IMMUNEBYTES

Audits

ethernity

SMART CONTRACT AUDIT FINAL REPORT

April 2, 2022

TOC

TABLE OF CONTENTS

Introduction

1. About Ethernity

Ethernity is a Decentralized Application (DAPP) Platform that allows artists to create and auction artwork inspired and backed by celebrities for charity.
The concept behind Ethernity is mutually beneficial for all actors involved:

1. **Public Figure:** by making it easier to create, store, back, and sell the artworks.
2. **Charity:** by getting 100% of the first sale proceeds (minus exchange fees). And the auction format maximizes the artwork value (increasing the charity's benefits) without the need of a promoter, leveraging the emotions that a bidding war involves.
3. **Collector:** by providing them with an easy, democratized platform to bid on these pieces of authentic digital art where they can thereafter take bids and auction their acquired artwork.

With ERN tokens collectors can acquire Ethernity's exclusive authenticated NFTs as a payment method and also yield farming rewards. Part of the sales proceeds goes to charity.

Visit https://ethernity.io/ to know more about it.

2. About ImmuneBytes

ImmuneBytes is a security start-up to provide professional services in the blockchain space. The team has hands-on experience in conducting smart contract audits, penetration testing, and security consulting. ImmuneBytes's security auditors have worked on various A-league projects and have a great understanding of DeFi projects like AAVE, Compound, 0x Protocol, Uniswap, dydx.

The team has been able to secure 125+ blockchain projects by providing security services on different frameworks. ImmuneBytes team helps start-ups with a detailed analysis of the system ensuring security and managing the overall project.

Visit http://immunebytes.com/ to know more about the services.

Documentation Details

The Ethernity team has provided the following doc for the purpose of audit:

1. https://ethernity.cloud/whitepaper/ETHERNITY_whitepaper.pdf

Audit Process & Methodology

ImmuneBytes team has performed thorough testing of the project starting with analyzing the code design patterns in which we reviewed the smart contract architecture to ensure it is structured and safe use of third-party smart contracts and libraries.

Our team then performed a formal line-by-line inspection of the Smart Contract in order to find any potential issues like Signature Replay Attacks, Unchecked External Calls, External Contract Referencing, Variable Shadowing, Race conditions, Transaction-ordering dependence, timestamp dependence, DoS attacks, and others.

In the Unit testing phase, we run unit tests written by the developer in order to verify the functions work as intended. In Automated Testing, we tested the Smart Contract with our in-house developed tools to identify vulnerabilities and security flaws.

The code was audited by a team of independent auditors which includes -

1. Testing the functionality of the Smart Contract to determine proper logic has been followed throughout.
2. Analyzing the complexity of the code by thorough, manual review of the code, line-by-line.
3. Deploying the code on testnet using multiple clients to run live tests.
4. Analyzing failure preparations to check how the Smart Contract performs in case of bugs and vulnerabilities.
5. Checking whether all the libraries used in the code are on the latest version.
6. Analyzing the security of the on-chain data.

Audit Details

- Project Name: Ethernity
- Token Name: MysteryDrop.sol, EnumerableSet.sol
- GitHub Address: https://github.com/extrawatts/ethernity-mystery-drop
- Commit Hash for initial audit: f5ddf8355240cd79efb0a5b56d694fb6ae3a9e98
- Commit Hash for final audit: 7690d8b0ac45a69e444def141249d3c44df78026
- Languages: Solidity(Smart contract), Typescript (Unit Testing)
- Platforms and Tools: Remix IDE, Truffle, Truffle Team, Ganache, Solhint, VScode, Contract Library, Slither, SmartCheck, echinda

Audit Goals

The focus of the audit was to verify that the smart contract system is secure, resilient, and working according to its specifications. The audit activities can be grouped into the following three categories:

1. Security: Identifying security-related issues within each contract and within the system of contracts.
2. Sound Architecture: Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.
3. Code Correctness and Quality: A full review of the contract source code. The primary areas of focus include
    a. Correctness
    b. Readability
    c. Sections of code with high complexity
    d. Quantity and quality of test coverage

Security Level Reference

Every issue in this report were assigned a severity level from the following:

Admin/Owner Privileges can be misused either intentionally or unintentionally.
High severity issues will bring problems and should be fixed.
Medium severity issues could potentially bring problems and should eventually be fixed.
Low severity issues are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

Issues	High	Medium	Low
Open	-	-	-
Closed	1	2	3

Contract Name: Ethernity

High Severity Issues

1. Missing Authentication

Contract: MysteryDrop.sol and ThirdAlternative.sol

Description:
Some methods are missing proper authority check

157(MysteryDrop)	function set(address[] calldata _collections, uint256[] calldata numberofIds) external
40(ThirdAlternative)	function tierSet(uint16[] memory _tiers, uint256[] memory _prices) external
29(ThirdAlternative)	function deleteToken(uint16 _tier,uint256 _collectionIndex,uint256 _tokenIndex) public

Recommendation:
Add modifiers to check caller authority

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

Medium Severity Issues

1. Missing Reentrancy Guard

Contract: MysteryDrop.sol

Description:
The method transfers tokens from user to self after executing the buy which mints the token for the user before fetching the amount. After minting the ERC1155 contracts executes a `_afterTokenTransfer` method which can be overridden to create a reentrancy.

Line	Code/Function

204	function buyMysteryBox(address _user, Tiers _tier) external isStarted {
	require(_user == msg.sender,"Not user!");
	uint256 _ernAmount = buy(_user, _tier);
	ern.transferFrom(_user, address(this), _ernAmount);
	
	}

Recommendation:
Create or Import a nonRentrancy guard from OpenZeppelin and apply it to the method.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

2. Hardcoded Address

Contract: MysteryDrop.sol

Description:
The address of the Oracle has been hardcoded, which needs to change for different networks.

Line	Code/Function
40	address ernOracleAddr = 0x0a87e12689374A4EF49729582B474a1013cceBf8;

Recommendation:
Set the value for `ernOracleAddr` in the constructor so that it can be set on deployment whenever deploying to new network.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

Low severity issues

1. Unused Imports

Contract: MysteryDrop.sol

Description:
The following import was used in the contract MysteryDrop but is not used at all.

Line	Code/Function
6	import "@openzeppelin/contracts/token/ERC1155/IERC1155.sol";

Recommendation:
We should remove the unnecessary imports to reduce contract size and hence deployment costs.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

2. **Unused Mapping**

Contract: MysteryDrop.sol

Description:
The contract defines a mapping called tierTokens but it is not being used in the code.

Line	Code/Function
45	mapping(Tiers => mapping(address => uint256[])) public tierTokens;

Recommendation:
We should remove the used variable declarations to reduce contract size and hence deployment costs.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

3. **Misleading variable name**

Contract: MysteryDrop.sol

Description:
The mapping is called `tiers` but it maps tiers to tier prices.

Line	Code/Function
44	mapping(Tiers => uint256) public tiers;

Recommendation:
We can call the variable `tierPrices` for readability and understanding purposes.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

Recommendations/Informational

1. Typecasting on every call

Contract: MysteryDrop.sol

Description:
Whenever we make a call to `getPrice` there is always a type casting of `ernOracleAddr` as `AggregatorV3Interface` which costs gas.

Line	Code/Function
216	AggregatorV3Interface priceFeed = AggregatorV3Interface(ernOracleAddr);

Recommendation:
Since `ernOracleAddr` is not being used as an address in the contract, we can initialize it as `AggregatorV3Interface` itself so that we can skip the typecasting in `getPrice` and save some gas on every call.

Also, we can refactor the constructor on similar lines, i.e.
from `constructor(address _ern)` to `constructor(IERC20 _ern)`

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

2. Commented Code

Contract: MysteryDrop.sol

Description:
The contract contains instances of code that has been commented and contribute nothing to the logic.

Line	Code/Function
216	// return 1;

130,153	// uint256 count;
	// count++;

Recommendation:
Remove commented code.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

3. Refactoring buyMysteryBox

Contract: MysteryDrop.sol

Description:
The method takes user as parameter then ensures that user is msg.sender, so by that logic only msg.sender can call buyMysteryBox for themselves.

Line	Code/Function

204	function buyMysteryBox(address _user, Tiers _tier) external isStarted {
	require(_user == msg.sender,"Not user!");
	uint256 _ernAmount = buy(_user, _tier);
	ern.transferFrom(_user, address(this), _ernAmount);
	}

Recommendation:

Code/Function

function buyMysteryBox(Tiers _tier) external isStarted {
uint256 _ernAmount = buy(msg.sender, _tier);
ern.transferFrom(msg.sender, address(this), _ernAmount);
}

We can skip getting the user value as a parameter itself and hence also also skip the require check for the same and use msg.sender directly.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

4. Similar code between two methods

Contract: MysteryDrop.sol

Description:
The methods `setCollectionsBatch` and `setCollections` share similar code.

Recommendation:
We recommend making an internal method and make a call to it from both methods to avoid writing repeated code

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

5. Incorrect naming convention

Contract: MysteryDrop.sol

Description:
The method `buy` is an internal function but appears to be a public or external function.

Recommendation:
The internal function names should be preceded by an underscore, so the method `buy` can be renamed as `_buy` hence following the naming conventions of solidity.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

6. Missing netspec comments

Recommendation:
We recommend adding netspec comments for each method and variables for better readability and understanding of code.

Amended (April 02, 2022): The issue has been fixed by the Ethernity team and is no longer present in the commit: 7690d8b0ac45a69e444def141249d3c44df78026

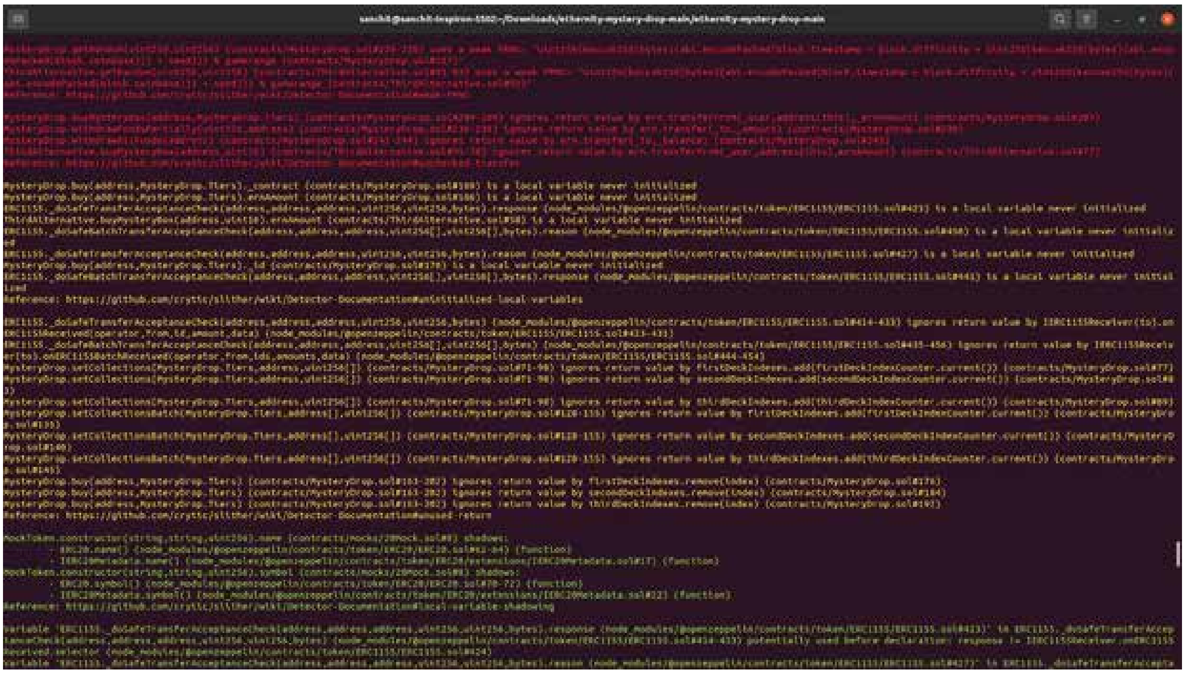
Functional Tests (Goerli testnet)

