SmartDec

Saga Smart Contracts Security Analysis

This report is private.

Published: August 23, 2019.

Abstract . . . 2
Disclaimer . . . 2
Summary . . . 2
General recommendations . . . 2
Checklist . . . 3
Procedure . . . 4
Checked vulnerabilities . . . 5
Project overview . . . 6
Project description . . . 6
The latest version of the code . . . 6
Project architecture . . . 6
Scope of work . . . 6
Automated analysis . . . 7
Manual analysis . . . 9
Critical issues . . . 9
Medium severity issues . . . 9
Overpowered owner . . . 9
Low severity issues . . . 10
revert() vs require() (fixed) . . . 10
Gas limit and loops . . . 10
Redundant code (fixed) . . . 11
Notes . . . 12
Gas cost . . . 12
Appendix . . . 13
Code coverage . . . 13
Tests output . . . 14
Compilation output . . . 14
Solhint output . . . 18
Solium output . . . 18

Abstract

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

The initial audit showed no critical issues. However, one medium and a number of low severity issues were found. They do not endanger project security.

Some of the issues were fixed in the latest version of the code.

General recommendations

The contracts code is of exceptional code quality. The audit did not reveal any issues that endanger project security. However, we recommend limiting The power of the owner.

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 **SGA, SGN** contracts were 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 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 Saga 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 Saga documentation (”Saga Monetary Model.pdf”, sha1sum: e4d42d9e540a77a01c86137c052cecd7b81a1845) and smart contracts’ code (”smartdec-audit-saga-contracts-2-master.zip”, sha1sum: 4d82b4fb8d7b9a6332ab6e95a51298918e64e45a).

The latest version of the code

After the initial audit, some fixes were applied and the code was updated to the latest version (”smartdec-audit-saga-contracts-3-master.zip” , sha1sum: ecc25be7c461605021347a3072fafbe94b573c19).

Project architecture

For the audit, we were provided with the truffle project. The project is an npm package and includes tests.

- The project successfully compiles with `node scripts/rebuild-all.js` command (see Compilation output in Appendix)
- The project successfully passes all the tests with 100% coverage

Scope of work

All solidity files were audited, excluding the following:

- all files under **helpers** folder
- all files under **interfaces** folder
- **Migrations.sol**
- **MultiSigWallet.sol**

The total LOC of audited Solidity sources is 1535.

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
SmartCheck	Locked money		2
	Costly loop	3	8
	Overpowered role		9
	Multiplication after division		3
	Revert inside the if-operator	2	
	Private modifier		5
	Upgrade code to Solidity 0.5.x		11
	Extra gas consumption		11
	Implicit visibility level		2
	Unsafe array’s length manipulation		3
	Non-initialized return value		2
	Use of assembly		1
	Use of safemath		9

Total SmartCheck		5	66
Remix	Fallback function requires too much gas		1
	Potential violation of checks-effects-interaction pattern		22
	Constant but potentially should not be		8
	Should be constant but is not		33
	Use of inline assembly		1
	Use of ”now”		1
	Use of ”send”		1
Total Remix		0	67
Solhint	Fallback function must be simple		1
	Avoid to make time-based decisions in your business logic		1
Total Solhint		0	2
Total Overall		**5**	**135**

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.

Overpowered owner

The token contract owner has the following powers:

1. The owner can change **ContractAddressLocator** contract by calling `upgrade()` function of **ContractAddressLocatorProxy** contract. By doing this, the owner can change certain parts of the system.

   Comment from the developers: "This is Intentional. We deal with the ability to update the smart contract as part of our Governance model."

2. The owner can change rights of existing users by calling `upsertOne()` and `upsertAll()` functions of **AuthorizationDataSource** contract. For example, the owner can freeze funds of any user.

   Comment from the developers: "This is intentional and needed in order to comply with AML regulations."

3. The owner can limit a user's transfers by calling `setLimit()` function of **TradingClasses** contract. As a result, the user's funds can be frozen.

   Comment from the developers: "This is intentional and needed in order to comply with AML regulations."

In the current implementation, the system depends heavily on the owner of the contract. In this case, there are scenarios that may lead to undesirable consequences for investors, e.g. if the owner's private keys become compromised. Thus, we recommend designing contracts in a trustless manner.

Comment from the developers: ”To mitigate the risk of compromised private keys, ownership will be transferred from the deployer to a MultiSig contract immediately after deployment.”

Low severity issues

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

revert() vs require() (fixed)

`revert()` is used in the following parts of the code:

1. **SGNToken.sol**, line 91:

```
revert("custodian-transfer of SGN into this contract is
illegal");
```

2. **SGAToken.sol**, line 91:

```
revert("custodian-transfer of SGA into this contract is
illegal");
```

We recommend using `require(condition);` instead of `if (!condition) revert();` to improve code readability and transparency.

The issues have been fixed and are not present in the latest version of the code.

Gas limit and loops

The following loops traverse through arrays of variable length:

1. **AuthorizationDataSource.sol**, line 80:

```
for (uint256 i = 0; i < _wallets.length; i++)
```

2. **AuthorizationDataSource.sol**, line 89:

```
for (uint256 i = 0; i < _wallets.length; i++)
```

3. **TradingDataSource.sol**, line 41:

```
for (uint256 i = 0; i < _wallets.length; i++)
```

The traversed arrays are passed as parameters of the functions. Therefore, if there are too many items in these arrays, the execution of the corresponding functions will fail due to an out-of-gas exception.

