SmartDec

Joyso Smart Contracts Security Analysis

This report is public.

Published: March 13, 2018

Abstract

In this report we consider the security of the Joyso project. Our task is to find and describe security issues in the smart contracts of the platform.

Procedure

In our audit, we consider the following crucial features of the smart contract code:

1. Whether the code is secure.
2. Whether the code corresponds to the documentation (including whitepaper).
3. Whether the code meets best practices in efficient use of gas, code readability, etc.

We perform our audit according to the following procedure:

- automated analysis
  - we scan project's smart contracts with our own Solidity static code analyzer SmartCheck
  - we scan project's smart contracts with several publicly available automated Solidity analysis tools such as Remix, Oyente, and Solhint
  - we manually verify (reject or confirm) all the issues found by tools
- manual audit
  - we manually analyze smart contracts for security vulnerabilities
  - we check smart contracts logic and compare it with the one described in the whitepaper
  - we check ERC20 compliance
  - we run tests and check code coverage
- report
  - we reflect all the gathered information in the report

Disclaimer

The audit does not give any warranties on the security of the code. One audit can not be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, security audit is not an investment advice.

Checked vulnerabilities

We have scanned Joyso smart contracts for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered (the full list includes them but is not limited to them):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DOS with (Unexpected) revert
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level
- Address hardcoded
- Using delete for arrays
- Integer overflow/underflow
- Locked money
- Private modifier
- Revert/require functions
- Using var
- Visibility
- Using blockhash
- Using SHA3
- Using suicide
- Using throw
- Using inline assembly

Project Overview

In our analysis we consider Joyso smart contracts code (Git repository, version on commit 67b0c6c or joyso-contracts-67b0c6c210fd2034734df57987b917fa72b9d103.zip, sha1sum 4fd1b984f2ca9de4e90d4943e1c1cec31de25f5d).

The latest version of the code

We have performed the check of the fixed vulnerabilities in the latest version of code (Git repository, version on commit 1b7986af880205a5d08d7851a73f95ba168fc7ca).

Project Architecture

For the audit, we have been provided with the following set of solidity files:

- JoysoDataDecoder.sol
- JoysoMock.sol
- Joyso.sol
- Migrations.sol
- libs/TestToken.sol
- libs/BasicToken.sol
- libs/ERC20Basic.sol
- libs/ERC20.sol
- libs/Ownable.sol
- libs/SafeMath.sol
- libs/StandardToken.sol

The files are the part of the truffle project. The project also contains tests, deploy scripts, and several files that are beyond the scope of the audit.
Main scope of the audit is Joyso.sol and its dependencies.
The total volume of the Solidity code that has been audited is 537 lines of code.
The project compiles successfully with `truffle compile` command (see the Compilation output in the Appendix)
The project successfully passes the tests (`truffle test` command, Tests output).

Code Logic

Joyso is token exchange contract that allows users to deposit allowed tokens and ETH, exchange tokens and ETH, withdraw tokens and ETH. Administrator and contract owner can process user orders, add new administrators, add new tokens, process user request of withdraw.
Contract implements standard Ownable logic. The implementation is similar to OpenZeppelin Ownable contract (99% identical).

Contract has complex data encoding and decoding logic, uses several cryptographic functions to sign and verify user data.
List of public function with descriptions by SmartDec team:

```
function Joyso (address _joysoWallet, address _joyToken) public
```

Call restrictions:
- everyone can make a call

Parameters requirements:
- no requirements

Logic:
- contract constructor
- function initializes several variables
  - beneficiary wallet as `_joysoWallet`
  - fee payment token as `_joyToken`
  - first admin as contract creator address

```
function depositToken (address token, uint256 amount) public
```

Call restrictions:
- everyone can make a call

Parameters requirements:
- `token` address must be listed in `address2Id` mapping

Logic:
- function adds caller to user list - `address2Id` mapping
- function transfers tokens of `token` contract from caller address to contract address

```
function depositEther () public
```

Call restrictions:
- everyone can make a call

Parameters requirements:
- no requirements

Logic:
- function adds caller to user list - `address2Id` mapping
- function accepts ETH from caller address to Joyso contract address

```
function withdraw (address token, uint256 amount) public
```