GameItems: 0x9522496Ed5887FF5fA82c6fD3bE3e0976de4D0b6
GameItems: 0x1CBf065E75C7f81cfa28B082A09B18744fe64e43
GameItems: 0x3d2AAA6C0ebD73EA2e8f694b777CC150d610Dda0
MockERC20: 0x8eab9046c03FbFF69f6274325dF65bEda2A98f62
MysteryDrop: 0x96C63fdf59703dc5c4f56567271eA530010332d9

tierSet	0x7c9a986f90f7882f7df1a7b078b18e57f562582ecc4b4e1704b6e5df811159d9	Pass
setCollection (1 tier, 2 tier, 3 tier)	0x045235b8dde4c1eb0d0068e701de4bd22f11e922be1a1e664eb27fb87a443ed5 0x90dc2332eb2eecda03978be05703938d6b52a636a4dbfb54b9995212865d4a42 0x53366c3a16d7b19c2b880259e78331416cee041f11d4fe24bf01390a2e989bbd 0xec5ee3ea6a99c75021b8432306faf506b377f56b2cbcc3652140ea41e684845f 0xda936278cdc5f7d2cb94145acbf9710d29fc1f15435b019b04ed8777496f2fff 0x827ea6107e77039981cd3c66ccce000da10e7ca906b34292376e1b2ebd5a7104 0xe254b42e3bb8ab8de09ab669002408519fa1e4cb8a6e398a381aa819216422d7 0xa3d10b72103bbe5e3eb5741861524c859818e439e66cc695eed5fc509ec0080b	Pass
setStart	0x0d300ee029b49040ed93777bb32b0d584c94fbe63cc009da990ea71857f0599c	Pass
buyMysteryBox	0x0480e43e01956384aca287fde3621a2629b97b1efa8ec21dc1ace8b0cf7695aa	Pass
withdrawFundsPartially	0xe0b1e2b13e2ea475d1c8b3d1fba7eb8dc9caea3266f9b716c9abc7ee1efebc1e	Pass
withdrawAllFunds	0xc22b99248c7f84be521bd6a2935d62fec936ea46b66a17538bcaa8235b92e982	Pass
buyCreditMysteryBox	0xd94ac48789150b361db3a0af8b9ccd02fbd8a9b13374da96138060bc15bfb2d2	Pass
resetTierDeck	0x8c7e5c7827de7e055bd17ed13475c29beb75b5e53b723124a871e7de8781e460	Pass

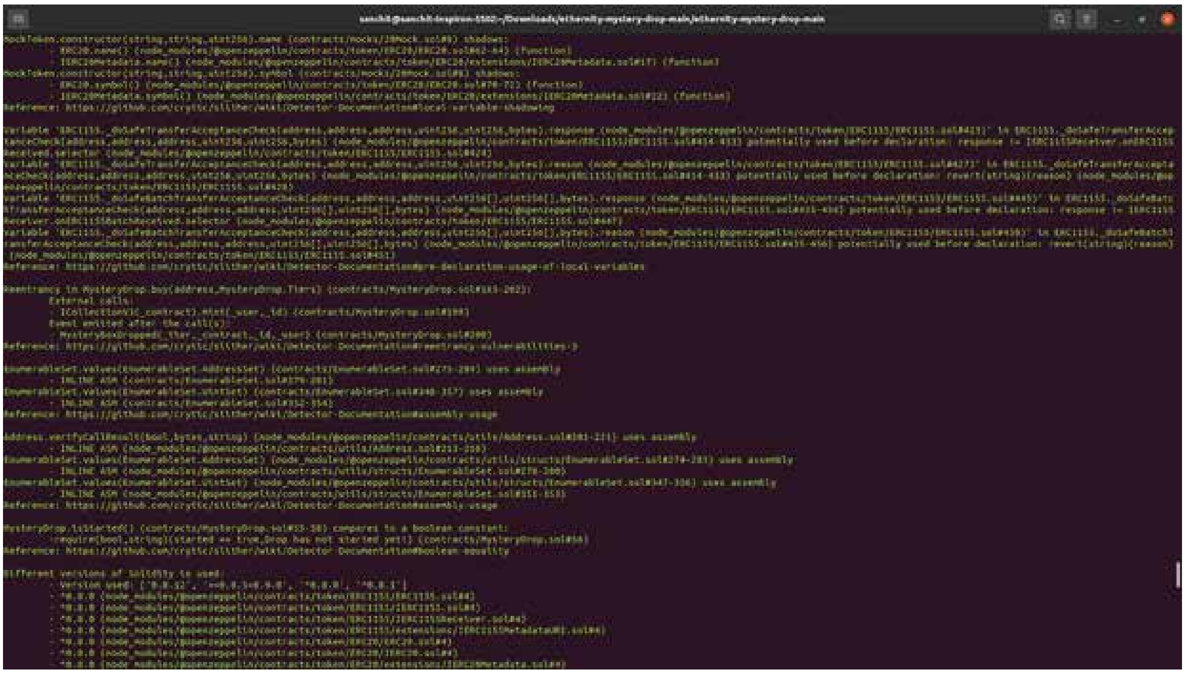
Automated Tools Result

1. **Slither**



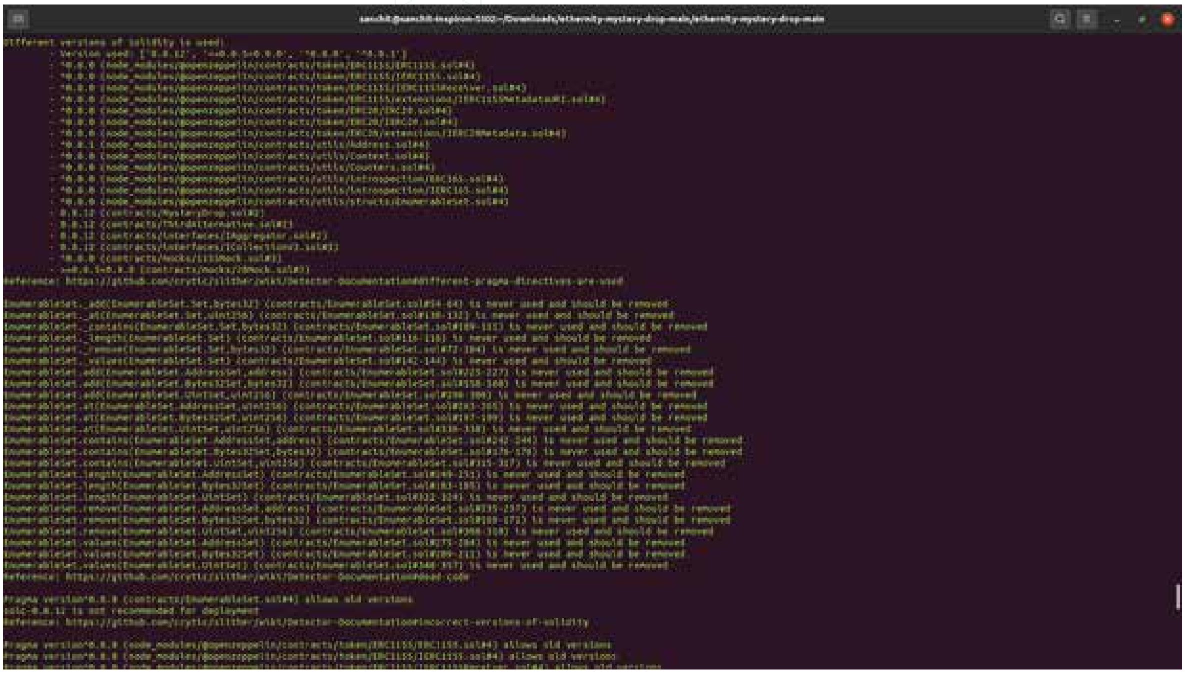
sanchit@sanchit-Inspiron-5502: ~/Downloads/ethernity-mystery-drop-main/ethernity-mystery-drop-main

```
MysteryDrop.getRandom(uint256,uint256) (contracts/MysteryDrop.sol#[illegible]) uses a weak PRNG: "uint256(keccak256(bytes)(abi.encodePacked(block.timestamp + block.difficulty + (uint256(keccak256(bytes)(abi.enco
dePacked(block.coinbase))) [illegible] seed)) % gameRange (contracts/MysteryDrop.sol#[illegible])"
ThirdAlternative.getRandom(uint256,uint256) (contracts/ThirdAlternative.sol#[illegible]) uses a weak PRNG: "uint256(keccak256(bytes)(abi.encodePacked(block.timestamp + block.difficulty + ((uint256(keccak256(bytes)(
abi.encodePacked(block.coinbase))) + seed)) % gameRange (contracts/ThirdAlternative.sol#[illegible])"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

MysteryDrop.buyMysteryBox(address,MysteryDrop.Tiers) (contracts/MysteryDrop.sol#[illegible]) ignores return value by erc.transferFrom(_user,address(this),_erAmount) (contracts/MysteryDrop.sol#[illegible])
MysteryDrop.[illegible] ignores return value by erc.transfer(_to,_amount) (contracts/MysteryDrop.sol#[illegible])
MysteryDrop.[illegible] ignores return value by erc.transfer(_to,_balance) (contracts/MysteryDrop.sol#[illegible])
ThirdAlternative.buyMysteryBox(address,uint8) (contracts/ThirdAlternative.sol#[illegible]) ignores return value by erc.transferFrom(_user,address(this),erAmount) (contracts/ThirdAlternative.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

MysteryDrop.buy(address,MysteryDrop.Tiers)._contract (contracts/MysteryDrop.sol#180) is a local variable never initialized
MysteryDrop.buy(address,MysteryDrop.Tiers).erAmount (contracts/MysteryDrop.sol#186) is a local variable never initialized
ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#421) is a local variable never initialized
ThirdAlternative.buyMysteryBox(address,uint16).erAmount (contracts/ThirdAlternative.sol#58) is a local variable never initialized
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#450) is a local variable never initializ
ed
ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#427) is a local variable never initialized
MysteryDrop.buy(address,MysteryDrop.Tiers)._id (contracts/MysteryDrop.sol#178) is a local variable never initialized
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#445) is a local variable never initial
ized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#414-433) ignores return value by IERC1155Receiver(to).on
ERC1155Received(operator,from,id,amount,data) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#423-431)
ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#435-456) ignores return value by IERC1155Receiv
er(to).onERC1155BatchReceived(operator,from,ids,amounts,data) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#444-454)
MysteryDrop.setCollections(MysteryDrop.Tiers,address,uint256[]) (contracts/MysteryDrop.sol#71-98) ignores return value by firstDeckIndexes.add(firstDeckIndexCounter.current()) (contracts/MysteryDrop.sol#77)
MysteryDrop.setCollections(MysteryDrop.Tiers,address,uint256[]) (contracts/MysteryDrop.sol#71-98) ignores return value by secondDeckIndexes.add(secondDeckIndexCounter.current()) (contracts/MysteryDrop.sol#8
3)
MysteryDrop.setCollections(MysteryDrop.Tiers,address,uint256[]) (contracts/MysteryDrop.sol#71-98) ignores return value by thirdDeckIndexes.add(thirdDeckIndexCounter.current()) (contracts/MysteryDrop.sol#89)
MysteryDrop.setCollectionsBatch(MysteryDrop.Tiers,address[],uint256[]) (contracts/MysteryDrop.sol#120-155) ignores return value by firstDeckIndexes.add(firstDeckIndexCounter.current()) (contracts/MysteryDro
p.sol#135)
MysteryDrop.setCollectionsBatch(MysteryDrop.Tiers,address[],uint256[]) (contracts/MysteryDrop.sol#120-155) ignores return value by secondDeckIndexes.add(secondDeckIndexCounter.current()) (contracts/MysteryD
rop.sol#140)
MysteryDrop.setCollectionsBatch(MysteryDrop.Tiers,address[],uint256[]) (contracts/MysteryDrop.sol#120-155) ignores return value by thirdDeckIndexes.add(thirdDeckIndexCounter.current()) (contracts/MysteryDro
p.sol#145)
MysteryDrop.buy(address,MysteryDrop.Tiers) (contracts/MysteryDrop.sol#163-202) ignores return value by firstDeckIndexes.remove(index) (contracts/MysteryDrop.sol#176)
MysteryDrop.buy(address,MysteryDrop.Tiers) (contracts/MysteryDrop.sol#163-202) ignores return value by secondDeckIndexes.remove(index) (contracts/MysteryDrop.sol#184)
MysteryDrop.buy(address,MysteryDrop.Tiers) (contracts/MysteryDrop.sol#163-202) ignores return value by thirdDeckIndexes.remove(index) (contracts/MysteryDrop.sol#192)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

MockToken.constructor(string,string,uint256).name (contracts/mocks/20Mock.sol#8) shadows:
        - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#62-64) (function)
        - IERC20Metadata.name() (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#17) (function)
MockToken.constructor(string,string,uint256).symbol (contracts/mocks/20Mock.sol#8) shadows:
        - ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#70-72) (function)
        - IERC20Metadata.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#22) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#421)' in ERC1155._doSafeTransferAccep
tanceCheck(address,address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#414-433) potentially used before declaration: response != IERC1155Receiver.onERC1155
Received.selector (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#424)
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#427)' in ERC1155._doSafeTransferAccepta
```



