pair_addr, 0, {"from": owner})
    assert tx.events['ListingRedacted'][0]['lpToken'] == pair_addr
    assert tx.events['ListingRedacted'][0]['lockID'] == 0

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

```

```

# Deploy contracts
factory = deploy_factory(owner, owner)
locker = deploy_locker(owner, factory.address)
weth = deploy_weth(owner)
router = deploy_router(owner, factory.address, weth.address)
liquidity = deploy_liquidity(owner, factory.address, router.address)
drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
market = deploy_lock_marketplace(owner, locker.address, owner, drops.address)

# Add liquidity
weth.deposit({"from": other, "value": 10e18})
drops.setTradingOpen({"from": owner})
drops.transfer(other, 5e18, {"from": owner})
weth.approve(liquidity.address, 100e18, {"from": other})
drops.approve(liquidity.address, 100e18, {"from": other})
liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

pair_addr = drops.uniswapV2Pair()
pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
pair.approve(locker.address, 10e18, {"from": other})
pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

# Lock tokens
locker.lockLPToken(pair_addr, 3e10, get_timestamp(), ZERO_ADDRESS, True, other, {"from": other, "value": 1e18})
market.initiateListing(pair_addr, 0, 1e18, 1e18, other, {"from": other})

with reverts("This listing does not belong to you."):
    market.changePriceInETH(pair_addr, 0, 2e18, {"from": extra})
market.changePriceInETH(pair_addr, 0, 2e18, {"from": other})

with reverts("This listing does not belong to you."):
    market.changePriceInDrops(pair_addr, 0, 2e18, {"from": extra})
market.changePriceInDrops(pair_addr, 0, 2e18, {"from": other})

```

DropsPinksaleBSCMarketplace.sol:

```

from brownie import (
    reverts,
    UniswapV2Pair
)

from brownie.network.contract import Contract

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

from scripts.deploy import (
    deploy_weth,
    deploy_router,
    deploy_drops,
    deploy_pinksale_marketplace,
    deploy_pink_lock,
    deploy_factory,
    deploy_liquidity
)

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

    # Deploy contracts
    factory = deploy_factory(owner, owner)
    locker = deploy_pink_lock(owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
    liquidity = deploy_liquidity(owner, factory.address, router.address)
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
    market = deploy_pinksale_marketplace(owner, locker.address, owner)

    # Add liquidity
    weth.deposit({"from": other, "value": 10e18})
    drops.setTradingOpen({"from": owner})
    drops.transfer(other, 5e18, {"from": owner})
    weth.approve(liquidity.address, 100e18, {"from": other})
    drops.approve(liquidity.address, 100e18, {"from": other})
    liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

    pair_addr = drops.uniswapV2Pair()
    pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
    pair.approve(locker.address, 10e18, {"from": other})

```

```

pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

# Lock tokens
tx = locker.lock(other, pair_addr, True, pair.balanceOf(other), get_timestamp(),
"some_description", {"from": other})
lock_id = tx.events['LockAdded'][0]['id']

with reverts("Invalid lock id"):
    market.initiateListing(pair_addr, 0, 1e16, other, {"from": extra})
with reverts("Invalid LP address"):
    market.initiateListing(factory.address, lock_id, 1e16, other, {"from": extra})
with reverts("You do not own this lock"):
    market.initiateListing(pair_addr, lock_id, 1e16, other, {"from": extra})
with reverts("Must set a lock price in ETH"):
    market.initiateListing(pair_addr, lock_id, 0, other, {"from": other})

tx = market.initiateListing(pair_addr, lock_id, 1e18, other, {"from": other})
assert tx.events['ListingInitiated'][0]['lpToken'] == pair_addr
assert tx.events['ListingInitiated'][0]['lockID'] == lock_id
assert tx.events['ListingInitiated'][0]['seller'] == other

with reverts("Lock doesnt belong to you."):
    market.activateListing(pair_addr, 0, {"from": extra})
with reverts("Lock ownership not yet transferred."):
    market.activateListing(pair_addr, lock_id, {"from": other})
locker.transferLockOwnership(lock_id, market.address, {"from": other})
market.activateListing(pair_addr, lock_id, {"from": other})
with reverts("Listing already active."):
    market.activateListing(pair_addr, lock_id, {"from": other})

with reverts("Listing must be active."):
    market.buyLockWithETH(ZERO_ADDRESS, lock_id, {"from": extra})
with reverts("Incorrect amount of ETH."):
    market.buyLockWithETH(pair_addr, lock_id, {"from": extra, "value": 1e17})
tx = market.buyLockWithETH(pair_addr, lock_id, {"from": extra, "value": 1e18})

with reverts("Ownable: caller is not the owner"):
    market.redactListing(pair_addr, lock_id, {"from": other})
with reverts("Listing does not exist."):
    market.redactListing(ZERO_ADDRESS, lock_id, {"from": owner})
with reverts("You are not the owner of this lock"):
    market.redactListing(pair_addr, lock_id, {"from": owner})

with reverts("This listing does not belong to you."):
    market.withdrawListing(pair_addr, 0, {"from": extra})
with reverts("Marketplace does not own your lock"):
    market.withdrawListing(pair_addr, lock_id, {"from": other})
locker.transferLockOwnership(lock_id, market.address, {"from": extra})

```