Call restrictions:
- everyone can make a call

Parameters requirements:
- `token` address must be listed in `address2Id` mapping or be `0`
- Current timestamp must be greater than user locked block

Logic:

- function transfers tokens of `token` contract from Joyso contract address to caller if `balances` allows it
- function transfers ETH from Joyso contract address to caller if `token` is `0` and `balances` allows it

```
function lockMe () public
```

Call restrictions:
- everyone can make a call

Parameters requirements:
- no requirements

Logic:
- function locks caller account for withdraw for 10 days

```
function unlockMe () public
```

Call restrictions:
- everyone can make a call

Parameters requirements:
- no requirements

Logic:
- function unlocks caller account for withdraw

```
function getTime () public view returns (uint256)
```

Call restrictions:
- everyone can make a call

Parameters requirements:
- no requirements

Logic:
- function returns current block timestamp

```
function getBalance (address token, address account) public view
returns (uint256)
```

Call restrictions:
- everyone can make a call

Parameters requirements:
- no requirements

Logic:
- function returns current balance of `token` of `address` user

```
function getWithdrawDataHash (uint256 amount, uint256 gas, uint256
data) public view returns (bytes32)
```

Call restrictions:
- everyone can make a call

Parameters requirements:

- no requirements

Logic:

- function returns hash of parameters for signing and transferring to an external system for further processing via `removeByAdmin` function

```
function getOrderDataHash (uint256 amountSell, uint256 amountBuy,
uint256 gasFee, uint256 data) public view returns (bytes32)
```

Call restrictions:

- everyone can make a call

Parameters requirements:

- no requirements

Logic:

- function returns hash of parameters for signed and passing to external system for future processing via `matchByAdmin` function

```
function verify (bytes32 hash, address sender, uint8 v, bytes32 r,
bytes32 s) public pure returns (bool)
```

Call restrictions:

- everyone can make a call

Parameters requirements:

- no requirements

Logic:

- function verifies that `hash` is signed by `sender` via `v`, `r`, `s` signature

```
function registerToken (address tokenAddress, uint256 index)
external onlyAdmin
```

Call restrictions:

- only from address listed in `isAdmin` mapping or owner address

Parameters requirements:

- no requirements

Logic:

- function adds `token` address as token listed with `index` in corresponding mappings

```
function addToAdmin (address admin, bool isAdd) onlyAdmin external
```

Call restrictions:

- only from address listed in `isAdmin` mapping or owner address

Parameters requirements:

- no requirements

Logic:

- function adds or removes `admin` address from `isAdmin` mapping

```
function withdrawByAdmin (uint256[] inputs) onlyAdmin external
```

Call restrictions:

- call must be only from address listed in `isAdmin` mapping or owner address

Parameters requirements:

- all input parameters must meet set of requirements, critical ones are
  - data for withdraw request must be signed by corresponding user
  - user must have enough `balance` of tokens or ETH
  - checks whether orders were not canceled

Logic:

- call force withdraw of tokens and ETH only by user signed request

```
function matchByAdmin (uint256[] inputs) onlyAdmin public
```

Call restrictions:

- call must be only from address listed in `isAdmin` mapping or owner address

Parameters requirements:

- all input parameters must meet set of requirements, critical ones are
  - data for order matching must be signed by corresponding users
  - all users must have enough `balance` of tokens or ETH
  - maker’s price must not be worse than the taker's order
  - orders must not be canceled

Logic:

- function exchanges tokens and ETH with prematched taker and makers orders

```
function cancelByAdmin(uint256[] inputs) onlyAdmin external
```

Call restrictions:

- call must be only from address listed in `isAdmin` mapping or owner address

Parameters requirements:

- all input parameters must pass set of requirements, critical one is
  - data for order matching must be signed by corresponding users

Logic:

- function makes exchange or withdraw order canceled

Automated Analysis

We used several publicly available automated Solidity analysis tools. Here are the combined results of SmartCheck, Solhint, and Remix. Oyente has found no issues.
All the issues found by tools were manually checked (rejected or confirmed). False positives are constructions that were discovered by the tools as vulnerabilities but do not consist a security threat. True positives are constructions that were discovered by the tools as vulnerabilities and can actually be exploited by attackers or lead to incorrect contracts operation. Cases when these issues lead to actual bugs or vulnerabilities are described in the next section.