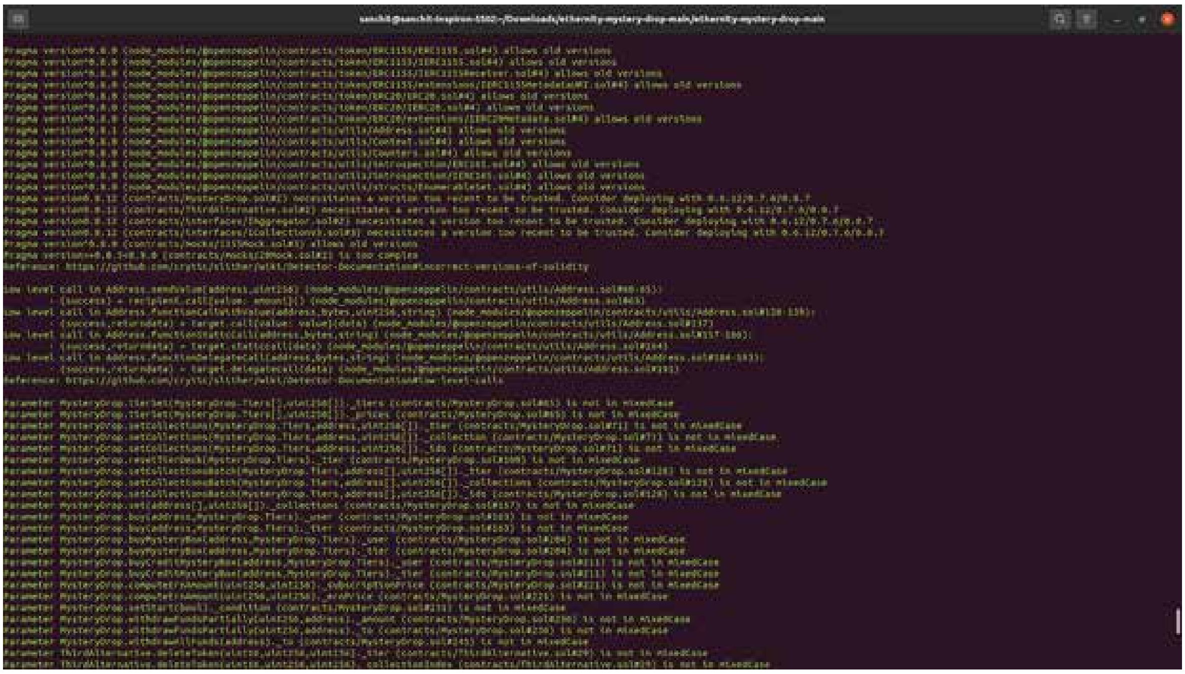
sanchit@sanchit-Inspiron-5502: ~/Downloads/ethernity-mystery-drop-main/ethernity-mystery-drop-main

```
MockToken.constructor(string,string,uint256).name (contracts/mocks/20Mock.sol#8) shadows:
        - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#62-64) (function)
        - IERC20Metadata.name() (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#17) (function)
MockToken.constructor(string,string,uint256).symbol (contracts/mocks/20Mock.sol#8) shadows:
        - ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#70-72) (function)
        - IERC20Metadata.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#22) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#421)' in ERC1155._doSafeTransferAccep
tanceCheck(address,address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#414-433) potentially used before declaration: response != IERC1155Receiver.onERC1155
Received.selector (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#424)
Variable 'ERC1155._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#427)' in ERC1155._doSafeTransferAccepta
nceCheck(address,address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#414-433) potentially used before declaration: revert(string)(reason) (node_modules/@op
enzeppelin/contracts/token/ERC1155/ERC1155.sol#428)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#445)' in ERC1155._doSafeBatc
hTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#435-456) potentially used before declaration: response != IERC1155
Receiver.onERC1155BatchReceived.selector (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#447)
Variable 'ERC1155._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#450)' in ERC1155._doSafeBatchT
ransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#435-456) potentially used before declaration: revert(string)(reason)
 (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#451)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in MysteryDrop.buy(address,MysteryDrop.Tiers) (contracts/MysteryDrop.sol#163-202):
        External calls:
        - ICollectionV1C(_contract).mint(_user,_id) (contracts/MysteryDrop.sol#199)
        Event emitted after the call(s):
        - MysteryBoxDropped(_tier,_contract,_id,_user) (contracts/MysteryDrop.sol#200)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EnumerableSet.values(EnumerableSet.AddressSet) (contracts/EnumerableSet.sol#275-284) uses assembly
        - INLINE ASM (contracts/EnumerableSet.sol#279-281)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/EnumerableSet.sol#348-357) uses assembly
        - INLINE ASM (contracts/EnumerableSet.sol#352-354)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
EnumerableSet.values(EnumerableSet.AddressSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#274-283) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#278-280)
EnumerableSet.values(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#347-356) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#351-353)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

MysteryDrop.isStarted() (contracts/MysteryDrop.sol#53-58) compares to a boolean constant:
        -require(bool,string)(started == true,Drop has not started yet!) (contracts/MysteryDrop.sol#56)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['0.8.12', '>=0.8.1<0.9.0', '^0.8.0', '^0.8.1']
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/extensions/IERC1155MetadataURI.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
```



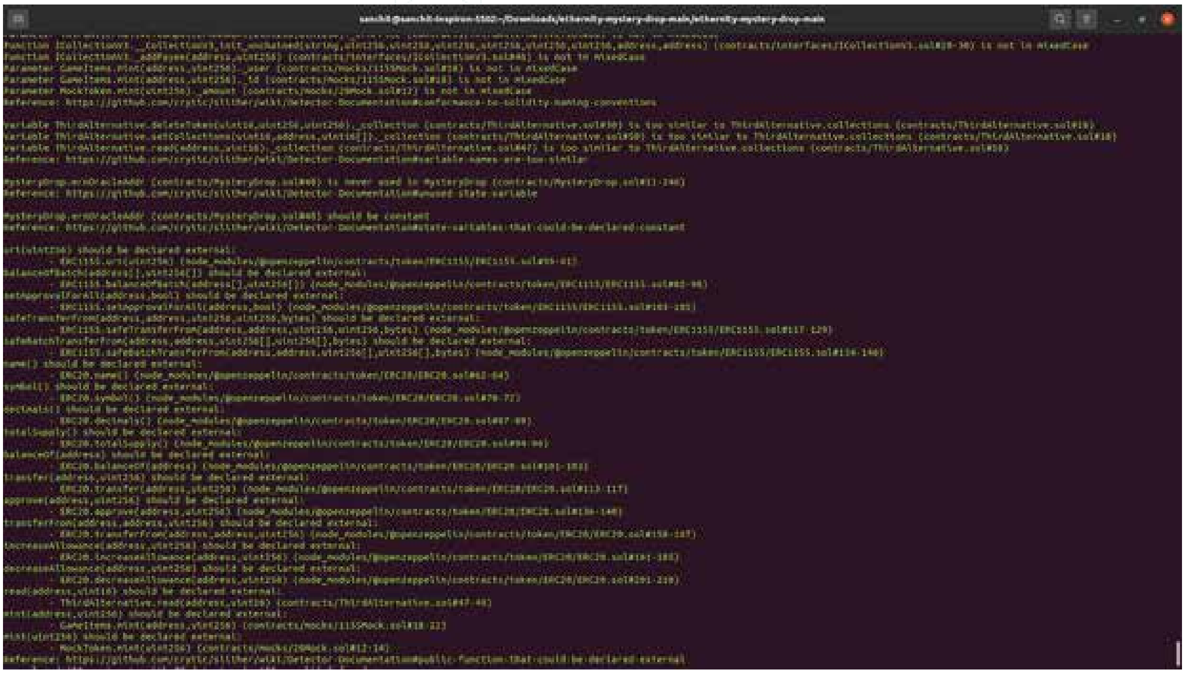
sanchit@sanchit-Inspiron-5502: ~/Downloads/ethernity-mystery-drop-main/ethernity-mystery-drop-main

```
Different versions of Solidity is used:
        - Version used: ['0.8.12', '>=0.6.1<0.9.0', '^0.8.0', '^0.8.1']
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/extensions/IERC1155MetadataURI.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/ERC165.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#4)
        - 0.8.12 (contracts/MysteryDrop.sol#2)
        - 0.8.12 (contracts/ThirdAlternative.sol#2)
        - 0.8.12 (contracts/interfaces/IAggregator.sol#2)
        - 0.8.12 (contracts/interfaces/ICollectionV3.sol#3)
        - ^0.8.0 (contracts/mocks/1155Mock.sol#3)
        - >=0.6.1<0.9.0 (contracts/mocks/20Mock.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

EnumerableSet._add(EnumerableSet.Set,bytes32) (contracts/EnumerableSet.sol#64-74) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (contracts/EnumerableSet.sol#130-132) is never used and should be removed
EnumerableSet._contains(EnumerableSet.Set,bytes32) (contracts/EnumerableSet.sol#109-111) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (contracts/EnumerableSet.sol#116-118) is never used and should be removed
EnumerableSet._remove(EnumerableSet.Set,bytes32) (contracts/EnumerableSet.sol#72-104) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/EnumerableSet.sol#142-144) is never used and should be removed
EnumerableSet.add(EnumerableSet.AddressSet,address) (contracts/EnumerableSet.sol#225-227) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/EnumerableSet.sol#158-160) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/EnumerableSet.sol#298-300) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (contracts/EnumerableSet.sol#263-265) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/EnumerableSet.sol#197-199) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/EnumerableSet.sol#336-338) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/EnumerableSet.sol#242-244) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/EnumerableSet.sol#176-178) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/EnumerableSet.sol#315-317) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (contracts/EnumerableSet.sol#249-251) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/EnumerableSet.sol#183-185) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/EnumerableSet.sol#322-324) is never used and should be removed
EnumerableSet.remove(EnumerableSet.AddressSet,address) (contracts/EnumerableSet.sol#235-237) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/EnumerableSet.sol#169-171) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/EnumerableSet.sol#308-310) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/EnumerableSet.sol#275-284) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/EnumerableSet.sol#209-211) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/EnumerableSet.sol#348-357) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (contracts/EnumerableSet.sol#4) allows old versions
solc-0.8.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol#4) allows old versions
```



