SmartDec

Wibson Smart Contracts Security Analysis

This report is public.

Published: May 24, 2019.

Abstract

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

Disclaimer

The audit does not give any warranties on the security of the code. One audit cannot 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.

Summary

In this report, we considered the security of Wibson smart contracts. We performed our audit according to the procedure described below.

The audit showed neither critical nor medium nor low severity issues.

General recommendations

The contracts code is of excellent code quality. The audit did not reveal any issues that endanger project security.

Checklist

Security

The audit showed no vulnerabilities.
Here by vulnerabilities we mean security issues that can be exploited by an external attacker. This does not include low severity issues, documentation mismatches, overpowered contract owner, and some other kinds of bugs.

Compliance with the documentation

The audit showed no discrepancies between the code and the provided documentation.

ERC20 compliance

We checked ERC20 compliance during the audit. The audit showed that **StandardToken** contract was fully ERC20 compliant.

ERC20 MUST

The audit showed no ERC20 “MUST” requirements violations.

ERC20 SHOULD

The audit showed no ERC20 “SHOULD” requirements violations.

Tests

The audit showed that the code was covered with tests sufficiently.

The text below is for technical use; it details the statements made in Summary and General recommendations.

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 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 categorize all the found security issues in accordance with the classification in order to identify developers' shortcomings
  - we check smart contracts logic and compare it with the one described in the documentation
  - we check ERC20 compliance
  - we run tests and check code coverage
- report
  - we reflect all the gathered information in the report

Checked vulnerabilities

We have scanned Wibson 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
- Front running
- DoS with (unexpected) revert
- DoS with block gas limit
- Gas limit and loops
- Locked money
- Integer overflow/underflow
- Unchecked external call
- ERC20 Standard violation
- Authentication with tx.origin
- Unsafe use of timestamp
- Using blockhash for randomness
- Balance equality
- Unsafe transfer of ether
- Fallback abuse
- Using inline assembly
- Short address attack
- Private modifier
- Compiler version not fixed
- Style guide violation
- Unsafe type deduction
- Implicit visibility level
- Use delete for arrays
- Byte array
- Incorrect use of assert/require
- Using deprecated constructions

Project overview

Project description

In our analysis we consider Wibson specification (”drive-download-20190506T122010Z-001.zip”, sha1sum: 67e9c89133df6d0ffcc616a910d419ae88c60207) BatchPayments smart contracts’ code (version on commit 2ca443b5c56da67bfaba3670b4e45634b6ded495) and MarketPlace smart contracts’ code (version on commit cb76369a33c6a48e633332ef41dc9641a77f2b9f).

Project architecture

For the audit, we were provided with two truffle projects. Both projects are npm packages and include tests.

- Both projects successfully compile with `truffle compile` command (see Compilation output in Appendix)
- Both projects successfully pass all the tests with 100% coverage

Scope of work

All Solidity files were audited, excluding the following:

- **MassExit.sol** from BatchPayments project
- **MassExitLib.sol** from BatchPayments project

The total LOC of audited Solidity sources is 980.

Automated analysis

We used several publicly available automated Solidity analysis tools. Here are the combined results of SmartCheck, Solhint, and Remix scanning. All the issues found by tools were manually checked (rejected or confirmed).

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.

False positives are constructions that were discovered by the tools as vulnerabilities but do not consist a security threat.

Cases when these issues lead to actual bugs or vulnerabilities are described in the next section.

Tool	Rule	True positives	False positives
Remix	Use of inline assembly		6
	Constant but potentially should not be		6
	Defines a return type but never explicitly returns a value		1
	Potential Violation of Checks-Effects-Interaction pattern		2
	Use of ”now”		2
Total Remix		0	17
Solhint	Event and function names must be different		3
	Avoid to use inline assembly. It is acceptable only in rare cases		4
	Avoid to make time-based decisions in your business logic		2
Total Solhint		0	9
SmartCheck	Unsafe array’s length manipulation		1

	Extra gas consumption		2
	Hardcoded address		3
	Non-initialized return value		1
	Pure-functions should not read/change state		4
	Upgrade code to Solidity 0.5.x.		5
	Use of assembly		4
	Costly loop		1
	Use of SafeMath		1
	Using approve function of the ERC-20 token standard	1	
	Replace multiple return values with a struct		1
Total SmartCheck		1	23
Total Overall		**1**	**49**

Manual analysis

The contracts were completely manually analyzed, their logic was checkedand compared with the one described in the documentation. Besides, the results of the automated analysis were manually verified. All the 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.

