BuyMainStreet Smart Contract Preliminary Audit Report

Project Synopsis

Project Name	**BuyMainStreet**
Platform	Ethereum, Solidity
Github Repo	Not Provided
Deployed Contract	https://bscscan.com/address/0x8fc1a944c149762b6b578a06c0de2abd6b7d2b89#code
Total Duration	4 Days
Timeline of Audit	**24th July 2021 to 30th July 2021**

Contract Details

Total Contract(s)	1
Name of Contract(s)	**BuyMainStreetToken**
Language	Solidity
Commit Hash	**Null**

Contract Vulnerabilities Synopsis

Issues	Open Issues	Closed Issues
Critical Severity	0	0
Medium Severity	2	0
Low Severity	4	0
Information	2	0
Total Found	8	0

Detailed Results

The contract has gone through several stages of the audit procedure that includes structural analysis, automated testing, manual code review etc.

All the issues have been explained and discussed in detail below. Along with the explanation of the issue found during the audit, the recommended way to overcome the issue or improve the code quality has also been mentioned.

A. Contract Name: BuyMainStreetToken

Medium Severity Issues

A.1 *_getTaxFee function* does not hold any significance.

Line no - 830-832

Description:

The protocol includes a function called ***_getTaxFee*** that returns the **_TAX_FEE** state variable in the protocol.

```
829
830  function _getTaxFee() private view returns(uint256) {
831      return _TAX_FEE;
832  }
833
```

However, the function has been marked as private but never called from within the contract at any instance. Moreover, since the visibility specifier is **private** the function cannot be called from outside the contract as well.

Recommendation:

If the ***getTaxFee*** doesn't hold any significance in the protocol, it should be removed to save space and gas.

However, if its a necessary function, the Function should either be marked as External or be called from within the contract at some instance to ensure that its used adequately.

A.2 Loops are extremely costly

Line no -615, 795

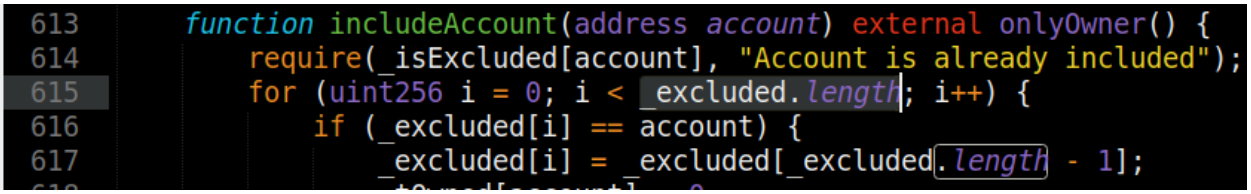
Description:

The **BuyMainStreetToken** contract has some **for loops** in the contract that include state variables like .length of a non-memory array, in the condition of the for loops.

As a result, these state variables consume a lot more extra gas for every iteration of the for loop.

The following function includes such loops at the above-mentioned lines:

- ***includeAccount***
- ***_getCurrentSupply***



Recommendation:

It's quite effective to use a local variable instead of a state variable like .length in a loop. This will be a significant step in optimizing gas usage.

For instance,

```
function includeAccount(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already included");

     uint256 local_variable = _excluded.length; // Storing Length in a local Variable
    for (uint256 i = 0; i < local_variable; i++) {
       if (_excluded[i] == account) {
          _excluded[i] = _excluded[_excluded.length - 1];
          _tOwned[account] = 0;
          _isExcluded[account] = false;
          _excluded.pop();
          break;
       }
    }
 }
```

Low Severity Issues

A.3 Absence of Error messages in Require Statements

Line no - 632

Description:

The **BuyMainStreetToken** contract includes a **require statement** in a functions(*at the above-mentioned lines*) that doesn't contain any error message.

```
631     function updateFee(uint256 _txFee,uint256 _burnFee,uint256 _marketingFee) onlyOwner() public{
632         require( _txFee < 100 && _burnFee < 100 && _marketingFee < 100);
633         _TAX_FEE = _txFee* 100;
634         _BURN_FEE = _burnFee * 100;
```

While this makes it troublesome to detect the reason behind a particular function revert, it also reduces the readability of the code.

Recommendation:

Error Messages must be included in every require statement in the contract

A.4 Absence of Zero Address Validation

Line no-

Description:

During the automated testing, it was found that the contract includes quite a few functions that update an imperative address in the contract like ***FeeAddress***.