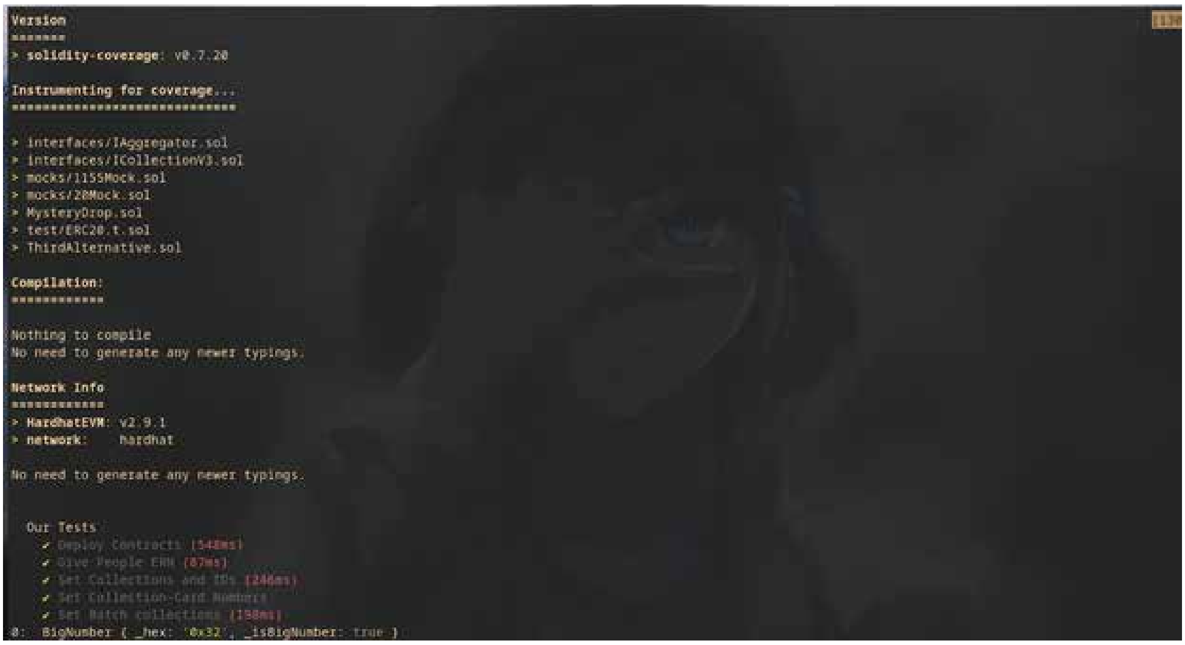
sanchit@sanchit-Inspiron-5502: ~/Downloads/ethernity-mystery-drop-main/ethernity-mystery-drop-main

```
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155Receiver.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/extensions/IERC1155MetadataURI.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/ERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#4) allows old versions
Pragma version0.8.12 (contracts/MysteryDrop.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.12 (contracts/ThirdAlternative.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.12 (contracts/interfaces/IAggregator.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.12 (contracts/interfaces/ICollectionV3.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/mocks/1155Mock.sol#3) allows old versions
Pragma version>=0.6.1<0.9.0 (contracts/mocks/20Mock.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter MysteryDrop.tierSet(MysteryDrop.Tiers[],uint256[])._tiers (contracts/MysteryDrop.sol#65) is not in mixedCase
Parameter MysteryDrop.tierSet(MysteryDrop.Tiers[],uint256[])._prices (contracts/MysteryDrop.sol#65) is not in mixedCase
Parameter MysteryDrop.setCollections(MysteryDrop.Tiers,address,uint256[])._tier (contracts/MysteryDrop.sol#71) is not in mixedCase
Parameter MysteryDrop.setCollections(MysteryDrop.Tiers,address,uint256[])._collection (contracts/MysteryDrop.sol#71) is not in mixedCase
Parameter MysteryDrop.setCollections(MysteryDrop.Tiers,address,uint256[])._ids (contracts/MysteryDrop.sol#71) is not in mixedCase
Parameter MysteryDrop.resetTierDeck(MysteryDrop.Tiers)._tier (contracts/MysteryDrop.sol#100) is not in mixedCase
Parameter MysteryDrop.setCollectionsBatch(MysteryDrop.Tiers,address[],uint256[][])._tier (contracts/MysteryDrop.sol#128) is not in mixedCase
Parameter MysteryDrop.setCollectionsBatch(MysteryDrop.Tiers,address[],uint256[][])._collections (contracts/MysteryDrop.sol#128) is not in mixedCase
Parameter MysteryDrop.setCollectionsBatch(MysteryDrop.Tiers,address[],uint256[][])._ids (contracts/MysteryDrop.sol#128) is not in mixedCase
Parameter MysteryDrop.set(address[],uint256[])._collections (contracts/MysteryDrop.sol#157) is not in mixedCase
Parameter MysteryDrop.buy(address,MysteryDrop.Tiers)._user (contracts/MysteryDrop.sol#163) is not in mixedCase
Parameter MysteryDrop.buy(address,MysteryDrop.Tiers)._tier (contracts/MysteryDrop.sol#163) is not in mixedCase
Parameter MysteryDrop.buyMysteryBox(address,MysteryDrop.Tiers)._user (contracts/MysteryDrop.sol#204) is not in mixedCase
Parameter MysteryDrop.buyMysteryBox(address,MysteryDrop.Tiers)._tier (contracts/MysteryDrop.sol#204) is not in mixedCase
Parameter MysteryDrop.buyCreditMysteryBox(address,MysteryDrop.Tiers)._user (contracts/MysteryDrop.sol#211) is not in mixedCase
Parameter MysteryDrop.buyCreditMysteryBox(address,MysteryDrop.Tiers)._tier (contracts/MysteryDrop.sol#211) is not in mixedCase
Parameter MysteryDrop.computeEthAmount(uint256,uint256)._subscriptionPrice (contracts/MysteryDrop.sol#221) is not in mixedCase
Parameter MysteryDrop.computeEthAmount(uint256,uint256)._ethPrice (contracts/MysteryDrop.sol#221) is not in mixedCase
Parameter MysteryDrop.setStart(bool)._condition (contracts/MysteryDrop.sol#233) is not in mixedCase
Parameter MysteryDrop.withdrawFundsPartially(uint256,address)._amount (contracts/MysteryDrop.sol#236) is not in mixedCase
Parameter MysteryDrop.withdrawFundsPartially(uint256,address)._to (contracts/MysteryDrop.sol#236) is not in mixedCase
Parameter MysteryDrop.withdrawAllFunds(address)._to (contracts/MysteryDrop.sol#245) is not in mixedCase
Parameter ThirdAlternative.deleteToken(uint256,uint256,uint256)._tier (contracts/ThirdAlternative.sol#29) is not in mixedCase
Parameter ThirdAlternative.deleteToken(uint256,uint256,uint256)._collectionIndex (contracts/ThirdAlternative.sol#29) is not in mixedCase
```