Tool	Vulnerability	False positives	True positives
Remix	Defines a return type but never explicitly returns a value	1	
	Gas requirement of function is high	17	
	Potential Violation of Checks-Effects-Interaction pattern	1	1
	Variables have very similar names	7	
Total Remix		26	1
SmartCheck	Dos with revert	8	
	Gas limit and loops		1
	No payable fallback	5	
	Pragmas version	4	2
	Reentrancy external call	9	1
	Unchecked math	8	
	Visibility		5
Total SmartCheck		34	9
Solhint	Compiler version must be fixed	5	1

Explicitly mark visibility of state	5	
Total Solhint	10	1
Overall Total	**70**	**11**

Manual Analysis

The contracts were completely manually analyzed, their logic was checked and compared with the one described in the documentation. Besides, the results of the automated analysis were manually verified. All confirmed issues are described below.

Critical issues

Critical issues seriously endanger smart contracts security. We highly recommend fixing them.

The audit showed no critical issues.

Medium severity issues

Medium issues can influence smart contracts operation in current implementation. We highly recommend addressing them.

Mixing entities in mapping

Both `user` and `token` entities are stored in the same `address2Id` mapping (`addUser` and `registerToken` functions). This allows attacker to add malicious token contracts address to `address2Id` mapping via `depositEther` and thus bypass the token check in `depositToken` function (Joyso.sol, line 56). Furthermore, more complex social engineering attacks are possible. We highly recommend implementing two separate mappings for `user` and `token` entities.
The issue has been fixed and is not present in the latest version of the code.

Overpowered owner

The contract owner can change tokens' IDs to address mapping using `registerToken` function with already used `index` parameter. We highly recommend banning this possibility since it might be undesirable for contract users.

Low severity issues

Low severity issues can influence smart contracts operation in future versions of code. We recommend taking them into account.

Possible out of gas

During tests, some functions consume a lot of gas (>4800000, which would even exceed the block gas limit until recent changes). `matchByAdmin` function iterates over the array passed as a parameter, which may consume much gas, too. We highly recommend checking gas amount in tests. Besides, we recommend avoiding loops with big or unknown number of steps.

Potential Violation of Checks-Effects-Interaction pattern

There is a Checks-Effects-Interaction violation in Joyso.sol, lines 58-59:

```
require(Token(token).transferFrom(msg.sender, this, amount));
balances[token][msg.sender] =
balances[token][msg.sender].add(amount);
```

In this case the CEI violation does not lead to an actual vulnerability. However, we highly recommend following best practices including Checks-Effects-Interactions pattern since it helps to avoid many serious vulnerabilities.

Using OpenZeppelin files in repo

OpenZeppelin files are added to the repo instead of being connected via npm. We highly recommend using npm in order to guarantee that original OpenZeppelin contracts are used with no modifications.

Redundant code

There are both `require` check of overflow and safeMath functions in Joyso.sol, lines 71, 166, 169-170. We recommend removing `require` checks since they are redundant.
The issue has been fixed and is not present in the latest version of the code.

Added tokens audit

Since Joyso contract interacts with third party token contracts, we recommend auditing these contracts before adding tokens to the platform.

Pragmas version

Solidity source files indicate the versions of the compiler they can be compiled with.
Example:

```
pragma solidity ^0.4.19; // bad: compiles w 0.4.19 and above
pragma solidity 0.4.19; // good: compiles w 0.4.19 only
```

We recommend following the latter example, as future compiler versions may handle certain language constructions in a way the developer did not foresee. Besides, we recommend using the latest compiler version — 0.4.19 at the moment (0.4.13 is used in the contracts).

Implicit visibility level

There are variables and functions with implicit visibility level in the code (Joysol.sol, lines 11, 12, 13, 14; JoysoDataDecoder.sol, line 5).
We recommend specifying visibility levels (`public`, `private`, `external`, `internal`) explicitly and correctly in order to improve code readability.
The issue has been fixed and is not present in the latest version of the code.

Conclusion

In this report we have considered the security of Joyso smart contracts. We performed our audit according to the procedure described above.
The initial audit showed high code quality and no critical issues. However, several medium and low severity issues were found. Most of them were successfully fixed by developer in the latest version of the code.

This analysis was performed by SmartDec.

