智能合约安全审计报告

慢雾安全团队于 2019-06-30 日，收到 BCAT 团队对 BCAT 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：

BCAT

合约地址：

0xfdeaa4ab9fea519afd74df2257a21e5bca0dfd3f

链接地址：

https://etherscan.io/address/0xfdeaa4ab9fea519afd74df2257a21e5bca0dfd3f

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：0X001906300002

审计日期：2019 年 06 月 30 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。合约没有使用了 OpenZeppelin 的 SafeMath 安全模块，合约在涉及算术运算的操作均进行了检查，合约不存在溢出、条件竞争问题，合约存在燃烧功能，允许用户燃烧自己的代币，综合评估合约无风险。

合约源代码如下：

```
/**
 *Submitted for verification at Etherscan.io on 2019-04-24
*/
```

//SlowMist// 合约不存在溢出、条件竞争问题

```
pragma solidity ^0.4.19;

contract BaseToken {
    string public name;
    string public symbol;
    uint8 public decimals;
    uint256 public totalSupply;

    mapping (address => uint256) public balanceOf;
    mapping (address => mapping (address => uint256)) public allowance;
```

```
    event Transfer(address indexed from, address indexed to, uint256 value);
    event Approval(address indexed owner, address indexed spender, uint256 value);

    function _transfer(address _from, address _to, uint _value) internal {
        require(_to != 0x0); //SlowMist// 这类检查很好，避免用户失误导致 Token 转丢
        require(balanceOf[_from] >= _value);
        require(balanceOf[_to] + _value > balanceOf[_to]);
        uint previousBalances = balanceOf[_from] + balanceOf[_to];
        balanceOf[_from] -= _value;
        balanceOf[_to] += _value;
        assert(balanceOf[_from] + balanceOf[_to] == previousBalances);
        Transfer(_from, _to, _value);
    }

    function transfer(address _to, uint256 _value) public returns (bool success) {
        _transfer(msg.sender, _to, _value);
        return true; //SlowMist// 返回值符合 EIP20 规范
    }

    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
        require(_value <= allowance[_from][msg.sender]);
        allowance[_from][msg.sender] -= _value;
        _transfer(_from, _to, _value);
        return true; //SlowMist// 返回值符合 EIP20 规范
    }

    function approve(address _spender, uint256 _value) public returns (bool success) {
        allowance[msg.sender][_spender] = _value;
        Approval(msg.sender, _spender, _value);
        return true; //SlowMist// 返回值符合 EIP20 规范
    }
}

contract BurnToken is BaseToken {
    event Burn(address indexed from, uint256 value);

    function burn(uint256 _value) public returns (bool success) {
```

```
        require(balanceOf[msg.sender] >= _value);
        balanceOf[msg.sender] -= _value;
        totalSupply -= _value;
        Burn(msg.sender, _value);
        return true;
    }

    function burnFrom(address _from, uint256 _value) public returns (bool success) {
        require(balanceOf[_from] >= _value);
        require(_value <= allowance[_from][msg.sender]);
        balanceOf[_from] -= _value;
        allowance[_from][msg.sender] -= _value;
        totalSupply -= _value;
        Burn(_from, _value);
        return true;
    }
}

contract CustomToken is BaseToken, BurnToken {
    function CustomToken() public {
        totalSupply = 100000000000000000000000000;
        name = 'BCAT';
        symbol = 'BCAT';
        decimals = 18;
        balanceOf[0x1f759fae44ca006a496434908b009820afea0a90] = totalSupply;
        Transfer(address(0), 0x1f759fae44ca006a496434908b009820afea0a90, 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