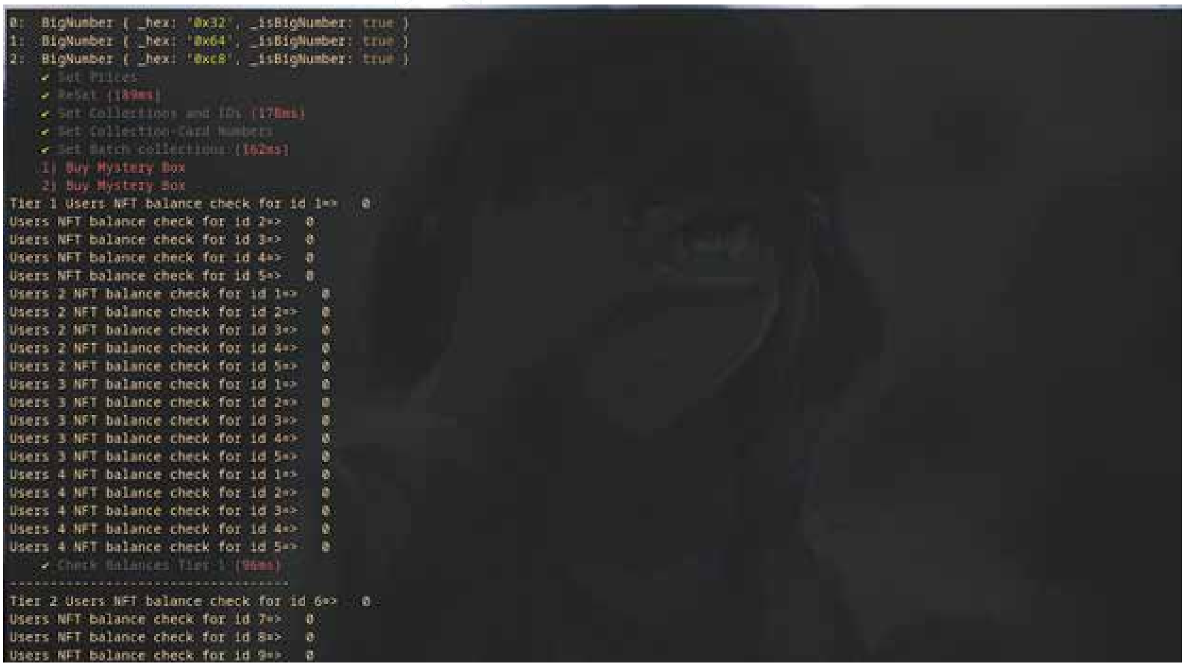


```
Function ICollectionV3._CollectionV3_init_unchained(string,uint256,uint256,uint256,uint256,uint256,uint256,address,address) (contracts/interfaces/ICollectionV3.sol#20-30) is not in mixedCase
Function ICollectionV3.__addPayee(address,uint256) (contracts/interfaces/ICollectionV3.sol#46) is not in mixedCase
Parameter GameItems.mint(address,uint256)._user (contracts/mocks/1155Mock.sol#18) is not in mixedCase
Parameter GameItems.mint(address,uint256)._id (contracts/mocks/1155Mock.sol#18) is not in mixedCase
Parameter MockToken.mint(uint256)._amount (contracts/mocks/20Mock.sol#12) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable ThirdAlternative.deleteToken(uint16,uint256,uint256)._collection (contracts/ThirdAlternative.sol#30) is too similar to ThirdAlternative.collections (contracts/ThirdAlternative.sol#18)
Variable ThirdAlternative.setCollections(uint16,address,uint16[])._collection (contracts/ThirdAlternative.sol#50) is too similar to ThirdAlternative.collections (contracts/ThirdAlternative.sol#18)
Variable ThirdAlternative.read(address,uint16)._collection (contracts/ThirdAlternative.sol#47) is too similar to ThirdAlternative.collections (contracts/ThirdAlternative.sol#18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

MysteryDrop.mcnOracleAddr (contracts/MysteryDrop.sol#88) is never used in MysteryDrop (contracts/MysteryDrop.sol#11-240)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

MysteryDrop.mcnOracleAddr (contracts/MysteryDrop.sol#88) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

uri(uint256) should be declared external:
        - ERC1155.uri(uint256) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#59-61)
balanceOfBatch(address[],uint256[]) should be declared external:
        - ERC1155.balanceOfBatch(address[],uint256[]) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#82-98)
setApprovalForAll(address,bool) should be declared external:
        - ERC1155.setApprovalForAll(address,bool) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#103-105)
safeTransferFrom(address,address,uint256,uint256,bytes) should be declared external:
        - ERC1155.safeTransferFrom(address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#117-129)
safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) should be declared external:
        - ERC1155.safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts/token/ERC1155/ERC1155.sol#134-146)
name() should be declared external:
        - ERC20.name() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#62-64)
symbol() should be declared external:
        - ERC20.symbol() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#70-72)
decimals() should be declared external:
        - ERC20.decimals() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#87-89)
totalSupply() should be declared external:
        - ERC20.totalSupply() (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#94-96)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#101-103)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#113-117)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#136-140)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#158-167)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#181-185)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol#201-210)
read(address,uint16) should be declared external:
        - ThirdAlternative.read(address,uint16) (contracts/ThirdAlternative.sol#47-49)
mint(address,uint256) should be declared external:
        - GameItems.mint(address,uint256) (contracts/mocks/1155Mock.sol#18-22)
mint(uint256) should be declared external:
        - MockToken.mint(uint256) (contracts/mocks/20Mock.sol#12-14)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

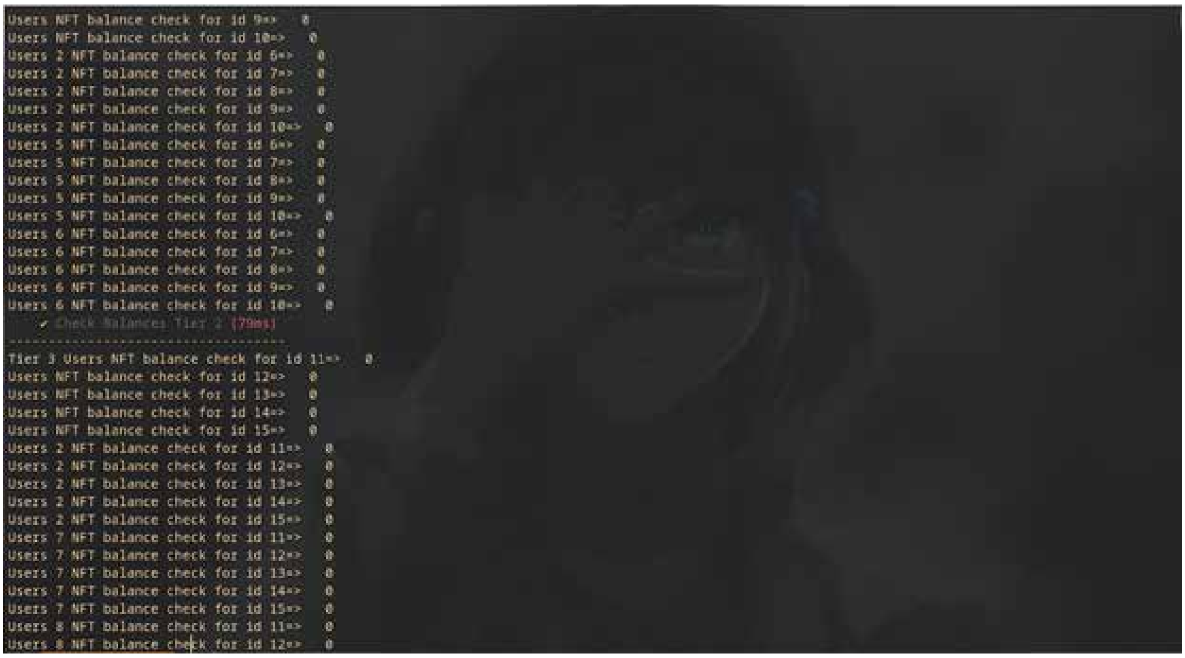
2. **Code Coverage**



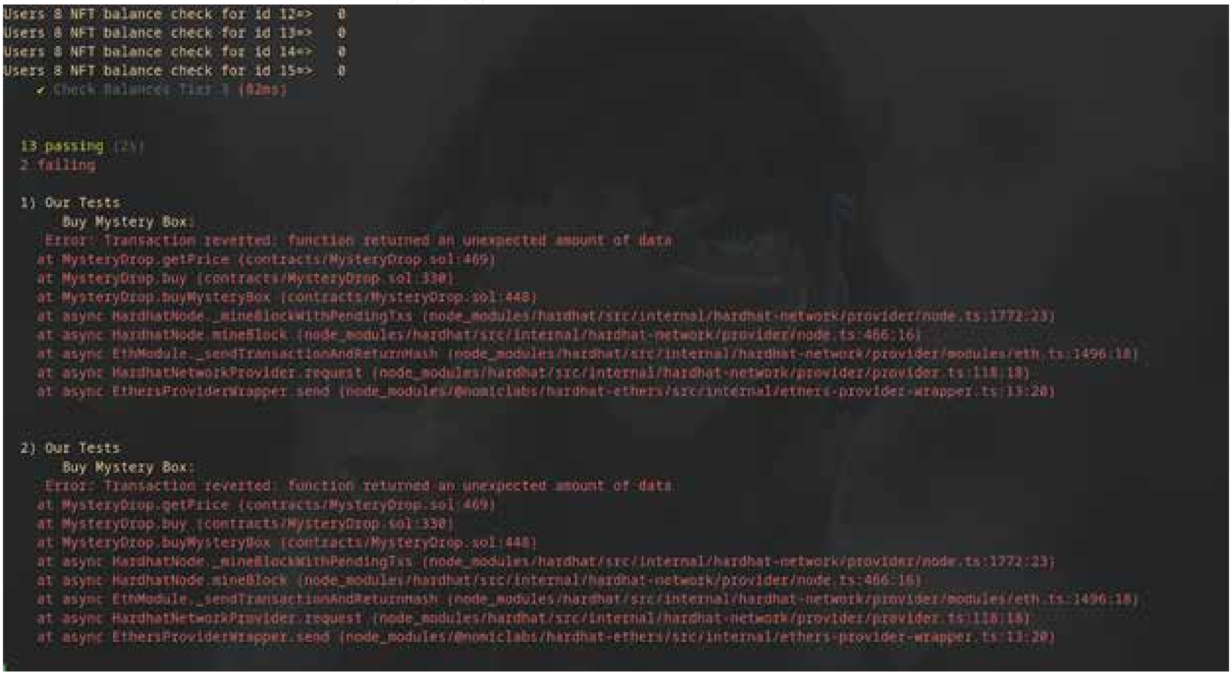
```
Version
=======
> solidity-coverage: v0.7.20

Instrumenting for coverage...
=============================

> interfaces/IAggregator.sol
> interfaces/ICollectionV3.sol
> mocks/1155Mock.sol
> mocks/20Mock.sol
> MysteryDrop.sol
> test/ERC20.t.sol
> ThirdAlternative.sol

Compilation:
============

Nothing to compile
No need to generate any newer typings.

Network Info
============
> HardhatEVM: v2.9.1
> network:    hardhat

No need to generate any newer typings.


