Smart Contract Security Audit Report

The SlowMist Security Team received the TROY team's application for smart contract security audit of the TROY Token on December 10, 2019. The following are the details and results of this smart contract security audit:

Token name :

TROY

The Contract address :

0x4574562e9310a94f9ca962bd23168d8a06875b1a

Link address :

https://etherscan.io/address/0x4574562e9310a94f9ca962bd23168d8a06875b1a

The audit items and results :

(Other unknown security vulnerabilities are not included in the audit responsibility scope)

No.	Audit Items	Audit Subclass	Audit Subclass Result
1	Overflow Audit	-	Passed
2	Race Conditions Audit	-	Passed
3	Authority Control Audit	Permission vulnerability audit	Passed
		Excessive auditing authority	Passed
4	Safety Design Audit	Zeppelin module safe use	Passed
		Compiler version security	Passed
		Hard-coded address security	Passed
		Fallback function safe use	Passed
		Show coding security	Passed
		Function return value security	Passed
		Call function security	Passed
5	Denial of Service Audit	-	Passed
6	Gas Optimization Audit	-	Passed
7	Design Logic Audit	-	Passed
8	“False Deposit” vulnerability Audit	-	Passed

9	Malicious Event Log Audit	-	Passed
10	Scoping and Declarations Audit	-	Passed
11	Replay Attack Audit	ECDSA's Signature Replay Audit	Passed
12	Uninitialized Storage Pointers Audit	-	Passed
13	Arithmetic Accuracy Deviation Audit	-	Passed

Audit Result : **Passed**

Audit Number : 0X001912180001

Audit Date : December 18, 2019

Audit Team : SlowMist Security Team

(**Statement :** SlowMist only issues this report based on the fact that has occurred or existed before the report is issued, and bears the corresponding responsibility in this regard. For the facts occur or exist later after the report, SlowMist cannot judge the security status of its smart contract. SlowMist is not responsible for it. The security audit analysis and other contents of this report are based on the documents and materials provided by the information provider to SlowMist as of the date of this report (referred to as "the provided information"). SlowMist assumes that: there has been no information missing, tampered, deleted, or concealed. If the information provided has been missed, modified, deleted, concealed or reflected and is inconsistent with the actual situation, SlowMist will not bear any responsibility for the resulting loss and adverse effects. SlowMist will not bear any responsibility for the background or other circumstances of the project.)

Summary: This is a token contract that does not contain the tokenVault section. OpenZeppelin's SafeMath security module is used, which is a commendable approach. The contract does not have the Overflow and the Race Conditions issue. The comprehensive evaluation contract is no risk.

The source code:

```
/**
 *Submitted for verification at Etherscan.io on 2019-12-16
*/

/**
 *Submitted for verification at Etherscan.io on 2019-11-28
*/

//SlowMist// The contract does not have the Overflow and the Race Conditions issue

pragma solidity ^0.5.0;
```