```

tx = market.withdrawListing(pair_addr, lock_id, {"from": other})
assert tx.events['ListingWithdrawn'][0]['lpToken'] == pair_addr
assert tx.events['ListingWithdrawn'][0]['lockID'] == lock_id

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

    # Deploy contracts
    factory = deploy_factory(owner, owner)
    locker = deploy_pink_lock(owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
    liquidity = deploy_liquidity(owner, factory.address, router.address)
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
    market = deploy_pinksale_marketplace(owner, locker.address, owner)

    # Add liquidity
    weth.deposit({"from": other, "value": 10e18})
    drops.setTradingOpen({"from": owner})
    drops.transfer(other, 5e18, {"from": owner})
    weth.approve(liquidity.address, 100e18, {"from": other})
    drops.approve(liquidity.address, 100e18, {"from": other})
    liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

    pair_addr = drops.uniswapV2Pair()
    pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
    pair.approve(locker.address, 10e18, {"from": other})
    pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

    # Lock tokens
    tx = locker.lock(other, pair_addr, True, pair.balanceOf(other), get_timestamp(),
"some_description", {"from": other})
    lock_id = tx.events['LockAdded'][0]['id']

    market.initiateListing(pair_addr, lock_id, 1e18, other, {"from": other})
    locker.transferLockOwnership(lock_id, market.address, {"from": other})
    market.activateListing(pair_addr, lock_id, {"from": other})
    tx = market.redactListing(pair_addr, lock_id, {"from": owner})
    assert tx.events['ListingRedacted'][0]['lpToken'] == pair_addr
    assert tx.events['ListingRedacted'][0]['lockID'] == lock_id

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

```

```

# Deploy contracts
factory = deploy_factory(owner, owner)
locker = deploy_pink_lock(owner)
weth = deploy_weth(owner)
router = deploy_router(owner, factory.address, weth.address)
liquidity = deploy_liquidity(owner, factory.address, router.address)
drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
market = deploy_pinksale_marketplace(owner, locker.address, owner)

# Add liquidity
weth.deposit({"from": other, "value": 10e18})
drops.setTradingOpen({"from": owner})
drops.transfer(other, 5e18, {"from": owner})
weth.approve(liquidity.address, 100e18, {"from": other})
drops.approve(liquidity.address, 100e18, {"from": other})
liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

pair_addr = drops.uniswapV2Pair()
pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
pair.approve(locker.address, 10e18, {"from": other})
pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

# Lock tokens
tx = locker.lock(other, pair_addr, True, pair.balanceOf(other), get_timestamp(),
"some_description", {"from": other})
lock_id = tx.events['LockAdded'][0]['id']
market.initiateListing(pair_addr, lock_id, 1e18, other, {"from": other})

with reverts("This listing does not belong to you."):
    market.changePriceInETH(pair_addr, lock_id, 2e18, {"from": extra})
market.changePriceInETH(pair_addr, lock_id, 2e18, {"from": other})

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

    # Deploy contracts
    locker = deploy_pink_lock(owner)
    market = deploy_pinksale_marketplace(owner, locker.address, owner)

    with reverts("Ownable: caller is not the owner"):
        market.setFeeWallet(extra, {"from": other})
    with reverts("Same wallet"):
        market.setFeeWallet(owner, {"from": owner})
    with reverts("Cant set fee wallet as zero address"):
        market.setFeeWallet(ZERO_ADDRESS, {"from": owner})

```

```
assert market.feeWallet() == owner
market.setFeeWallet(extra, {"from": owner})
assert market.feeWallet() == extra
```

PinksaleBSCMarketplace.sol:

```
from brownie import (
    reverts,
    UniswapV2Pair
)

from brownie.network.contract import Contract

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

from scripts.deploy import (
    deploy_weth,
    deploy_router,
    deploy_drops,
    deploy_pinksale_marketplace,
    deploy_pink_lock,
    deploy_factory,
    deploy_liquidity
)

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

    # Deploy contracts
    factory = deploy_factory(owner, owner)
    locker = deploy_pink_lock(owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
    liquidity = deploy_liquidity(owner, factory.address, router.address)
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
```

```

market = deploy_pinksale_marketplace(owner, locker.address, owner, drops.address)

