Beauty Chain(美链)

背景介绍

- 美链(Beauty Chain)是一个部署在以太坊上的智能合约，有自己的代币BEC。
  - 没有自己的区块链，代币的发行、转账都是通过调用智能合约中的函数来完成的
  - 可以自己定义发行规则，每个账户有多少代币也是保存在智能合约的状态变量里
  - ERC 20是以太坊上发行代币的一个标准，规范了所有发行代币的合约应该实现的功能和遵循的接口
  - 美链中有一个叫batchTransfer的函数，它的功能是向多个接收者发送代币，然后把这些代币从调用者的帐户上扣除

```
  function batchTransfer(address[] _receivers, uint256 _value) public whenNotPaused returns (bool) {
    uint cnt = _receivers.length;
    uint256 amount = uint256(cnt) * _value;
    require(cnt > 0 && cnt <= 20);
    require(_value > 0 && balances[msg.sender] >= amount);

    balances[msg.sender] = balances[msg.sender].sub(amount);
    for (uint i = 0; i < cnt; i++) {
        balances[_receivers[i]] = balances[_receivers[i]].add(_value);
        Transfer(msg.sender, _receivers[i], _value);
    }
    return true;
  }
}
```

来源：https://etherscan.io/address/0xc5d105e63711398af9bbff092d4b6769c82f793d#code

```
  function batchTransfer(address[] _receivers, uint256 _value) public whenNotPaused returns (bool) {
    uint cnt = _receivers.length;
    uint256 amount = uint256(cnt) * _value;
    require(cnt > 0 && cnt <= 20);
    require(_value > 0 && balances[msg.sender] >= amount);

    balances[msg.sender] = balances[msg.sender].sub(amount);
    for (uint i = 0; i < cnt; i++) {
        balances[_receivers[i]] = balances[_receivers[i]].add(_value);
        Transfer(msg.sender, _receivers[i], _value);
    }
    return true;
  }
}
```

来源：https://etherscan.io/address/0xc5d105e63711398af9bbff092d4b6769c82f793d#code

攻击细节

```
Function: batchTransfer(address[] _receivers, uint256 _value)

MethodID: 0x83f12fec
[0]:  0000000000000000000000000000000000000000000000000000000000000040
[1]:  8000000000000000000000000000000000000000000000000000000000000000
[2]:  0000000000000000000000000000000000000000000000000000000000000002
[3]:  000000000000000000000000b4d30cac5124b46c2df0cf3e3e1be05f42119033
[4]:  000000000000000000000000e823ffe018727585eaf5bc769fa80472f76c3d7
```

- 第0号参数是_receivers数组在参数列表中的位置，即从第64个byte开始，也就是第2号参数
  - 第2号参数先指明数组长度为2，然后第3号参数和第4号参数表明两个接受者的地址
- 第1号参数是给每个接受者转账的金额
- 通过这样的参数计算出来的amount恰好溢出为0！

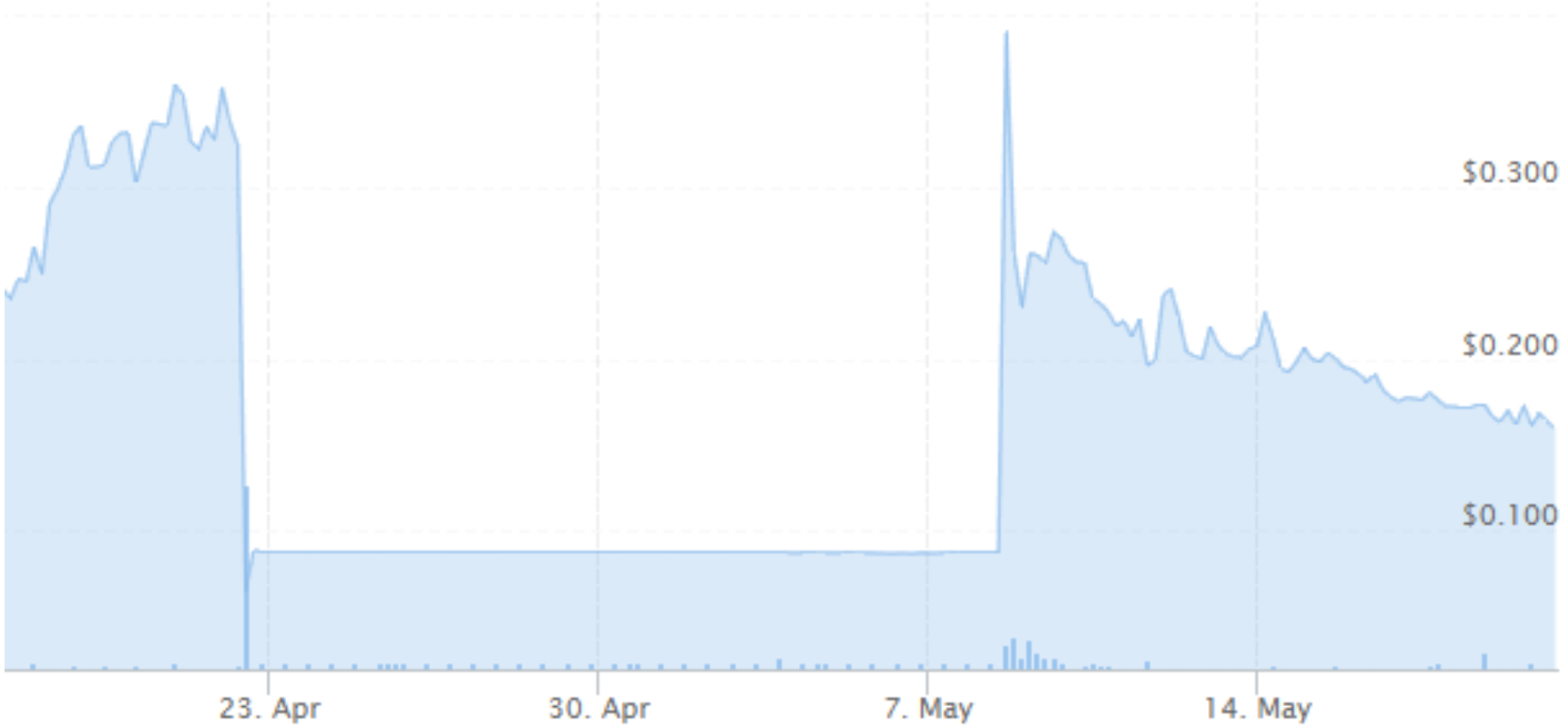
来源：https://etherscan.io/tx/0xad89ff16fd1ebe3a0a7cf4ed282302c06626c1af33221ebe0d3a470aba4a660f