The audit showed no medium severity issues.

Low severity issues

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

The audit showed no low severity issues.

Notes

ERC20 approve issue

There is ERC20 approve issue: changing the approved amount from a nonzero value to another nonzero value allows a double spending with a front-running attack.

We recommend instructing users to follow one of two ways:

- not to use `approve()` function directly and to use `increaseApproval()`/`decreaseApproval()` functions instead
- to change the approved amount to 0, wait for the transaction to be mined, and then to change the approved amount to the desired value

This analysis was performed by SmartDec.

Alexander Seleznev, Chief Business Development Officer
Boris Nikashin, Project Manager
Igor Sobolev, Analyst
Alexander Drygin, Analyst

May 24, 2019

Appendix

Compilation output

```
./node_modules/.bin/truffle compile
Compiling ./contracts/Accounts.sol...
Compiling ./contracts/BatPay.sol...
Compiling ./contracts/Challenge.sol...
Compiling ./contracts/Data.sol...
Compiling ./contracts/IERC20.sol...
Compiling ./contracts/Merkle.sol...
Compiling ./contracts/Migrations.sol...
Compiling ./contracts/Payments.sol...
Compiling ./contracts/SafeMath.sol...
Compiling ./contracts/StandardToken.sol...
Compiling ./contracts/TestHelper.sol...
Writing artifacts to ./build/contracts

Compiling your contracts...
===========================
> Compiling ./contracts/DataExchange.sol
> Compiling ./contracts/Migrations.sol
> Artifacts written to /home/igor/job/audits/Wibson/wibson/c
ore/marketplace/build/contracts
> Compiled successfully using:
   - solc: 0.5.7+commit.6da8b019.Emscripten.clang
```