# Add liquidity
weth.deposit({"from": other, "value": 10e18})
drops.setTradingOpen({"from": owner})
drops.transfer(other, 5e18, {"from": owner})
weth.approve(liquidity.address, 100e18, {"from": other})
drops.approve(liquidity.address, 100e18, {"from": other})
liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

pair_addr = drops.uniswapV2Pair()
pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
pair.approve(locker.address, 10e18, {"from": other})
pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

# Lock tokens
tx = locker.lock(other, pair_addr, True, pair.balanceOf(other), get_timestamp(),
"some_description", {"from": other})
lock_id = tx.events['LockAdded'][0]['id']

with reverts("Invalid lock id"):
    market.initiateListing(pair_addr, 0, 1e16, 1e16, other, {"from": extra})
with reverts("Invalid LP address"):
    market.initiateListing(factory.address, lock_id, 1e16, 1e16, other, {"from":
extra})
with reverts("You do not own this lock"):
    market.initiateListing(pair_addr, lock_id, 1e16, 1e16, other, {"from": extra})
with reverts("Must set a lock price in Drops or ETH"):
    market.initiateListing(pair_addr, lock_id, 0, 0, other, {"from": other})

tx = market.initiateListing(pair_addr, lock_id, 1e18, 1e18, other, {"from":
other})
assert tx.events['ListingInitiated'][0]['lpToken'] == pair_addr
assert tx.events['ListingInitiated'][0]['lockID'] == lock_id
assert tx.events['ListingInitiated'][0]['seller'] == other

with reverts("Lock doesnt belong to you."):
    market.activateListing(pair_addr, 0, {"from": extra})
with reverts("Lock ownership not yet transferred."):
    market.activateListing(pair_addr, lock_id, {"from": other})
locker.transferLockOwnership(lock_id, market.address, {"from": other})
market.activateListing(pair_addr, lock_id, {"from": other})
with reverts("Listing already active."):
    market.activateListing(pair_addr, lock_id, {"from": other})

with reverts("Listing must be active."):
    market.buyLockWithETH(ZERO_ADDRESS, lock_id, {"from": extra})
with reverts("Incorrect amount of ETH."):
    market.buyLockWithETH(pair_addr, lock_id, {"from": extra, "value": 1e17})

```



```

market.buyLockWithETH(pair_addr, lock_id, {"from": extra, "value": 1e18})

with reverts("Ownable: caller is not the owner"):
    market.redactListing(pair_addr, lock_id, {"from": other})
with reverts("Listing does not exist."):
    market.redactListing(ZERO_ADDRESS, lock_id, {"from": owner})
with reverts("You are not the owner of this lock"):
    market.redactListing(pair_addr, lock_id, {"from": owner})

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

    # Deploy contracts
    factory = deploy_factory(owner, owner)
    locker = deploy_pink_lock(owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
    liquidity = deploy_liquidity(owner, factory.address, router.address)
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
    market = deploy_pinksale_marketplace(owner, locker.address, owner, drops.address)

    # Add liquidity
    weth.deposit({"from": other, "value": 10e18})
    drops.setTradingOpen({"from": owner})
    drops.transfer(other, 5e18, {"from": owner})
    weth.approve(liquidity.address, 100e18, {"from": other})
    drops.approve(liquidity.address, 100e18, {"from": other})
    liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

    pair_addr = drops.uniswapV2Pair()
    pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
    pair.approve(locker.address, 10e18, {"from": other})
    pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

    # Lock tokens
    tx = locker.lock(other, pair_addr, True, pair.balanceOf(other), get_timestamp(),
"some_description", {"from": other})
    lock_id = tx.events['LockAdded'][0]['id']
    tx = market.initiateListing(pair_addr, lock_id, 1e18, 1e18, other, {"from":
other})
    assert tx.events['ListingInitiated'][0]['lpToken'] == pair_addr
    assert tx.events['ListingInitiated'][0]['lockID'] == lock_id
    assert tx.events['ListingInitiated'][0]['seller'] == other

    locker.transferLockOwnership(lock_id, market.address, {"from": other})
    market.activateListing(pair_addr, lock_id, {"from": other})

```

```

with reverts("Listing must be active."):
    market.buyLockWithDrops(ZERO_ADDRESS, lock_id, {"from": extra})
with reverts("Insufficient drops."):
    market.buyLockWithDrops(pair_addr, lock_id, {"from": extra})
drops.transfer(extra, 5e18, {"from": owner})
drops.approve(market.address, 1e18, {"from": extra})
tx = market.buyLockWithDrops(pair_addr, lock_id, {"from": extra})
assert tx.events['LockPurchasedWithDrops'][0]['lpToken'] == pair_addr
assert tx.events['LockPurchasedWithDrops'][0]['lockID'] == lock_id
assert tx.events['LockPurchasedWithDrops'][0]['profitInDrops'] == 1e18

with reverts("This listing does not belong to you."):
    market.withdrawListing(pair_addr, lock_id, {"from": extra})
with reverts("Marketplace does not own your lock"):
    market.withdrawListing(pair_addr, lock_id, {"from": other})
locker.transferLockOwnership(lock_id, market.address, {"from": extra})
tx = market.withdrawListing(pair_addr, lock_id, {"from": other})
assert tx.events['ListingWithdrawn'][0]['lpToken'] == pair_addr
assert tx.events['ListingWithdrawn'][0]['lockID'] == lock_id

```

