

Security Audit

Name: HolidayCoin

Date: 04/26/2021

Contract: <https://bscscan.com/address/0xF081C3fa389f5333214e46E38dc0060cD3619703>

<https://lemonsec.com>

Lemon
Sec

Issues Checking Status

Compiler Errors: **PASSED**

Race conditions: **PASSED**

Possible delays in data delivery: **PASSED**

Oracle calls: **PASSED**

Front running: **PASSED**

Timestamp dependence: **PASSED**

Integer Overflow and Underflow: **PASSED**

DoS with Revert: **PASSED**

Economy model of the contract: **PASSED**

Methods execution permissions: **PASSED**

DoS with block gas limit: **2 LOW ISSUES**

The impact of the exchange rate on the logic: **PASSED**

Private user data leaks: **PASSED**

Malicious Event log: **PASSED**

Issues Checking Status

Scoping and Declarations: **PASSED**

Uninitialized storage pointers: **PASSED**

Arithmetic accuracy: **PASSED**

Design Logic: **PASSED**

Cross-function race conditions: **PASSED**

Fallback function security: **PASSED**

Security Issues

Critical Severity Issues: 0

High Severity Issues: 0

Medium Severity Issues: 0

Low Severity Issues: 2

The function **includeInReward()** uses the loop to find and remove addresses from the **_isExcluded** list. Function will be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

Security Issues

The function `_getCurrentSupply` also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

```
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

Recommendation:

Use EnumerableSet instead of array or do not use long arrays.

Token Holders

A screenshot of a web-based token holder dashboard. The top navigation bar includes links for Transfers, Holders (which is currently selected), Info, Read Contract, Write Contract, Analytics, and Comments. A search icon is located in the top right corner. Below the navigation, a section titled "Token Holders Chart" displays a single entry: "A total of 1 token holder". The main table lists the following data:

Rank	Address	Quantity	Percentage	Analytics
1	0x9153e21954dc0446427422e2116df321ba8e60f2	1,000,000,000	100.0000%	🔗

At the bottom right of the table, there is a link "[Download CSV Export]". Navigation controls at the bottom include "First", "Previous", "Page 1 of 1", "Next", and "Last".

Conclusion

APPROVED

No vulnerabilities were found.

No Backdoors or Scam Scripts found.

The Smart Contract is Safe to use in the Binance Smart Chain.

REMEMBER: This REPORT is APPROVED only on BSC Contract listed on First Page.

If you are interested in Auditing your project contact us on:



<https://t.me/lemonsec>



<https://lemonsec.com>



https://twitter.com/lemon_sec