Tests output

```
  Contract: Accounts
    deposits
       Should fail on not enough approval (221ms)
       Should accept deposits for new accounts (156ms)
       Should record deposits on account storage (208ms)
       Should reject 0-token deposits
    withdraw
       Should accept withdrawals for existing accounts (212m
s)
       Should reject withdrawals for existing accounts when
sender is not the owner (128ms)
       Should reject withdrawals for invalid accounts (191ms
```

```
)
       Should reject withdrawals for sums larger than balanc
e (159ms)
       Should reject withdrawals for $0 (163ms)
    registration
       deposit() should register new accounts (274ms)
       Bulk registration should reserve new accounts (132ms)
       Bulk registration root hashes should be stored (151ms
)
       Bulk registration should respect account limits (61ms
)
       Bulk registration should fail for n == 0
       Bulk registration should fail for rootHash == 0
       register() adds 1 account at a time (125ms)
       register() emits AccountRegistered event (57ms)
    claim
       claim happy case (42ms)
       cannot claim using an invalid bulkId (56ms)
       cannot claim using an id not in the bulk (79ms)
       cannot claim using an invalid proof

  Contract: BatPay
    input Validation
       cannot create a BatPay with token address at zero (27
3ms)
       cannot create a BatPay with maxBulk at zero (261ms)
       cannot create a BatPay with maxTransfer at zero (272m
s)
       cannot create a BatPay with challengeBlocks at zero (
268ms)
       cannot create a BatPay with challengeStepBlocks at ze
ro (261ms)
       cannot create a BatPay with collectStake at zero (259
ms)
       cannot create a BatPay with challengeStake at zero (2
58ms)
       cannot create a BatPay with unlockBlocks at zero (340
ms)
       cannot create a BatPay with maxCollectAmount at zero
(282ms)
    misc
       cannot obtain the balance for invalid id (46ms)

  Contract: challenge
```

```
     should complete a full challenge game #0 (269ms)
     should complete a full challenge game #1 (276ms)
     should claim a success after no response for challenge_
1 (115ms)
     should claim a failure after no response for challenge_
2 (180ms)
     should claim a success after no response for challenge_
3 (233ms)
     should reject an invalid index (223ms)
     should reject an invalid payData (202ms)
     should reject an invalid amount (78ms)
     should reject an invalid payment amount (196ms)
     should reject a locked payment (184ms)
     Should challenge inline bulkRegistered accounts (857ms)
     challenger winning should get back both collectStake +
challengeStake (251ms)
     delegate winning should get challengeStake (386ms)

  Contract: Challenge
    getDataSum
       returns the sum of the amounts in data
       returns the sum of the amounts even when there is onl
y one element in data
       rejects when the data has wrong format (45ms)
       rejects when the summarization causes an overflow
    getDataAtIndex
       returns the amount and payIndex in data at the specif
ied index
       returns the amount and payIndex even when there is on
ly one element in data
       rejects when the data has wrong format (47ms)
       rejects when index does not exist in data
    getPayDataSum
       returns the sum when the id is present in payData
       returns zero when the id is not present in payData
       rejects when the payData has wrong format (125ms)
    getPayDataCount
       returns the record count
       returns zero when there are no records present in pay
Data
       rejects when the payData has wrong format (124ms)
    recoverHelper
       Should return 0x0 if signature has wrong length
       Should return 0x0 if version v is not correct
```

```
  merkle lib
    sha3
       should pass test vectors with uint256
       should match solidity's sha3
       should pass test vectors with BigNumbers
       should pass test vectors with uint32
    merkle tree
       should calculate root hash
       should calculate root hash for odd number of elements
       should generate proper proofs
       should match solidity's eval proofs (1319ms)

  Contract: Payments
    registerPayment
       should support up 100s of ids (104ms)
       should reject invalid ids (61ms)
       should reject amount zero as payment (60ms)
       should reject rootHash zero if newCount > 0 (77ms)
       should reject fee payment with no lock (64ms)
       should accept registerPayment+unlock with good key (1
84ms)
       should reject registerPayment+unlock with bad key (14
4ms)
       should accept registerPayment+refund after timeout (1
95ms)
       should reject registerPayment+refund before timeout (
162ms)
       should reject registerPayment if bytes per id is 0 (5
0ms)