//SlowMist// OpenZeppelin's SafeMath security module is used, which is a commendable

```
library SafeMath {
    function add(uint256 a, uint256 b) internal pure returns (uint256 c) {
        c = a + b;
        require(c >= a);
    }
    function sub(uint256 a, uint256 b) internal pure returns (uint256 c) {
        require(b <= a);
        c = a - b;
    }
    function mul(uint256 a, uint256 b) internal pure returns (uint256 c) {
        c = a * b;
        require(a == 0 || c / a == b);
    }
    function div(uint256 a, uint256 b) internal pure returns (uint256 c) {
        require(b > 0);
        c = a / b;
    }
}

contract ERC20Interface {
    function totalSupply() public view returns (uint256);
    function balanceOf(address tokenOwner) public view returns (uint256 balance);
    function allowance(address tokenOwner, address spender) public view returns (uint256 remaining);
    function transfer(address to, uint256 value) public returns (bool success);
    function approve(address spender, uint256 value) public returns (bool success);
    function transferFrom(address from, address to, uint256 value) public returns (bool success);

    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(address indexed tokenOwner, address indexed spender, uint256 value);
}

contract TROY is ERC20Interface {
    using SafeMath for uint256;
    string public symbol;
    string public name;
    uint8 public decimals;
    uint256 _totalSupply;
    address public owner;
    bool public activeStatus = true;
```

```
event Active(address msgSender);
event Reset(address msgSender);
event Freeze(address indexed from, uint256 value);
event Unfreeze(address indexed from, uint256 value);

mapping(address => uint256) public balances;
mapping(address => uint256) public freezeOf;
mapping(address => mapping(address => uint256)) public allowed;

constructor() public {
    symbol = "TROY";
    name = "TROY";
    decimals = 18;
    _totalSupply = 10000000000 * 10**uint(decimals);
    owner = msg.sender;
    balances[owner] = _totalSupply;
    emit Transfer(address(0), owner, _totalSupply);
}

function isOwner(address add) public view returns (bool) {
  if (add == owner) {
    return true;
  } else return false;
}

modifier onlyOwner {
if (!isOwner(msg.sender)) {
            revert();
      }
_;
}

modifier onlyActive {
 if (!activeStatus) {
            revert();
    }
_;
}

function activeMode() public onlyOwner {
    activeStatus = true;
```

```
        emit Active(msg.sender);
    }

    //SlowMist// Suspending all transactions upon major abnormalities is a recommended approach
    function resetMode() public onlyOwner {
        activeStatus = false;
        emit Reset(msg.sender);
    }

    function totalSupply() public view returns (uint256) {
        return _totalSupply;
    }

    function balanceOf(address tokenOwner) public view returns (uint256 balance) {
        return balances[tokenOwner];
    }

    function allowance(address tokenOwner, address spender) public view returns (uint256 remaining) {
        return allowed[tokenOwner][spender];
    }

    function transfer(address to, uint256 value) public onlyActive returns (bool success) {
        if (to == address(0)) { //SlowMist// This kind of check is very good, avoiding user mistake leading to the loss of token during transfer
            revert();
        }
        if (value <= 0) {
            revert();
        }
        if (balances[msg.sender] < value) {
            revert();
        }
        balances[msg.sender] = balances[msg.sender].sub(value);
        balances[to] = balances[to].add(value);
        emit Transfer(msg.sender, to, value);
        return true; //SlowMist// The return value conforms to the EIP20 specification
```

```
    }

    function approve(address spender, uint256 value) public onlyActive returns (bool success) {
        if (value <= 0) {
            revert();
        }
        allowed[msg.sender][spender] = value;
        emit Approval(msg.sender, spender, value);

        return true; //SlowMist// The return value conforms to the EIP20 specification

    }

    function transferFrom(address from, address to, uint256 value) public onlyActive returns (bool success) {

        if (to == address(0)) { //SlowMist// This kind of check is very good, avoiding user mistake
leading to the loss of token during transfer
            revert();
        }
        if (value <= 0) {
            revert();
        }
        if (balances[from] < value) {
            revert();
        }
        if (value > allowed[from][msg.sender]) {
            revert();
        }
        balances[from] = balances[from].sub(value);
        allowed[from][msg.sender] = allowed[from][msg.sender].sub(value);
        balances[to] = balances[to].add(value);
        emit Transfer(from, to, value);

        return true; //SlowMist// The return value conforms to the EIP20 specification

    }


        function freeze(uint256 value) public onlyActive returns (bool success) {
        if (balances[msg.sender] < value) {
            revert();
        }
                        if (value <= 0){
                            revert();
```

```
                        }
        balances[msg.sender] = balances[msg.sender].sub(value);
        freezeOf[msg.sender] = freezeOf[msg.sender].add(value);
        emit Freeze(msg.sender, value);
        return true;
    }

        function unfreeze(uint256 value) public onlyActive returns (bool success) {
        if (freezeOf[msg.sender] < value) {
            revert();
        }
                        if (value <= 0) {
                            revert();
                        }
        freezeOf[msg.sender] = freezeOf[msg.sender].sub(value);
                        balances[msg.sender] = balances[msg.sender].add(value);
        emit Unfreeze(msg.sender, value);
        return true;
    }

    function () external payable {
        revert();
    }
}
```

慢雾科技
slow mist

Official Website
www.slowmist.com

E-mail
team@slowmist.com

Twitter
@SlowMist_Team

WeChat Official Account