慢雾科技
slow mist

智能合约安全审计报告

慢雾安全团队于 2018-05-04 日，收到 Mixin 团队对 Mixin 项目智能合约安全审计申请。如下为本次智能合约安全审计细节及结果：

Token 名称：

Mixin

合约地址：

0xa974c709cfb4566686553a20790685a47aceaa33

链接地址：

https://etherscan.io/address/0xa974c709cfb4566686553a20790685a47aceaa33#code

本次审计项及结果：

（其他未知安全漏洞不包含在本次审计责任范围）

序号	审计子类	审计子类结果
1	溢出审计	通过
2	条件竞争审计	通过
3	权限控制审计	通过
4	安全设计审计	通过
5	拒绝服务审计	通过
6	Gas 优化审计	优秀
7	设计逻辑审计	通过

备注：审计意见及建议见代码注释 **//SlowMist//......**

审计结果：**优秀**

审计编号：0X001805040001

审计日期：2018 年 5 月 4 日

审计团队：慢雾安全团队

智能合约源代码如下：

```
pragma solidity ^0.4.18;

/**
 * @title SafeMath
 * @dev Math operations with safety checks that throw on error
 */
//SlowMist// 参考了 openzeppelin-solidity 安全函数，非常好
library SafeMath {
  function mul(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a * b;
    assert(a == 0 || c / a == b);
    return c;
  }

  function div(uint256 a, uint256 b) internal pure returns (uint256) {
    // assert(b > 0); // Solidity automatically throws when dividing by 0
    uint256 c = a / b;
    // assert(a == b * c + a % b); // There is no case in which this doesn't hold
    return c;
  }

  function sub(uint256 a, uint256 b) internal pure returns (uint256) {
    assert(b <= a);
    return a - b;
  }

  function add(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a + b;
    assert(c >= a);
    return c;
  }
}

/**
 * @title ERC20 interface
 * @dev see https://github.com/ethereum/EIPs/issues/20
 */
contract ERC20 {
  uint256 public totalSupply;
  function balanceOf(address who) public view returns (uint256);
  function transfer(address to, uint256 value) public returns (bool);
  function allowance(address owner, address spender) public view returns (uint256);
  function transferFrom(address from, address to, uint256 value) public returns (bool);
  function approve(address spender, uint256 value) public returns (bool);
  event Transfer(address indexed from, address indexed to, uint256 value);
  event Approval(address indexed owner, address indexed spender, uint256 value);
}

/**
 * @title Standard ERC20 token
 *
 * @dev Implementation of the basic standard token.
 * @dev https://github.com/ethereum/EIPs/issues/20
 * @dev Based on code by FirstBlood:
https://github.com/Firstbloodio/token/blob/master/smart_contract/FirstBloodToken.sol
 */
contract StandardToken is ERC20 {
  using SafeMath for uint256;

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

  /**
   * @dev Gets the balance of the specified address.
   * @param _owner The address to query the the balance of.
   * @return An uint256 representing the amount owned by the passed address.
   */
  function balanceOf(address _owner) public view returns (uint256 balance) {
    return balances[_owner];
  }

  /**
   * @dev transfer token for a specified address
   * @param _to The address to transfer to.
   * @param _value The amount to be transferred.
   */
  function transfer(address _to, uint256 _value) public returns (bool) {
    require(_to != address(0)); //SlowMist// 这个检查很好，避免用户失误导致转丢

    // SafeMath.sub will throw if there is not enough balance.
    balances[msg.sender] = balances[msg.sender].sub(_value);
    balances[_to] = balances[_to].add(_value);
    Transfer(msg.sender, _to, _value);
    return true;
  }

  /**
   * @dev Transfer tokens from one address to another
   * @param _from address The address which you want to send tokens from
   * @param _to address The address which you want to transfer to
   * @param _value uint256 the amount of tokens to be transferred
   */
  function transferFrom(address _from, address _to, uint256 _value) public returns (bool) {
    var _allowance = allowed[_from][msg.sender];
    require(_to != address(0));
    require (_value <= _allowance);  //SlowMist// 没有对 balances[_from]进行无效判断节约 Gas
    balances[_from] = balances[_from].sub(_value);
    balances[_to] = balances[_to].add(_value);
    allowed[_from][msg.sender] = _allowance.sub(_value);
    Transfer(_from, _to, _value);
    return true;
  }

  /**
   * @dev Approve the passed address to spend the specified amount of tokens on behalf of msg.sender.
   * @param _spender The address which will spend the funds.
   * @param _value The amount of tokens to be spent.
   */
  function approve(address _spender, uint256 _value) public returns (bool) {
    // To change the approve amount you first have to reduce the addresses`
    //  allowance to zero by calling `approve(_spender, 0)` if it is not
    //  already 0 to mitigate the race condition described here:
    //  https://github.com/ethereum/EIPs/issues/20#issuecomment-263524729
    require((_value == 0) || (allowed[msg.sender][_spender] == 0));  //SlowMist// 实际无意义建议去掉节
约 Gas
    allowed[msg.sender][_spender] = _value;
    Approval(msg.sender, _spender, _value);
    return true;
  }

  /**
   * @dev Function to check the amount of tokens that an owner allowed to a spender.
   * @param _owner address The address which owns the funds.
   * @param _spender address The address which will spend the funds.
   * @return A uint256 specifying the amount of tokens still available for the spender.
   */
  function allowance(address _owner, address _spender) public view returns (uint256 remaining) {
    return allowed[_owner][_spender];
  }
}

contract MixinToken is StandardToken {
  string public constant name = "Mixin";
  string public constant symbol = "XIN";
  uint8 public constant decimals = 18;

  function MixinToken() public {
    totalSupply = 1000000000000000000000000; //SlowMist// 可读性差，建议科学计数法表示
    balances[msg.sender] = totalSupply;
  }
}
```

慢雾科技
slow mist

官方网址

www.slowmist.com

电子邮箱

team@slowmist.com

微信公众号