However, no Zero Address Validation is implemented on the following function while updating such state variables of the contract:

- ***setAsmarketingAccount***

Recommendation:

A **require** statement should be included in such functions to ensure no zero address is passed in the arguments.

A.5 External Visibility should be preferred

Description:

Those functions that are never called throughout the contract should be marked as ***external*** visibility instead of ***public*** visibility.
This will effectively result in Gas Optimization as well.

Therefore, the following function must be marked as **external** within the contract:

- ***isExcluded()***
- ***totalFees()***
- ***deliver()***

- ***reflectionFromToken()***
- ***totalBurn()***
- ***totalmarketing()***
- ***updateFee()***

Recommendation:
If the PUBLIC visibility of the above-mentioned functions is not intended, then the EXTERNAL Visibility keyword should be preferred.

A.6 Constant declaration should be preferred

Line no- 469, 471

Description:

State variables that are not supposed to change throughout the contract should be declared as **constant**.

Recommendation:

The following state variables need to be declared as **constant**, unless the current contract design is intended.

- **_GRANULARITY**
- **_MAX**

Informational

A.7 Coding Style Issues in the Contract

Explanation:

Code readability of a Smart Contract is largely influenced by the Coding Style issues and in some specific scenarios may lead to bugs in the future.

```
Variable BuyMainStreetToken._NAME (contracts/buyMainStreet.sol#464) is not in mixedCase
Variable BuyMainStreetToken._SYMBOL (contracts/buyMainStreet.sol#465) is not in mixedCase
Variable BuyMainStreetToken._DECIMALS (contracts/buyMainStreet.sol#466) is not in mixedCase
Variable BuyMainStreetToken.FeeAddress (contracts/buyMainStreet.sol#467) is not in mixedCase
Variable BuyMainStreetToken._MAX (contracts/buyMainStreet.sol#469) is not in mixedCase
Variable BuyMainStreetToken._DECIMALFACTOR (contracts/buyMainStreet.sol#470) is not in mixedCase
Variable BuyMainStreetToken._GRANULARITY (contracts/buyMainStreet.sol#471) is not in mixedCase
Variable BuyMainStreetToken._TAX_FEE (contracts/buyMainStreet.sol#480) is not in mixedCase
Variable BuyMainStreetToken._BURN_FEE (contracts/buyMainStreet.sol#481) is not in mixedCase
Variable BuyMainStreetToken._marketing_FEE (contracts/buyMainStreet.sol#482) is not in mixedCase
Variable BuyMainStreetToken.ORIG_TAX_FEE (contracts/buyMainStreet.sol#485) is not in mixedCase
Variable BuyMainStreetToken.ORIG_BURN_FEE (contracts/buyMainStreet.sol#486) is not in mixedCase
Variable BuyMainStreetToken.ORIG_marketing_FEE (contracts/buyMainStreet.sol#487) is not in mixedCase
```

During the automated testing, it was found that the **BuyMainStreetToken** contract had quite a few code style issues.

Recommendation:

Therefore, it is highly recommended to fix the issues like naming convention, indentation, and code layout issues in a smart contract.

A.8 NatSpec Annotations must be included

Description:

The smart contracts do not include the NatSpec annotations adequately.

Recommendation:

Cover by NatSpec all Contract methods.

Automated Test Results

```
Compiled with solc
Number of lines: 835 (+ 0 in dependencies, + 0 in tests)
Number of assembly lines: 0
Number of contracts: 6 (+ 0 in dependencies, + 0 tests)

Number of optimization issues: 23
Number of informational issues: 112
Number of low issues: 5
Number of medium issues: 0
Number of high issues: 0

ERCs: ERC20

+--------------------+-------------+-------+--------------------+--------------+----------+
|        Name        | # functions | ERCS  |     ERC20 info     | Complex code | Features |
+--------------------+-------------+-------+--------------------+--------------+----------+
|      SafeMath      |      8      |       |                    |      No      |          |
|      Address       |      7      |       |                    |      No      | Send ETH |
|                    |             |       |                    |              | Assembly |
| BuyMainStreetToken |      57     | ERC20 |     No Minting     |     Yes      |          |
|                    |             |       | Approve Race Cond. |              |          |
|                    |             |       |                    |              |          |
+--------------------+-------------+-------+--------------------+--------------+----------+
INFO:Slither:contracts/buyMainStreet.sol analyzed (6 contracts)
```