  Our Tests
    ✔ Deploy Contracts (548ms)
    ✔ Give People ERN (87ms)
    ✔ Set Collections and IDs (246ms)
    ✔ Set Collection-Card Numbers
    ✔ Set Batch collections (198ms)
0:  BigNumber { _hex: '0x32', _isBigNumber: true }
```



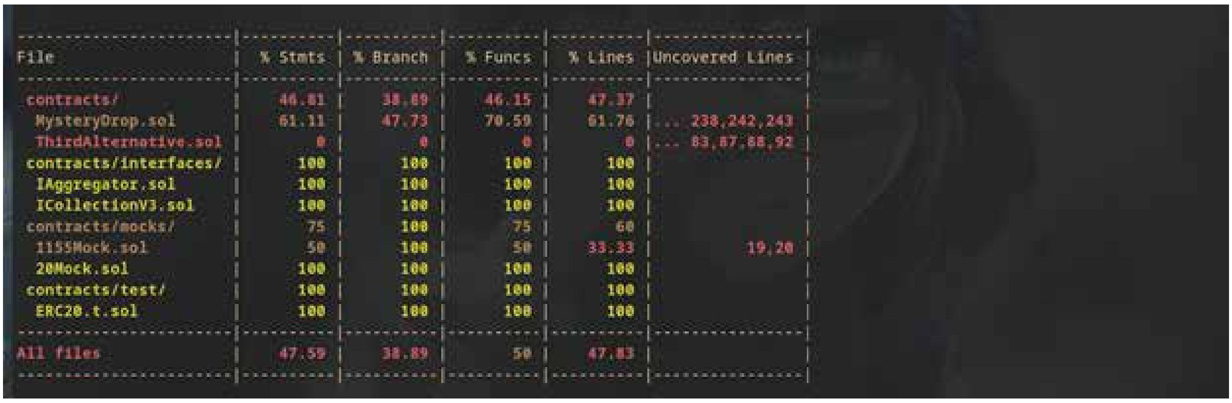
```
0:  BigNumber { _hex: '0x32', _isBigNumber: true }
1:  BigNumber { _hex: '0x64', _isBigNumber: true }
2:  BigNumber { _hex: '0xc8', _isBigNumber: true }
    ✓ Set Prices
    ✓ ReSet (189ms)
    ✓ Set Collections and IDs (178ms)
    ✓ Set Collection Card Numbers
    ✓ Set Batch collections (162ms)
    1) Buy Mystery Box
    2) Buy Mystery Box
Tier 1 Users NFT balance check for id 1=>   0
Users NFT balance check for id 2=>   0
Users NFT balance check for id 3=>   0
Users NFT balance check for id 4=>   0
Users NFT balance check for id 5=>   0
Users 2 NFT balance check for id 1=>   0
Users 2 NFT balance check for id 2=>   0
Users 2 NFT balance check for id 3=>   0
Users 2 NFT balance check for id 4=>   0
Users 2 NFT balance check for id 5=>   0
Users 3 NFT balance check for id 1=>   0
Users 3 NFT balance check for id 2=>   0
Users 3 NFT balance check for id 3=>   0
Users 3 NFT balance check for id 4=>   0
Users 3 NFT balance check for id 5=>   0
Users 4 NFT balance check for id 1=>   0
Users 4 NFT balance check for id 2=>   0
Users 4 NFT balance check for id 3=>   0
Users 4 NFT balance check for id 4=>   0
Users 4 NFT balance check for id 5=>   0
    ✓ Check Balances Tier 1 (96ms)
-----------------------------------
Tier 2 Users NFT balance check for id 6=>   0
Users NFT balance check for id 7=>   0
Users NFT balance check for id 8=>   0
Users NFT balance check for id 9=>   0
```



```
Users NFT balance check for id 9=>   0
Users NFT balance check for id 10=>   0
Users 2 NFT balance check for id 6=>   0
Users 2 NFT balance check for id 7=>   0
Users 2 NFT balance check for id 8=>   0
Users 2 NFT balance check for id 9=>   0
Users 2 NFT balance check for id 10=>   0
Users 5 NFT balance check for id 6=>   0
Users 5 NFT balance check for id 7=>   0
Users 5 NFT balance check for id 8=>   0
Users 5 NFT balance check for id 9=>   0
Users 5 NFT balance check for id 10=>   0
Users 6 NFT balance check for id 6=>   0
Users 6 NFT balance check for id 7=>   0
Users 6 NFT balance check for id 8=>   0
Users 6 NFT balance check for id 9=>   0
Users 6 NFT balance check for id 10=>   0
    ✓ Check Balances Tier 2 (79ms)
-----------------------------------
Tier 3 Users NFT balance check for id 11=>   0
Users NFT balance check for id 12=>   0
Users NFT balance check for id 13=>   0
Users NFT balance check for id 14=>   0
Users NFT balance check for id 15=>   0
Users 2 NFT balance check for id 11=>   0
Users 2 NFT balance check for id 12=>   0
Users 2 NFT balance check for id 13=>   0
Users 2 NFT balance check for id 14=>   0
Users 2 NFT balance check for id 15=>   0
Users 7 NFT balance check for id 11=>   0
Users 7 NFT balance check for id 12=>   0
Users 7 NFT balance check for id 13=>   0
Users 7 NFT balance check for id 14=>   0
Users 7 NFT balance check for id 15=>   0
Users 8 NFT balance check for id 11=>   0
Users 8 NFT balance check for id 12=>   0
```

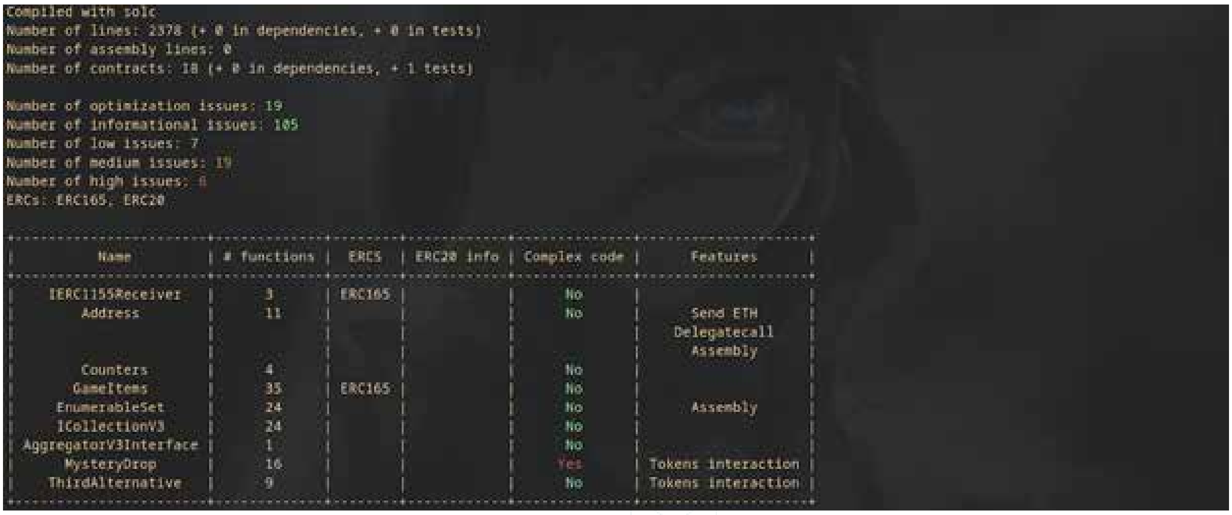


```
Users 8 NFT balance check for id 12=>   0
Users 8 NFT balance check for id 13=>   0
Users 8 NFT balance check for id 14=>   0
Users 8 NFT balance check for id 15=>   0
    ✔ Check Balances Tier 3 (82ms)


  13 passing (2s)
  2 failing

  1) Our Tests
       Buy Mystery Box:
     Error: Transaction reverted: function returned an unexpected amount of data
    at MysteryDrop.getPrice (contracts/MysteryDrop.sol:469)
    at MysteryDrop.buy (contracts/MysteryDrop.sol:330)
    at MysteryDrop.buyMysteryBox (contracts/MysteryDrop.sol:448)
    at async HardhatNode._mineBlockWithPendingTxs (node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:1772:23)
    at async HardhatNode.mineBlock (node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:466:16)
    at async EthModule._sendTransactionAndReturnHash (node_modules/hardhat/src/internal/hardhat-network/provider/modules/eth.ts:1496:18)
    at async HardhatNetworkProvider.request (node_modules/hardhat/src/internal/hardhat-network/provider/provider.ts:118:18)
    at async EthersProviderWrapper.send (node_modules/@nomiclabs/hardhat-ethers/src/internal/ethers-provider-wrapper.ts:13:20)


  2) Our Tests
       Buy Mystery Box:
     Error: Transaction reverted: function returned an unexpected amount of data
    at MysteryDrop.getPrice (contracts/MysteryDrop.sol:469)
    at MysteryDrop.buy (contracts/MysteryDrop.sol:330)
    at MysteryDrop.buyMysteryBox (contracts/MysteryDrop.sol:448)
    at async HardhatNode._mineBlockWithPendingTxs (node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:1772:23)
    at async HardhatNode.mineBlock (node_modules/hardhat/src/internal/hardhat-network/provider/node.ts:466:16)
    at async EthModule._sendTransactionAndReturnHash (node_modules/hardhat/src/internal/hardhat-network/provider/modules/eth.ts:1496:18)
    at async HardhatNetworkProvider.request (node_modules/hardhat/src/internal/hardhat-network/provider/provider.ts:118:18)
    at async EthersProviderWrapper.send (node_modules/@nomiclabs/hardhat-ethers/src/internal/ethers-provider-wrapper.ts:13:20)
```