       should reject registerPayment if bytes payData length
is invalid (48ms)
       should reject registerPayment if there are too many p
ayees (79ms)
       should reject registerPayment if balance is not enoug
h (107ms)
       should correctly substract balance on registerPayment
with new-count=0 (118ms)
       should correctly substract balance on registerPayment
with new-count=10 (129ms)
       should correctly substract balance on registerPayment
with new-count=1 (138ms)
       should correctly substract balance on registerPayment
with no payData (119ms)
    collect
```

```
        should collect (188ms)
        should instant-collect (192ms)
        should reuse slot (431ms)
        should pay fee on instant-collect (216ms)
        should pay fee (218ms)
        should withdraw if requested (262ms)
        should withdraw if requested instant (261ms)
        should reject if payIndex is less or equal to last co
llected payment ID (75ms)
        should reject if payIndex is invalid (73ms)
        should reject if invalid toAccountId
        should reject wrong fromPayIndex / Invalid Signature
(47ms)
        Should not allow race condition on collect (352ms)
        Should reject collects over maxCollectAmount (237ms)
        Should accept collects with just maxCollectAmount (28
1ms)

  SafeMath
    uint64 arithmetic
       mul64 with trivial case (zero)
       mul64 boundary for overflow
       mul64 should check uint64 overflow
       div64 should fail when divide by zero
       div64 should fail with remainder larger than uint64
       div64 happy case with reaminder = 0
       div64 happy case with remainder = 1
       sub64 should check uint64 overflow
       sub64 with positive result
       sub64 cannot produce a negative result
       sub64 with result = 0
       add64 should check uint64 overflow
       add64 should happy case
    uint32 arithmetic
       mul32 with trivial case (zero)
       mul32 boundary for overflow
       mul32 should check uint32 overflow
       div32 should fail when divide by zero
       div32 should fail with remainder larger than uint32
       div32 happy case with remainder = 0
       div32 happy case with remainder = 1
      sub32
         should check uint32 overflow
         with positive result
```

```
        cannot produce a negative result
        with result = 0
    add32
        adds correctly
        correctly adds up to maximum value
        fails on overflow
  uint256 arithmetic
    add256
        adds correctly
        correctly adds up to maximum value
        reverts on overflow
    sub256
        subtracts correctly
        reverts if subtrahend > minuend
    mul256
        multiplies correctly
        reverts on overflow
        multiplies by zero correctly
    div256
        divides correctly when remainder == 0
        divides correctly when remainder != 0
        reverts when divisor is zero
        divides zero correctly


  138 passing (32s)

Contract: DataExchange
    closeDataOrder
       emits an event when a DataOrder is closed (59ms)
       updates the DataOrder's closedAt field (90ms)
       closes an open order even if another order is created
(111ms)
       closes an open order even if another order is closed
(165ms)
       fails when called by other than the buyer (57ms)
       fails when order is already closed (101ms)
       fails when order does not exist (47ms)
       fails when order is invalid (54ms)

  Contract: DataExchange
    createDataOrder
       emits an event when a DataOrder is created (152ms)
       preserves params order when a DataOrder is created (1
```

```
30ms)
       assigns the sender as the buyer of the DataOrder (144
ms)
       adds the createdAt field to the DataOrder (136ms)
       adds the closedAt field to the DataOrder (126ms)
       cannot create a DataOrder if audience field is empty
(56ms)
       cannot create a DataOrder if price == 0 (58ms)
       cannot create a DataOrder if requestedData field is e
mpty (57ms)
       cannot create a DataOrder if termsAndConditionsHash f
ield is empty (59ms)
       cannot create a DataOrder if buyerUrl field is empty
(62ms)

  Contract: DataExchange
    registerNotary
       should register a new notary (80ms)
       should validate registered public URL. (112ms)
       should fail when passed an invalid url (49ms)
       should fail to replace a notary (119ms)

  Contract: DataExchange
    unregisterNotary
       should unregister a notary (51ms)
       should fail when passed an inexistent notary address
(48ms)
       should register notary after unregistering it (75ms)

  Contract: DataExchange
    updateNotaryUrl
       should update a registered notary (80ms)
       should fail when passed an invalid url (102ms)
       should fail when passed an invalid notary (53ms)

  28 passing (5s)
```