Pavel Yushchenko, Chief Technical Officer
Yaroslav Alexandrov, Head of Development Department
Ivan Ivanitskiy, Chief Analytics Officer
Elizaveta Kharlamova, Analyst
Alexander Seleznev, Chief Business Development Officer

Sergey Pavlin, Chief Operating Officer 

March 13, 2018

Appendix

Code coverage

```
----------------------|----------|----------|----------|----------
|----------------|
File                  |  % Stmts | % Branch |  % Funcs |  % Lines
|Uncovered Lines |
----------------------|----------|----------|----------|----------
|----------------|
 contracts/           |    99.46 |    97.56 |    97.62 |    99.48
|                |
  Joyso.sol           |    99.33 |    97.44 |    96.43 |    99.36
|            101 |
  JoysoDataDecoder.sol |      100 |      100 |      100 |      100 |
|
  JoysoMock.sol       |      100 |      100 |      100 |      100
|                |
----------------------|----------|----------|----------|----------
|----------------|
All files             |    99.46 |    97.56 |    97.62 |    99.48
|                |
----------------------|----------|----------|----------|----------
|----------------|
```

Compilation output

```
Compiling ./contracts/Joyso.sol...
Compiling ./contracts/JoysoDataDecoder.sol...
Compiling ./contracts/JoysoMock.sol...
Compiling ./contracts/Migrations.sol...
Compiling ./contracts/TestToken.sol...
Compiling ./contracts/libs/BasicToken.sol...
Compiling ./contracts/libs/ERC20.sol...
Compiling ./contracts/libs/ERC20Basic.sol...
Compiling ./contracts/libs/Ownable.sol...
```

```
Compiling ./contracts/libs/SafeMath.sol...
Compiling ./contracts/libs/StandardToken.sol...
```

Tests output

```
TestDecoder
    ✓ testDecodeWithdraw (127ms)
    ✓ testDecodeOrderData (64ms)

  Contract: cancel.js
    ✓ cancelByAdmin should update the user nonce (990ms)
    ✓ nonce should more than current nonce (1069ms)
    ✓ pay joy for fee to cancel the order (923ms)
    ✓ cancel should fail if user's signature is wrong (943ms)
    ✓ cancel should fail if user's balance is not enough (990ms)
    ✓ match should fail if the taker order's nonce is less than
userNonce (1144ms)
    ✓ match should fail if the maker order's nonce is less than
userNonce (1089ms)

  Contract: match.js
    ✓ case1, details in google doc (1120ms)
    ✓ case2, details in google doc (1408ms)
    ✓ case3, details in google doc (1190ms)
    ✓ case4 (1066ms)
    ✓ case5 (1001ms)
    ✓ case6 trade all the user balance (1077ms)
    ✓ taker paid Joy for fee (1109ms)
    ✓ gasFee can only charge once for each order (1626ms)
    ✓ gasFee (JOY) can only charge once for each order (1523ms)
    ✓ it should fail if taker's signature is wrong. (1130ms)
    ✓ it should fail if the maker's signature is wrong (941ms)
    ✓ it should fail if the price taker's price is worse than
maker's (987ms)
    ✓ split a taker order into two transactions (1261ms)

  Contract: Joyso misc.js
    ✓ it should fail if not admin send the match (895ms)
```

```
    ✓ deposit should fail, if the deposit token is not
registered  (231ms)
    ✓ it should fail if the token is not approved to the joyso
contract (217ms)
    ✓ registerToken's index should more than 1 (164ms)
    ✓ the same token can not registered twice  (190ms)
    ✓ add new admin (1094ms)
    ✓ for case1, maker and taker order exchage the place should
still success (1016ms)

  Contract: Joyso mock
    ✓ withdraw ether directly by user (1139ms)
    ✓ withdraw token directly by user (1168ms)
    ✓ unlockMe should reset the user lock (837ms)
    ✓ withdraw ether should fail if no balance (837ms)
    ✓ withdraw token should fail if no balance (787ms)
    ✓ withdraw directly should fail (800ms)

  Contract: gas analysis
2 order match: 149599
3 order match: 192958
4 order match: 251318
5 order match: 310075
6 order match: 368253
7 order match: 426710
8 order match: 485252
9 order match: 543729
10 order match: 602379
withdraw by admin (ether): 75302
withdraw by admin (token): 103595
    ✓ case 1  (10103ms)

  Contract: joyso withdraw
    ✓ withdraw token, pay by ether (899ms)
    ✓ withdraw joy, pay by ether (980ms)
    ✓ withdraw ether, pay by ether (928ms)
    ✓ withdraw token, pay by JOY (891ms)
    ✓ withdraw joy, pay by JOY (803ms)
    ✓ withdraw ether, pay by JOY (1005ms)
    ✓ withdraw token, pay by token (838ms)
    ✓ it should fail if use the same withdraw hash (790ms)
    ✓ it should fail if the signature is wrong (937ms)
```

```
    ✓ withdraw token, pay by ether. Should fail if no token balance.
(779ms)
    ✓ withdraw token, pay by ether. Should fail if no ether balance.
(753ms)

  47 passing (54s)
```