```
------------------------|----------|----------|----------|----------|----------------|
File                    |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
------------------------|----------|----------|----------|----------|----------------|
 contracts/             |    46.81 |    38.89 |    46.15 |    47.37 |                |
  MysteryDrop.sol       |    61.11 |    47.73 |    70.59 |    61.76 |... 238,242,243 |
  ThirdAlternative.sol  |        0 |        0 |        0 |        0 |... 83,87,88,92 |
 contracts/interfaces/  |      100 |      100 |      100 |      100 |                |
  IAggregator.sol       |      100 |      100 |      100 |      100 |                |
  ICollectionV3.sol     |      100 |      100 |      100 |      100 |                |
 contracts/mocks/       |       75 |      100 |       75 |       60 |                |
  1155Mock.sol          |       50 |      100 |       50 |    33.33 |          19,20 |
  20Mock.sol            |      100 |      100 |      100 |      100 |                |
 contracts/test/        |      100 |      100 |      100 |      100 |                |
  ERC20.t.sol           |      100 |      100 |      100 |      100 |                |
------------------------|----------|----------|----------|----------|----------------|
All files               |    47.59 |    38.89 |       50 |    47.83 |                |
------------------------|----------|----------|----------|----------|----------------|
```

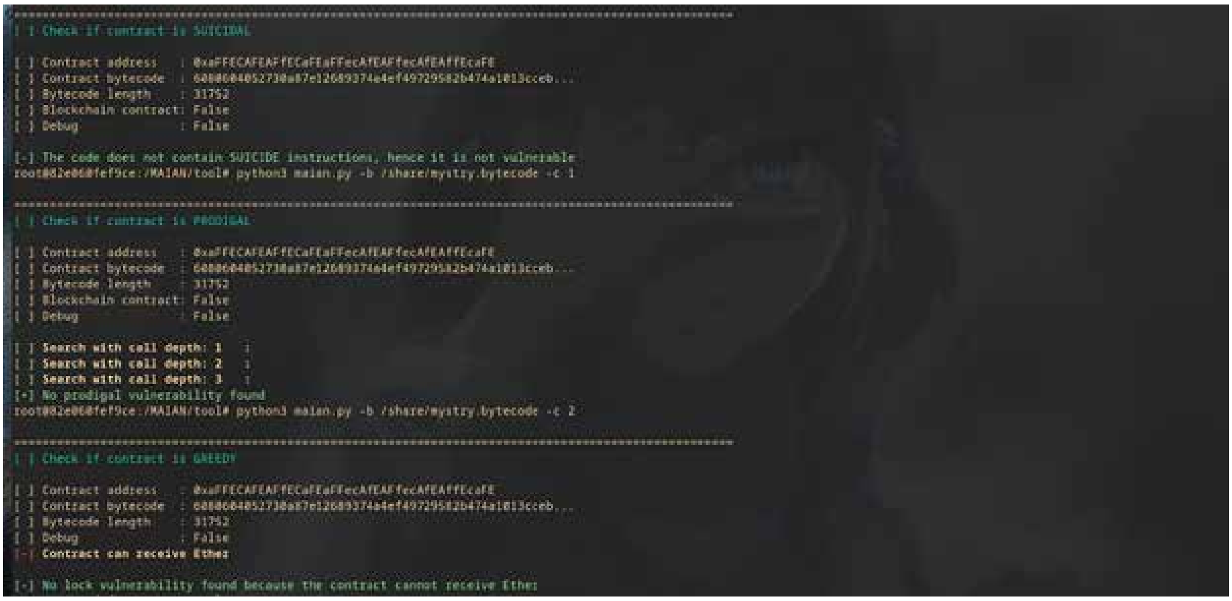
3. **Automated testing**



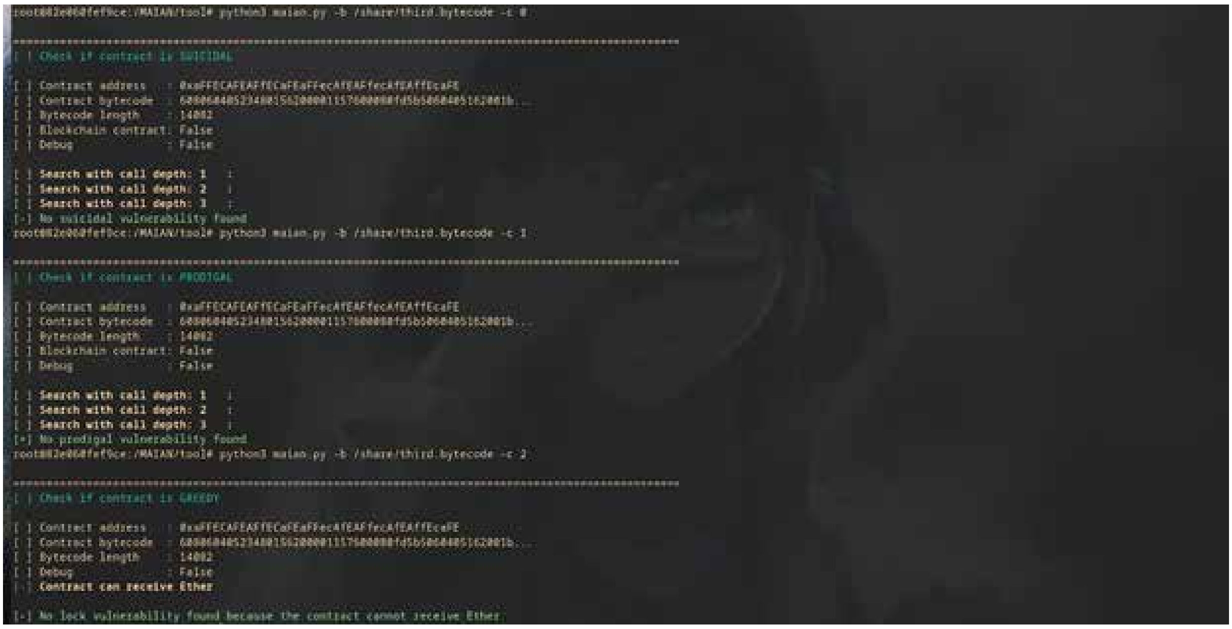
```
Compiled with solc
Number of lines: 2378 (+ 0 in dependencies, + 0 in tests)
Number of assembly lines: 0
Number of contracts: 18 (+ 0 in dependencies, + 1 tests)

Number of optimization issues: 19
Number of informational issues: 105
Number of low issues: 7
Number of medium issues: 19
Number of high issues: 6
ERCs: ERC165, ERC20

+------------------------+-------------+--------+------------+--------------+--------------------+
|          Name          | # functions |  ERCS  | ERC20 info | Complex code |      Features      |
+------------------------+-------------+--------+------------+--------------+--------------------+
|    IERC1155Receiver    |      3      | ERC165 |            |      No      |                    |
|        Address         |      11     |        |            |      No      |      Send ETH      |
|                        |             |        |            |              |    Delegatecall    |
|                        |             |        |            |              |      Assembly      |
|        Counters        |      4      |        |            |      No      |                    |
|       GameItems        |      35     | ERC165 |            |      No      |                    |
|     EnumerableSet      |      24     |        |            |      No      |      Assembly      |
|      ICollectionV3     |      24     |        |            |      No      |                    |
| AggregatorV3Interface  |      1      |        |            |      No      |                    |
|      MysteryDrop       |      16     |        |            |     Yes      | Tokens interaction |
|    ThirdAlternative    |      9      |        |            |      No      | Tokens interaction |
+------------------------+-------------+--------+------------+--------------+--------------------+
```

4. Maian

MysteryDrop bytecode

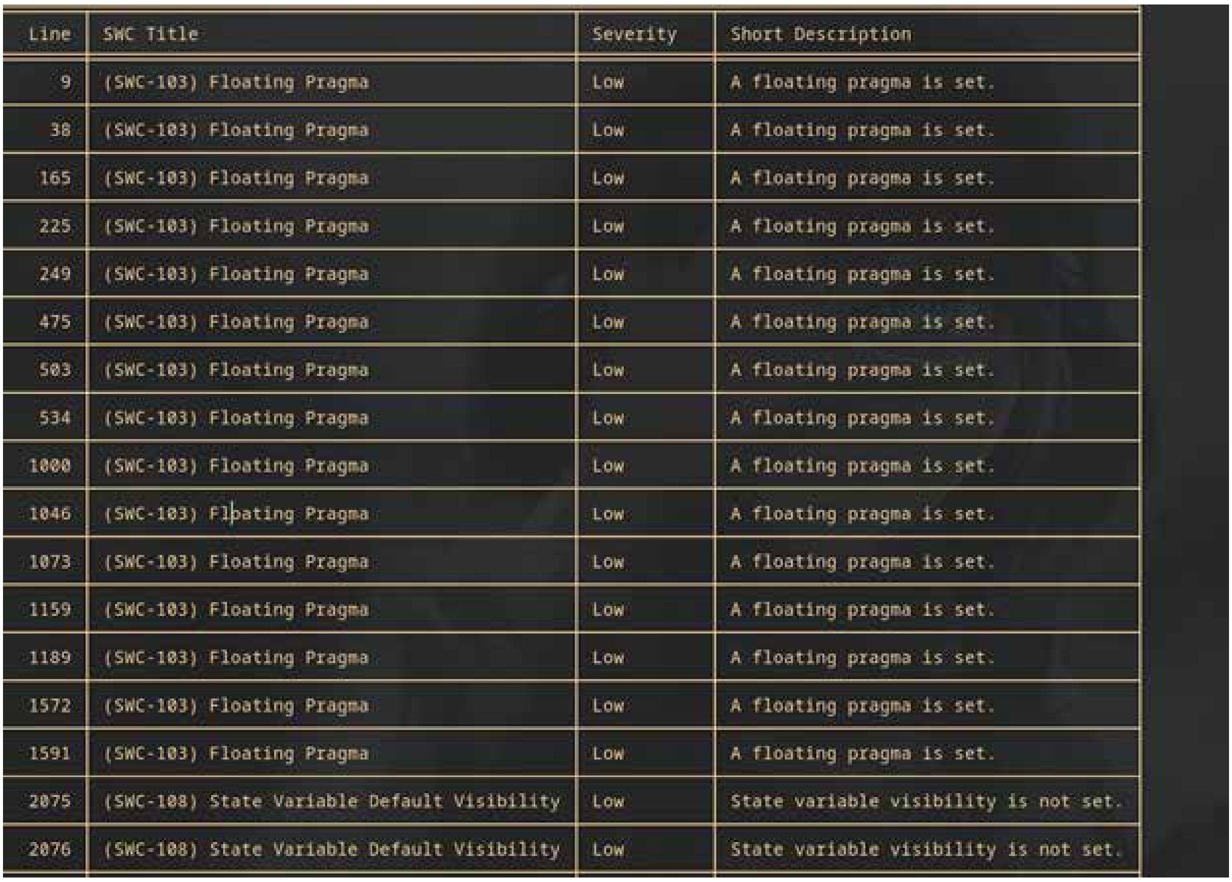


```
=====================================================================================================
[ ] Check if contract is SUICIDAL

[ ] Contract address   : 0xaFFECAFEAFfECaFEaFFecAfEAFfecAfEAffEcaFE
[ ] Contract bytecode  : 6080604052730a87e12689374a4ef49729582b474a1013cceb...
[ ] Bytecode length    : 31752
[ ] Blockchain contract: False
[ ] Debug              : False

[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
root@82e060fef9ce:/MAIAN/tool# python3 maian.py -b /share/mystry.bytecode -c 1

=====================================================================================================
[ ] Check if contract is PRODIGAL

[ ] Contract address   : 0xaFFECAFEAFfECaFEaFFecAfEAFfecAfEAffEcaFE
[ ] Contract bytecode  : 6080604052730a87e12689374a4ef49729582b474a1013cceb...
[ ] Bytecode length    : 31752
[ ] Blockchain contract: False
[ ] Debug              : False

[ ] Search with call depth: 1   :
[ ] Search with call depth: 2   :
[ ] Search with call depth: 3   :
[+] No prodigal vulnerability found
root@82e060fef9ce:/MAIAN/tool# python3 maian.py -b /share/mystry.bytecode -c 2

=====================================================================================================
[ ] Check if contract is GREEDY

[ ] Contract address   : 0xaFFECAFEAFfECaFEaFFecAfEAFfecAfEAffEcaFE
[ ] Contract bytecode  : 6080604052730a87e12689374a4ef49729582b474a1013cceb...
[ ] Bytecode length    : 31752
[ ] Debug              : False
[-] Contract can receive Ether

[-] No lock vulnerability found because the contract cannot receive Ether
```

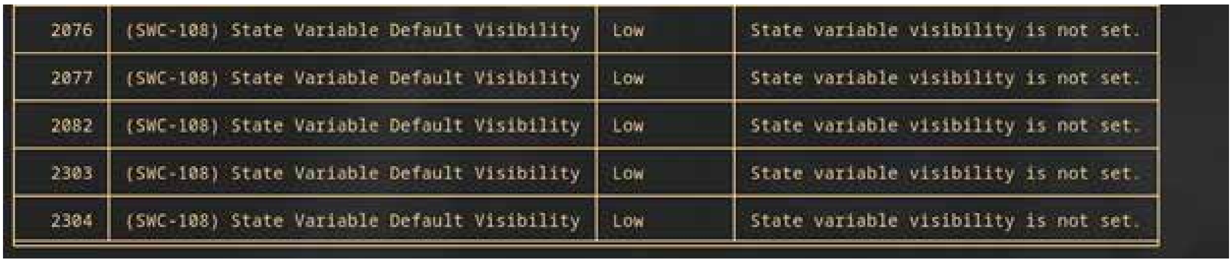


```
root@82e060fef9ce:/MAIAN/tool# python3 maian.py -b /share/third.bytecode -c 0

