智能合约安全审计报告

慢雾安全团队于 2018-11-27 日，收到 OKNC 团队对 Ok Node Community Token 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：

OKNC

合约文件名和 SHA256 值：

oknc_v4.sol (SHA256):

c9888d0f19b5f344c8bb5cde4ced5a1391cc8697c24fa58a0cffec74f740d300

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	审计大类	审计子类	审计结果
1	溢出审计	-	通过
2	条件竞争审计	-	通过
3	权限控制审计	权限漏洞审计	通过
		权限过大审计	通过
4	安全设计审计	Zeppelin 模块使用安全	通过
		编译器版本安全	通过
		硬编码地址安全	通过
		Fallback 函数使用安全	通过
		显现编码安全	通过
		函数返回值安全	通过
		call 调用安全	通过
5	拒绝服务审计	-	通过
6	Gas 优化审计	-	通过
7	设计逻辑审计	-	通过
8	“假充值”漏洞审计	-	通过
9	恶意 Event 事件日志审计	-	通过

10	未初始化的存储指针	-	通过
11	算术精度误差	-	通过

备注：审计意见及建议见代码注释 **//SlowMist//.....**

审计结果：**通过**

审计编号：0X001811290001

审计日期：2018 年 11 月 29 日

审计团队：慢雾安全团队

总结：此为代币(token)合约，不包含锁仓(tokenVault)部分。合约合理地使用了 OpenZeppelin 的 SafeMath 安全模块，不存在溢出、条件竞争问题，综合评估合约无风险。

合约源代码如下：

//SlowMist// 合约不存在溢出、条件竞争问题

//SlowMist// 使用了 OpenZeppelin 的 SafeMath 安全模块，值得称赞的做法

```
pragma solidity ^0.4.25;

/**
 * @title SafeMath
 * @dev Math operations with safety checks that throw on error
 */
library SafeMath {
  function mul(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a * b;
    require(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) {
    require(b <= a);
    return a - b;
  }

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


// Abstract contract for the full ERC 20 Token standard
// https://github.com/ethereum/EIPs/issues/20

contract Token {
    /* This is a slight change to the ERC20 base standard.
    function totalSupply() constant returns (uint256 supply);
    is replaced with:
    uint256 public totalSupply;
    This automatically creates a getter function for the totalSupply.
    This is moved to the base contract since public getter functions are not
    currently recognised as an implementation of the matching abstract
    function by the compiler.
    */
    /// total amount of tokens
    //uint256 public totalSupply;
    function totalSupply()  public view returns (uint256 supply);

    /// @param _owner The address from which the balance will be retrieved
    /// @return The balance
    function balanceOf(address _owner)  public view returns (uint256 balance);

    /// @notice send `_value` token to `_to` from `msg.sender`
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
```

```
    /// @return Whether the transfer was successful or not
    function transfer(address _to, uint256 _value)  public returns (bool success);

    /// @notice send `_value` token to `_to` from `_from` on the condition it is approved by `_from`
    /// @param _from The address of the sender
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transferFrom(address _from, address _to, uint256 _value) public  returns (bool success);

    /// @notice `msg.sender` approves `_addr` to spend `_value` tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @param _value The amount of wei to be approved for transfer
    /// @return Whether the approval was successful or not
    function approve(address _spender, uint256 _value)  public returns (bool success);

    /// @param _owner The address of the account owning tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @return Amount of remaining tokens allowed to spent
    function allowance(address _owner, address _spender)  public view returns (uint256 remaining);

    event Transfer(address indexed _from, address indexed _to, uint256 _value);
    event Approval(address indexed _owner, address indexed _spender, uint256 _value);
}

/// OKNC token, ERC20 compliant
contract OKNC is Token{
    using SafeMath for uint256;

    string public name = "Ok Node Community Token"; // Set the name for display purposes
    string public symbol = "OKNC";// Set the symbol for display purposes
    uint8 public decimals = 4;
    uint256 public totalSupply;
          address public owner;

    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;
    mapping (address => mapping (address => uint256)) public allowance;


    /* Initializes contract with initial supply tokens to the creator of the contract */
    constructor() public {
```

```
        totalSupply = 21000000000 * 10 ** uint256(decimals);                    // Update total supply
        balanceOf[msg.sender] = totalSupply;             // Give the creator all initial tokens

        owner = msg.sender;
    }

    function totalSupply() public view returns (uint256 supply){
        return totalSupply;
    }
    function balanceOf(address _owner) public view returns (uint256 balance) {
        return balanceOf[_owner];
    }

    /* Send coins */
    function transfer(address _to, uint256 _value) public returns (bool){
        require(_to != address(0)); //SlowMist// 这类检查很好，避免用户失误导致 Token 转丢
// Prevent transfer to 0x0 address. Use burn() instead

        require(_value <= balanceOf[msg.sender]);           // Check if the sender has enough
        require(balanceOf[_to] + _value >= balanceOf[_to]); // Check for overflows
        balanceOf[msg.sender] = balanceOf[msg.sender].sub(_value);                     // Subtract from
the sender
        balanceOf[_to] = balanceOf[_to].add(_value);                            // Add the same to the
recipient
        emit Transfer(msg.sender, _to, _value);                   // Notify anyone listening that this
transfer took place


        return true; //SlowMist// 返回值符合 EIP20 规范
    }

    /* Allow another contract to spend some tokens in your behalf */
    function approve(address _spender, uint256 _value)public
        returns (bool success) {

        allowance[msg.sender][_spender] = _value;
        emit Approval(msg.sender, _spender, _value);

        return true; //SlowMist// 返回值符合 EIP20 规范
    }
```

```
    /* A contract attempts to get the coins */
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
        require(_to != address(0)); //SlowMist// 这类检查很好，避免用户失误导致 Token 转丢
// Prevent transfer to 0x0 address. Use burn() instead

        require(_value <= balanceOf[_from]);               // Check if the sender has enough
        require(balanceOf[_to] + _value >= balanceOf[_to]); // Check for overflows
        require(_value <= allowance[_from][msg.sender]);     // Check allowance
        balanceOf[_from] = balanceOf[_from].sub(_value);                           // Subtract from the
sender
        balanceOf[_to] = balanceOf[_to].add(_value);                             // Add the same to the
recipient
        allowance[_from][msg.sender] = allowance[_from][msg.sender].sub(_value);
        emit Transfer(_from, _to, _value);
        return true; //SlowMist// 返回值符合 EIP20 规范
    }

    function allowance(address _owner, address _spender)  public view returns (uint256 remaining) {
        return allowance[_owner][_spender];
    }


}
```

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