攻击细节

TxHash:	0xad89ff16fd1ebe3a0a7cf4ed282302c06626c1af33221ebe0d3a470aba4a660f
TxReceipt Status:	Success
Block Height:	5483643 (344833 block confirmations)
TimeStamp:	60 days 9 hrs ago (Apr-22-2018 03:28:52 AM +UTC)
From:	0x09a34e01fbaa49f27b0b129d3c5e6e21ed5fe93c
To:	Contract 0xc5d105e63711398af9bbff092d4b6769c82f793d (BeautyChainToken)
Token Transfer:	▸ 57,896,044,618,658,100,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000.792003956564819968 ERC20 (BeautyChain Token) from 0x09a34e01fbaa49f... to ➞ 0xb4d30cac5124b4... ▸ 57,896,044,618,658,100,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000.792003956564819968 ERC20 (BeautyChain Token) from 0x09a34e01fbaa49f... to ➞ 0x0e823ffe0187275...
Value:	0 Ether ($0.00)
Gas Limit:	76737
Gas Used By Txn:	76737
Gas Price:	0.000000001 Ether (1 Gwei)
Actual Tx Cost/Fee:	0.000076737 Ether ($0.04)
Nonce & {Position}:	5 | {94}

来源：https://etherscan.io/tx/0xad89ff16fd1ebe3a0a7cf4ed282302c06626c1af33221ebe0d3a470aba4a660f

攻击结果



- 攻击在2018年4月22日发生，攻击发生后币值暴跌

来源：https://coinlib.io/coin/BEC4/Beauty+Chain

关于OKEx暂停BEC交易和提现的公告【更新】

OKEx
22 天前 · 更新于

尊敬的OKEx用户，

2018年4月22日13时左右，BEC出现异常交易，应BeautyChain (BEC)项目方的要求，暂时关闭BEC/USDT、BEC/BTC、BEC/ETH的交易和BEC的提现，具体开放时间另行通知。OKEx会随时与项目方保持联络，待项目方有最新进展我们会第一时间公布，给您带来的不便深表歉意，感谢您对OKEx的支持和理解。

OKEx

2018-4-22

----------2018-4-24 17:00 更新----------

经讨论决定OKEx将BEC的BEC/USDT、BEC/BTC、BEC/ETH三个交易区的交易数据回滚至香港时间2018-04-22 13:18:00，在此时间以后所有参与BEC交易的账户会根据BEC的交易账单记录进行回滚，其他币种的交易记录不受影响，回滚以后所有参与BEC交易的账户均不会有任何资金损失。在2018年4月22日13:18:00以后没有参与BEC交易的账户不受此次数据回滚的影响。BEC的交易和提现开放时间将另行通知。

OKEx

2018-4-24

来源：https://support.okex.com/hc/zh-cn/articles/360002944212

预防措施

- SafeMath库
  - 只要通过SafeMath提供的乘法计算amount，就可以很容易地检测到溢出

```
library SafeMath {

  /**
  * @dev Multiplies two numbers, throws on overflow.
  */
  function mul(uint256 a, uint256 b) internal pure returns (uint256 c) {
    if (a == 0) {
      return 0;
    }
    c = a * b;
    assert(c / a == b);
    return c;
  }
```

来源：https://github.com/OpenZeppelin/openzeppelin-solidity/blob/master/contracts/math/SafeMath.sol

```
  function batchTransfer(address[] _receivers, uint256 _value) public whenNotPaused returns (bool) {
    uint cnt = _receivers.length;
    uint256 amount = uint256(cnt) * _value;
    require(cnt > 0 && cnt <= 20);
    require(_value > 0 && balances[msg.sender] >= amount);

    balances[msg.sender] = balances[msg.sender].sub(amount);
    for (uint i = 0; i < cnt; i++) {
        balances[_receivers[i]] = balances[_receivers[i]].add(_value);
        Transfer(msg.sender, _receivers[i], _value);
    }
    return true;
  }
}
```

来源：https://etherscan.io/address/0xc5d105e63711398af9bbff092d4b6769c82f793d#code