Solhint output

```
contracts/Joyso.sol
    8:1    error    Definition must be surrounded with two blank
line indent                                two-lines-top-level-
separator
    8:1    error    Contract has 16 states declarations but allowed
no more than 15                          max-states-count
   73:5    warning  Event and function names must be
different                                              no-simple-
event-func-name
   86:14   error    Expression indentation is incorrect. Required no
spaces before (                          expression-indent
   91:14   error    Expression indentation is incorrect. Required no
spaces before (                          expression-indent
   95:9    error    Expression indentation is incorrect. Required no
spaces after queueLength                 expression-indent
  101:16   warning  Avoid to make time-based decisions in your
business logic                                  not-rely-on-time
  104:5    error    Function order is incorrect, external_const
function can not go after public function  func-order
  116:2    error    Line length must be no more than 120 but current
length is 131                            max-line-length
  128:5    error    Function order is incorrect, external function
can not go after public function         func-order
  129:17   error    Expression indentation is incorrect. Required no
spaces before (                          expression-indent
  130:17   error    Expression indentation is incorrect. Required no
spaces before (                          expression-indent
  135:53   error    Visibility modifier must be first in list of
modifiers                                   visibility-modifier-order
  135:5    error    Function order is incorrect, external function
can not go after public function         func-order
  139:5    error    Function order is incorrect, external function
can not go after public function         func-order
```

```
  139:68   error    Function body contains 55 lines but allowed no
more than 50 lines                      function-max-lines
  139:49   error    Visibility modifier must be first in list of
modifiers                                 visibility-modifier-order
  160:17   error    Variable name must be in
mixedCase                                                    var-
name-mixedcase
  170:17   error    Expression indentation is incorrect. Required no
spaces before (                         expression-indent
  171:17   error    Expression indentation is incorrect. Required no
spaces before (                         expression-indent
  197:2    error    Line length must be no more than 120 but current
length is 122                           max-line-length
  198:46   error    Visibility modifier must be first in list of
modifiers                                 visibility-modifier-order
  198:5    error    Definitions inside contract / library must be
separated by one line                    separate-by-one-line-in-
contract
  198:65   error    Function body contains 59 lines but allowed no
more than 50 lines                      function-max-lines
  198:5    error    Function order is incorrect, external function
can not go after public function        func-order
  231:2    error    Line length must be no more than 120 but current
length is 146                           max-line-length
  232:17   error    Expression indentation is incorrect. Required no
spaces before (                         expression-indent
  232:2    error    Line length must be no more than 120 but current
length is 132                           max-line-length
  233:17   error    Expression indentation is incorrect. Required no
spaces before (                         expression-indent
  233:2    error    Line length must be no more than 120 but current
length is 157                           max-line-length
  237:17   error    Expression indentation is incorrect. Required no
spaces before (                         expression-indent
  241:49   error    Expression indentation is incorrect. Required
space after +=                            expression-indent
  242:2    error    Line length must be no more than 120 but current
length is 154                           max-line-length
  242:21   error    Expression indentation is incorrect. Required no
spaces before (                         expression-indent
  243:2    error    Line length must be no more than 120 but current
length is 140                           max-line-length
  243:21   error    Expression indentation is incorrect. Required no
spaces before (                         expression-indent
  244:2    error    Line length must be no more than 120 but current
length is 161                           max-line-length
```

```
  245:2    error    Line length must be no more than 120 but current
length is 174                          max-line-length
  245:21   error    Expression indentation is incorrect. Required no
spaces before (                        expression-indent
  246:2    error    Line length must be no more than 120 but current
length is 167                          max-line-length
  255:13   error    Expression indentation is incorrect. Required