Solhint output

```
./contracts/Merkle.sol
  39:9   warning  Statement indentation is incorrect. Requir
ed space after for  statement-indent
```

```
  39:26  warning  Expression indentation is incorrect. Requi
red space after <   expression-indent

./contracts/Migrations.sol
   3:1   warning  Definition must be surrounded with two bla
nk line indent  two-lines-top-level-separator
   5:17  warning  Variable name must be in mixedCase
ar-name-mixedcase
  19:30  warning  Function param name must be in mixedCase
unc-param-name-mixedcase

./contracts/Payments.sol
  79:2  error  Line length must be no more than 120 but curr
ent length is 121  max-line-length

./contracts/SafeMath.sol
  7:1  warning  Definition must be surrounded with two blank
line indent  two-lines-top-level-separator

./contracts/TestHelper.sol
  10:1  warning  Definition must be surrounded with two blan
k line indent  two-lines-top-level-separator

 37 problems (1 error, 36 warnings)

./contracts/DataExchange.sol
  115:20  warning  Avoid to make time-based decisions in you
r business logic  not-rely-on-time
  136:37  warning  Avoid to make time-based decisions in you
r business logic  not-rely-on-time

 2 problems (0 errors, 2 warnings)
```

Solium output

```
contracts/Accounts.sol
  1:0      error      Inconsistent line-break style
inebreak-style
  12:0     warning    Contract 'Accounts' must be preceded b
y 2 blank lines.                                      blank-
```

```
lines
  80:4      warning    In case of more than 3 parameters, dro
p each into its own line.                                arg-ov
erflow
  87:16     error      Only use indent of 12 spaces.
ndentation
  87:62     error      Avoid use of arithmetic operation '+'
directly. Use SafeMath instead.                          zeppel
in/no-arithmetic-operations
  112:4     warning    Functions should be in order: construc
tor, fallback, external, public, internal, private    functi
on-order
  134:4     warning    Functions should be in order: construc
tor, fallback, external, public, internal, private    functi
on-order
  180:4     warning    Functions should be in order: construc
tor, fallback, external, public, internal, private    functi
on-order
  193:4     warning    Functions should be in order: construc
tor, fallback, external, public, internal, private    functi
on-order
  201:4     warning    Functions should be in order: construc
tor, fallback, external, public, internal, private    functi
on-order

contracts/BatPay.sol
  1:0     error    Inconsistent line-break style    linebreak
-style

contracts/Challenge.sol
  1:0       error      Inconsistent line-break style
inebreak-style
  122:16    error      Only use indent of 12 spaces.
ndentation
  122:16    error      Avoid use of arithmetic operation '-'
directly. Use SafeMath instead.                    zeppelin/n
o-arithmetic-operations
  123:8     error      Only use indent of 12 spaces.
ndentation
  172:28    error      Avoid use of arithmetic operation '-'
directly. Use SafeMath instead.                    zeppelin/n
o-arithmetic-operations
  182:4     warning    'challenge_1' doesn't follow the mixe
dCase notation                                      mixedcase
```

```
  206:4     warning    'challenge_2' doesn't follow the mixe
dCase notation                                      mixedcase
  228:4     warning    'challenge_3' doesn't follow the mixe
dCase notation                                      mixedcase
  237:16    error      Only use indent of 12 spaces.
ndentation
  238:8     error      Only use indent of 12 spaces.
ndentation
  251:4     warning    'challenge_4' doesn't follow the mixe
dCase notation                                      mixedcase
  259:16    error      Only use indent of 12 spaces.
ndentation
  262:16    error      Only use indent of 12 spaces.
ndentation
  263:8     error      Only use indent of 12 spaces.
ndentation
  273:16    error      Only use indent of 12 spaces.
ndentation
  274:8     error      Only use indent of 12 spaces.
ndentation
  286:4     warning    'challenge_success' doesn't follow th
e mixedCase notation                                mixedcase
  293:16    error      Only use indent of 12 spaces.
ndentation
  295:16    error      Only use indent of 12 spaces.
ndentation
  312:4     warning    'challenge_failed' doesn't follow the
mixedCase notation                                  mixedcase
  319:16    error      Only use indent of 12 spaces.
ndentation
  373:15    warning    Function "ecrecover": in case of more
than 3 arguments, drop each into its own line.    arg-overfl
ow

contracts/Data.sol
  1:0       error    Inconsistent line-break style
inebreak-style
  72:42     error    Avoid use of arithmetic operation '-' di
rectly. Use SafeMath instead.    zeppelin/no-arithmetic-oper
ations
  73:44     error    Avoid use of arithmetic operation '-' di
rectly. Use SafeMath instead.    zeppelin/no-arithmetic-oper
ations
```

```
contracts/IERC20.sol
  1:0    error    Inconsistent line-break style    linebreak
-style


contracts/Merkle.sol
  1:0      error      Inconsistent line-break style
inebreak-style
  30:1     warning    Line contains trailing whitespace
o-trailing-whitespace
  39:11    error      There should be exactly a single space
between the 'for' token and the parenthetic block representi
ng the conditional.    conditionals-whitespace
  41:16    error      Avoid use of arithmetic operation '/'
directly. Use SafeMath instead.
eppelin/no-arithmetic-operations

contracts/Migrations.sol
  1:0      error      Inconsistent line-break style
inebreak-style
  3:0      warning    Contract 'Migrations' must be preceded
by 2 blank lines.            blank-lines
  5:4      warning    'last_completed_migration' doesn't fol
low the mixedCase notation    mixedcase
  19:21    warning    'new_address' doesn't follow the mixed
Case notation                 mixedcase

contracts/Payments.sol
  1:0      error      Inconsistent line-break style
inebreak-style
  87:16    error      Only use indent of 12 spaces.
ndentation
  88:8     error      Only use indent of 12 spaces.
ndentation
  115:13   warning    Function "PaymentRegistered": in case
of more than 3 arguments, drop each into its own line.    ar
g-overflow
  144:4    warning    Functions should be in order: constru