=====================================================================================================
[ ] Check if contract is SUICIDAL

[ ] Contract address   : 0xaFFECAFEAFfECaFEaFFecAfEAFfecAfEAffEcaFE
[ ] Contract bytecode  : 608060405234801562000011576000080fd5b5060405162001b...
[ ] Bytecode length    : 14082
[ ] Blockchain contract: False
[ ] Debug              : False

[ ] Search with call depth: 1   :
[ ] Search with call depth: 2   :
[ ] Search with call depth: 3   :
[-] No suicidal vulnerability found
root@82e060fef9ce:/MAIAN/tool# python3 maian.py -b /share/third.bytecode -c 1

=====================================================================================================
[ ] Check if contract is PRODIGAL

[ ] Contract address   : 0xaFFECAFEAFfECaFEaFFecAfEAFfecAfEAffEcaFE
[ ] Contract bytecode  : 608060405234801562000011576000080fd5b5060405162001b...
[ ] Bytecode length    : 14082
[ ] Blockchain contract: False
[ ] Debug              : False

[ ] Search with call depth: 1   :
[ ] Search with call depth: 2   :
[ ] Search with call depth: 3   :
[+] No prodigal vulnerability found
root@82e060fef9ce:/MAIAN/tool# python3 maian.py -b /share/third.bytecode -c 2

=====================================================================================================
[ ] Check if contract is GREEDY

[ ] Contract address   : 0xaFFECAFEAFfECaFEaFFecAfEAFfecAfEAffEcaFE
[ ] Contract bytecode  : 608060405234801562000011576000080fd5b5060405162001b...
[ ] Bytecode length    : 14082
[ ] Debug              : False
[-] Contract can receive Ether

[-] No lock vulnerability found because the contract cannot receive Ether
```

5. Mythx

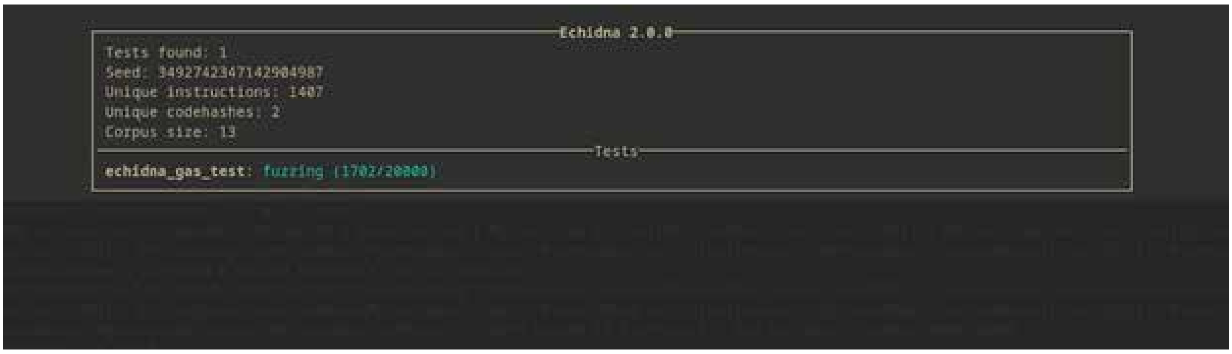


Line	SWC Title	Severity	Short Description
9	(SWC-103) Floating Pragma	Low	A floating pragma is set.
38	(SWC-103) Floating Pragma	Low	A floating pragma is set.
165	(SWC-103) Floating Pragma	Low	A floating pragma is set.
225	(SWC-103) Floating Pragma	Low	A floating pragma is set.
249	(SWC-103) Floating Pragma	Low	A floating pragma is set.
475	(SWC-103) Floating Pragma	Low	A floating pragma is set.
503	(SWC-103) Floating Pragma	Low	A floating pragma is set.
534	(SWC-103) Floating Pragma	Low	A floating pragma is set.
1000	(SWC-103) Floating Pragma	Low	A floating pragma is set.
1046	(SWC-103) Floating Pragma	Low	A floating pragma is set.
1073	(SWC-103) Floating Pragma	Low	A floating pragma is set.
1159	(SWC-103) Floating Pragma	Low	A floating pragma is set.
1189	(SWC-103) Floating Pragma	Low	A floating pragma is set.
1572	(SWC-103) Floating Pragma	Low	A floating pragma is set.
1591	(SWC-103) Floating Pragma	Low	A floating pragma is set.
2075	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
2076	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

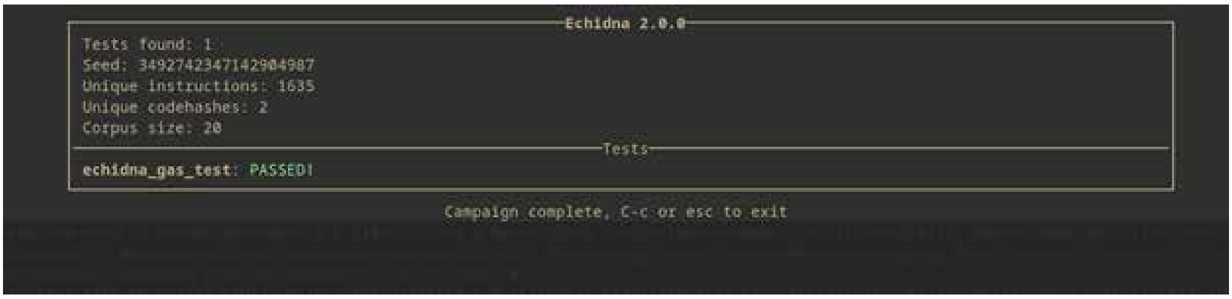


2076	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
2077	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
2082	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
2303	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
2304	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

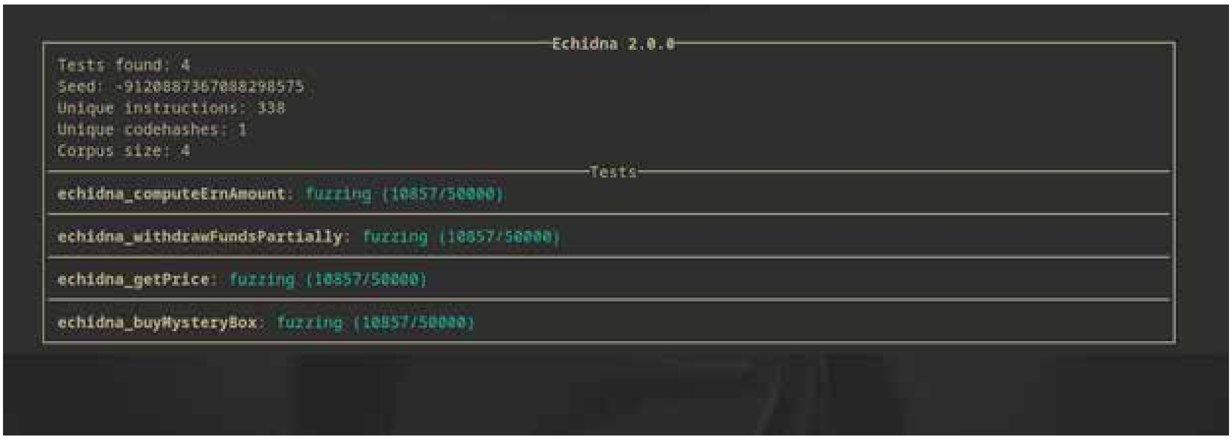
6. Echidna Test:



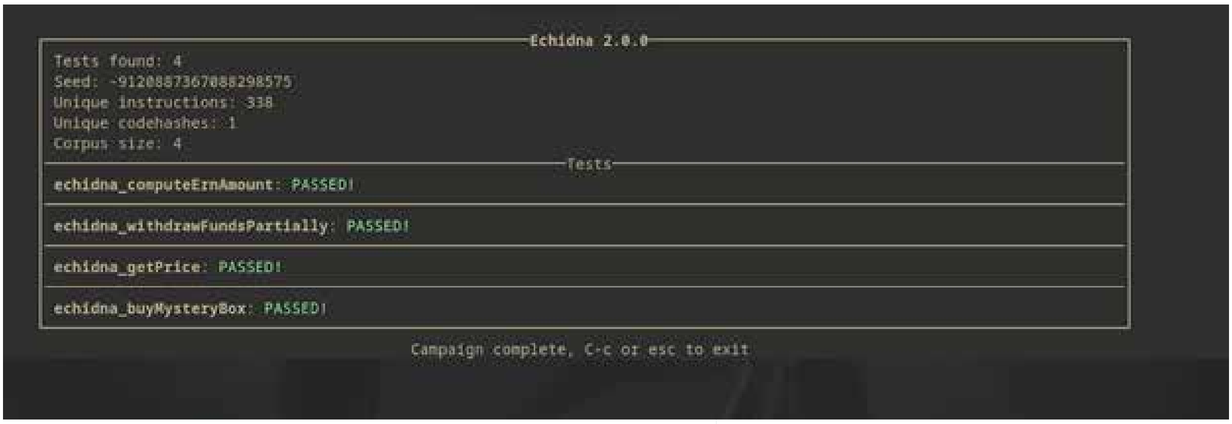
```
Echidna 2.0.0
Tests found: 1
Seed: 3492742347142904987
Unique instructions: 1407
Unique codehashes: 2
Corpus size: 13
Tests
echidna_gas_test: fuzzing (1702/20000)
```



```
Echidna 2.0.0
Tests found: 1
Seed: 3492742347142904987
Unique instructions: 1635
Unique codehashes: 2
Corpus size: 20
Tests
echidna_gas_test: PASSED!

Campaign complete, C-c or esc to exit
```



```
Echidna 2.0.0
Tests found: 4
Seed: -9120887367088298575
Unique instructions: 338
Unique codehashes: 1
Corpus size: 4
Tests
echidna_computeErnAmount: fuzzing (10857/50000)

echidna_withdrawFundsPartially: fuzzing (10857/50000)

echidna_getPrice: fuzzing (10857/50000)

echidna_buyMysteryBox: fuzzing (10857/50000)
```



```
Echidna 2.0.0
Tests found: 4
Seed: -9120887367088298575
Unique instructions: 338
Unique codehashes: 1
Corpus size: 4
Tests
echidna_computeErnAmount: PASSED!

echidna_withdrawFundsPartially: PASSED!

echidna_getPrice: PASSED!

echidna_buyMysteryBox: PASSED!

Campaign complete, C-c or esc to exit
```

Concluding Remarks

While conducting the audits of the Ethernity smart contract, it was observed that the contracts contain High, Medium and Low severity issues.

Our auditors suggest that High, Medium, and Low severity issues should be resolved by the developers. The recommendations given will improve the operations of the smart contract.
Notes:

- The Ethernity team has fixed the issues based on the auditor's recommendation.

Disclaimer

ImmuneBytes's audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

Our team does not endorse the Ethernity platform or its product nor this audit is investment advice.
Notes:

- Please make sure contracts deployed on the mainnet are the ones audited.
- Check for the code refactor by the team on critical issues.

IMMUNEBYTES

Audits