space after delete                       expression-indent
  256:13   error    Expression indentation is incorrect. Required no
spaces after queueLength               expression-indent
  260:5    error    Function order is incorrect, external function
can not go after public function        func-order
  260:46   error    Visibility modifier must be first in list of
modifiers                                  visibility-modifier-order
  272:17   error    Variable name must be in
mixedCase                                                        var-
name-mixedcase
  307:2    error    Line length must be no more than 120 but current
length is 157                          max-line-length
  317:2    error    Line length must be no more than 120 but current
length is 194                          max-line-length
  337:2    error    Line length must be no more than 120 but current
length is 158                          max-line-length
  348:9    error    Statement indentation is incorrect. Required
space after if                           statement-indent
  358:9    error    Expected indentation of 12 spaces but found
8                                        indent
  359:9    error    Expected indentation of 12 spaces but found
8                                        indent
  360:13   error    Expected indentation of 16 spaces but found
12                                       indent
  361:13   error    Expected indentation of 16 spaces but found
12                                       indent
  362:13   error    Expected indentation of 16 spaces but found
12                                       indent
  363:17   error    Expected indentation of 20 spaces but found
16                                       indent
  364:13   error    Expected indentation of 16 spaces but found
12                                       indent
  365:17   error    Expected indentation of 20 spaces but found
16                                       indent
  366:13   error    Expected indentation of 16 spaces but found
12                                       indent
  367:2    error    Line length must be no more than 120 but current
length is 144                          max-line-length
  368:13   error    Expected indentation of 16 spaces but found
12                                       indent
```

```
  369:13   error    Expected indentation of 16 spaces but found
12                                      indent
  370:9    error    Expected indentation of 12 spaces but found
8                                       indent
  371:13   error    Expected indentation of 16 spaces but found
12                                      indent
  372:9    error    Expected indentation of 12 spaces but found
8                                       indent
  373:5    error    Expected indentation of 8 spaces but found
4                                        indent
  375:2    error    Line length must be no more than 120 but current
length is 146                          max-line-length
  386:2    error    Line length must be no more than 120 but current
length is 144                          max-line-length
  391:2    error    Line length must be no more than 120 but current
length is 137                          max-line-length
  392:115  error    Semicolon must not have spaces
before                                                  no-spaces-
before-semicolon
  395:2    error    Line length must be no more than 120 but current
length is 161                          max-line-length

contracts/JoysoDataDecoder.sol
    3:1  error  Definition must be surrounded with two blank line
indent        two-lines-top-level-separator
   79:2  error  Line length must be no more than 120 but current
length is 128  max-line-length
   94:2  error  Line length must be no more than 120 but current
length is 124  max-line-length
  114:2  error  Line length must be no more than 120 but current
length is 124  max-line-length

✖ 74 problems (72 errors, 2 warnings)
```

Solium output

```
contracts/Joyso.sol
  101:15    warning    Avoid using 'now' (alias to
'block.timestamp').                              security/no-block-
members
  241:47    warning    Assignment operator must have exactly single
space on both sides of it.    operator-whitespace

contracts/JoysoDataDecoder.sol
```

```
  22:4     error    "decodeOrderTakerFee": Avoid assigning to
function parameters.           security/no-assign-params
  33:4     error    "decodeOrderMakerFee": Avoid assigning to
function parameters.           security/no-assign-params
  44:4     error    "decodeOrderJoyPrice": Avoid assigning to
function parameters.           security/no-assign-params
  54:4     error    "decodeOrderTokenIdAndIsBuy": Avoid assigning to
function parameters.    security/no-assign-params
  54:4     error    "decodeOrderTokenIdAndIsBuy": Avoid assigning to
function paramters.    security/no-assign-params
  79:4     error    "decodeWithdrawData": Avoid assigning to
function parameters.            security/no-assign-params
  109:4    error    "genUserSignedWithdrawData": Avoid assigning to
function parameters.     security/no-assign-params

✖ 7 errors, 2 warnings found.
```