ctor, fallback, external, public, internal, private       f
unction-order
  207:12   error      Assignment operator must have exactly
single space on both sides of it.                         op
erator-whitespace
  209:12   error      Only use indent of 16 spaces.
ndentation
```

```
  217:16    error      Only use indent of 12 spaces.
ndentation
  218:8     error      Only use indent of 12 spaces.
ndentation
  220:16    error      Only use indent of 12 spaces.
ndentation
  220:25    error      Avoid use of arithmetic operation '-'
directly. Use SafeMath instead.                              ze
ppelin/no-arithmetic-operations
  271:13    warning    Function "Collect": in case of more t
han 3 arguments, drop each into its own line.                a
rg-overflow
  278:4     warning    Functions should be in order: constru
ctor, fallback, external, public, internal, private          f
unction-order
  290:4     warning    'challenge_1' doesn't follow the mixe
dCase notation                                               m
ixedcase
  299:8     warning    Function "undefined": in case of more
than 3 arguments, drop each into its own line.               ar
g-overflow
  309:4     warning    'challenge_2' doesn't follow the mixe
dCase notation                                               m
ixedcase
  328:4     warning    'challenge_3' doesn't follow the mixe
dCase notation                                               m
ixedcase
  339:8     warning    Function "undefined": in case of more
than 3 arguments, drop each into its own line.               ar
g-overflow
  349:4     warning    'challenge_4' doesn't follow the mixe
dCase notation                                               m
ixedcase
  370:4     warning    'challenge_success' doesn't follow th
e mixedCase notation                                         m
ixedcase
  386:4     warning    'challenge_failed' doesn't follow the
mixedCase notation                                           mi
xedcase

contracts/SafeMath.sol
  1:0       error      Inconsistent line-break style
inebreak-style
  7:0       warning    Library 'SafeMath' must be preceded b
```

```
y 2 blank lines.                    blank-lines
  19:20     error      Avoid use of arithmetic operation '*'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  20:8      warning    Provide an error message for require(
).                                  error-reason
  20:16     error      Avoid use of arithmetic operation '/'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  21:8      warning    Provide an error message for require(
).                                  error-reason
  32:20     error      Avoid use of arithmetic operation '/'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  33:8      warning    Provide an error message for require(
).                                  error-reason
  45:8      warning    Provide an error message for require(
).                                  error-reason
  46:20     error      Avoid use of arithmetic operation '-'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  47:8      warning    Provide an error message for require(
).                                  error-reason
  59:20     error      Avoid use of arithmetic operation '+'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  60:8      warning    Provide an error message for require(
).                                  error-reason
  75:20     error      Avoid use of arithmetic operation '*'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  76:8      warning    Provide an error message for require(
).                                  error-reason
  76:16     error      Avoid use of arithmetic operation '/'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  77:8      warning    Provide an error message for require(
).                                  error-reason
  89:20     error      Avoid use of arithmetic operation '/'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  90:8      warning    Provide an error message for require(
).                                  error-reason
  102:8     warning    Provide an error message for require(
```

```
).                                        error-reason
  103:20    error      Avoid use of arithmetic operation '-'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  104:8     warning    Provide an error message for require(
).                                        error-reason
  116:20    error      Avoid use of arithmetic operation '+'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  117:8     warning    Provide an error message for require(
).                                        error-reason
  131:20    error      Avoid use of arithmetic operation '*'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  132:8     warning    Provide an error message for require(
).                                        error-reason
  132:16    error      Avoid use of arithmetic operation '/'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  144:20    error      Avoid use of arithmetic operation '/'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  156:8     warning    Provide an error message for require(
).                                        error-reason
  157:15    error      Avoid use of arithmetic operation '-'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  167:20    error      Avoid use of arithmetic operation '+'
directly. Use SafeMath instead.    zeppelin/no-arithmetic-op
erations
  168:8     warning    Provide an error message for require(
).                                        error-reason

contracts/StandardToken.sol
  1:0     error      Inconsistent line-break style
inebreak-style
  19:4    warning    In case of more than 3 parameters, drop
each into its own line.    arg-overflow

contracts/TestHelper.sol
  1:0     error      Inconsistent line-break style
inebreak-style
  10:0    warning    Contract 'TestHelper' must be preceded
by 2 blank lines.    blank-lines
```

```
 61 errors, 89 warnings found.

contracts/DataExchange.sol
  1:0       error      Inconsistent line-break style
inebreak-style
  115:19    warning    Avoid using 'now' (alias to 'block.ti
mestamp').    security/no-block-members
  136:36    warning    Avoid using 'now' (alias to 'block.ti
mestamp').    security/no-block-members

 1 error, 2 warnings found.
```