```
def test_redact_listing(only_local):
```

```
    # Arrange
```

```
    owner = get_account(0)
```

```
    other = get_account(1)
```

```
    # Deploy contracts
```

```
    factory = deploy_factory(owner, owner)
```

```
    locker = deploy_pink_lock(owner)
```

```
    weth = deploy_weth(owner)
```

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

```
    liquidity = deploy_liquidity(owner, factory.address, router.address)
```

```
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
```

```
    market = deploy_pinksale_marketplace(owner, locker.address, owner, drops.address)
```

```
    # Add liquidity
```

```
    weth.deposit({"from": other, "value": 10e18})
```

```
    drops.setTradingOpen({"from": owner})
```

```
    drops.transfer(other, 5e18, {"from": owner})
```

```
    weth.approve(liquidity.address, 100e18, {"from": other})
```

```
    drops.approve(liquidity.address, 100e18, {"from": other})
```

```
    liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})
```

```
    pair_addr = drops.uniswapV2Pair()
```

```
    pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
```

```
    pair.approve(locker.address, 10e18, {"from": other})
```

```
    pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})
```

```
    # Lock tokens
```

```

    tx = locker.lock(other, pair_addr, True, pair.balanceOf(other), get_timestamp(),
"some_description", {"from": other})
    lock_id = tx.events['LockAdded'][0]['id']

    market.initiateListing(pair_addr, lock_id, 1e18, 1e18, other, {"from": other})
    locker.transferLockOwnership(lock_id, market.address, {"from": other})
    market.activateListing(pair_addr, lock_id, {"from": other})
    tx = market.redactListing(pair_addr, lock_id, {"from": owner})
    assert tx.events['ListingRedacted'][0]['lpToken'] == pair_addr
    assert tx.events['ListingRedacted'][0]['lockID'] == lock_id

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

    # Deploy contracts
    factory = deploy_factory(owner, owner)
    locker = deploy_pink_lock(owner)
    weth = deploy_weth(owner)
    router = deploy_router(owner, factory.address, weth.address)
    liquidity = deploy_liquidity(owner, factory.address, router.address)
    drops = deploy_drops(owner, router.address, 1000, 5, 5, 10e18, owner)
    market = deploy_pinksale_marketplace(owner, locker.address, owner, drops.address)

    # Add liquidity
    weth.deposit({"from": other, "value": 10e18})
    drops.setTradingOpen({"from": owner})
    drops.transfer(other, 5e18, {"from": owner})
    weth.approve(liquidity.address, 100e18, {"from": other})
    drops.approve(liquidity.address, 100e18, {"from": other})
    liquidity.addLiquidity(drops.address, weth.address, 1000e6, 1e18, {"from": other})

    pair_addr = drops.uniswapV2Pair()
    pair = Contract.from_abi("UniswapV2Pair", pair_addr, UniswapV2Pair.abi)
    pair.approve(locker.address, 10e18, {"from": other})
    pair.transfer(other, pair.balanceOf(liquidity), {"from": liquidity})

    # Lock tokens
    tx = locker.lock(other, pair_addr, True, pair.balanceOf(other), get_timestamp(),
"some_description", {"from": other})
    lock_id = tx.events['LockAdded'][0]['id']
    market.initiateListing(pair_addr, lock_id, 1e18, 1e18, other, {"from": other})

    with reverts("This listing does not belong to you."):
        market.changePriceInETH(pair_addr, lock_id, 2e18, {"from": extra})
    market.changePriceInETH(pair_addr, lock_id, 2e18, {"from": other})

```

```
with reverts("This listing does not belong to you."):
    market.changePriceInDrops(pair_addr, lock_id, 2e18, {"from": extra})
    market.changePriceInDrops(pair_addr, lock_id, 2e18, {"from": other})
```

Technical Findings Summary

Findings

Vulnerability Level	Total	Pending	Not Apply	Acknowledged	Partially Fixed	Fixed
<div><div></div>High</div>	1					1
<div><div></div>Medium</div>	0					
<div><div></div>Low</div>	0					
<div><div></div>Informational</div>	0					

Assessment Results

Score Results

Review	Score
Audit Score	90/100
Assure KYC	Pending
Audit Score	95/100

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 Drops project, we inform you that the project has met the necessary security standards and the issues reported were fixed on the deployed version.

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 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 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 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.