In these cases, we recommend separating the calls into several transactions.

Comment from the developers: ”These functions are designated for reducing gas-cost and will be called with an input that fits current gas limits.”

Redundant code (fixed)

The following variables are not used and thus redundant:

- **SGNTokenManager.sol**, line 74, `_value` variable:

```
function uponTransfer(address _sender, address _to, uint256
_value) external only(_ISGNToken_) {
```

- **SGNTokenManager.sol**, line 86, `_value` variable:

```
function uponTransferFrom(address _sender, address _from,
address _to, uint256 _value) external only(_ISGNToken_) {
```

- **SGAToken.sol**, line 102, `wallet`, `amount` variables:

```
(address wallet, uint256 amount) =
getSGATokenManager().uponDeposit(msg.sender,
address(this).balance - msg.value);
```

We recommend removing redundant code in order to improve code readability and transparency.

The issues have been fixed and are not present in the latest version of the code.

Notes

Gas cost

The number of storage slots required for **WalletInfo** (**AuthorizationDataSource.sol**) and **Timestamp** (**TimeManager.sol**) structures can be reduced in the following way:

- **WalletInfo**

```
struct WalletInfo {
    uint208 sequenceNum;
    bool isAuthorized;
    uint32 actionRole;
    uint256 tradeLimit;
    uint8 tradeClass;
}
```

- **Timestamp**

```
struct Timestamp {
    bool valid;
    uint248 value;
}
```

Comment from the developers: ”In order to reduce any slightest chance for casting errors which will lead to overflow, we prefer to stay on the safe side and use ‘uint256‘ conclusively (for any ‘uint‘ in the code). Gas optimization here is negligible.”

This analysis was performed by SmartDec.

Alexander Seleznev, Chief Business Development Officer
Boris Nikashin, Project Manager
Alexander Drygin, Analyst
Pavel Kondratenkov, Analyst

August 23, 2019

Appendix

Code coverage

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
authorization/	100	100	100	100	
AuthorizationActionRoles.sol	100	100	100	100	
AuthorizationDataSource.sol	100	100	100	100	
authorization/helpers/	100	100	100	100	
AuthorizationDataSourceMockup.sol	100	100	100	100	
authorization/interfaces/	100	100	100	100	
IAuthorizationDataSource.sol	100	100	100	100	
registry/	100	100	100	100	
ContractAddressLocator.sol	100	100	100	100	
ContractAddressLocatorHolder.sol	100	100	100	100	
ContractAddressLocatorProxy.sol	100	100	100	100	
registry/helpers/	100	100	100	100	
ContractAddressLocatorHolderExposure.sol	100	100	100	100	
ContractAddressLocatorMockup.sol	100	100	100	100	
ContractAddressLocatorProxyMockup.sol	100	100	100	100	
registry/interfaces/	100	100	100	100	
IContractAddressLocator.sol	100	100	100	100	
saga-genesis/	100	100	100	100	
ConversionManager.sol	100	100	100	100	
SGNAuthorizationManager.sol	100	100	100	100	
SGNToken.sol	100	100	100	100	
SGNTokenManager.sol	100	100	100	100	
saga-genesis/helpers/	100	100	100	100	
ConversionManagerMockup.sol	100	100	100	100	
MintManagerMockup.sol	100	100	100	100	
SGNAuthorizationManagerMockup.sol	100	100	100	100	
SGNTokenManagerMockup.sol	100	100	100	100	
SagaExchangerMockup.sol	100	100	100	100	
saga-genesis/interfaces/	100	100	100	100	
IConversionManager.sol	100	100	100	100	
IMintHandler.sol	100	100	100	100	
IMintManager.sol	100	100	100	100	
ISGNAuthorizationManager.sol	100	100	100	100	
ISGNTokenManager.sol	100	100	100	100	
ISagaExchanger.sol	100	100	100	100	
saga/	100	100	100	100	
DataSource.sol	100	100	100	100	
DebtManager.sol	100	100	100	100	
DebtQueue.sol	100	100	100	100	
InterestConverter.sol	100	100	100	100	
IntervalIterator.sol	100	100	100	100	
MintManager.sol	100	100	100	100	
PriceBandManager.sol	100	100	100	100	
PriceCalculator.sol	100	100	100	100	
RedButton.sol	100	100	100	100	
ReserveManager.sol	100	100	100	100	
SGAAuthorizationManager.sol	100	100	100	100	
SGAToken.sol	100	100	100	100	
SGATokenManager.sol	100	100	100	100	
SagaModel.sol	100	100	100	100	
SagaModelState.sol	100	100	100	100	
TimeManager.sol	100	100	100	100	
TradingClasses.sol	100	100	100	100	
TradingConverter.sol	100	100	100	100	
TradingDataSource.sol	100	100	100	100	
TradingManager.sol	100	100	100	100	
TransactionConverter.sol	100	100	100	100	
TransactionLimiter.sol	100	100	100	100	
TransactionManager.sol	100	100	100	100	
saga/helpers/	100	100	100	100	
DataSourceMockup.sol	100	100	100	100	
DebtManagerMockup.sol	100	100	100	100	
DebtManagerUser.sol	100	100	100	100	
DebtQueueMockup.sol	100	100	100	100	
InterestConverterMockup.sol	100	100	100	100	
IntervalIteratorExposure.sol	100	100	100	100	
IntervalIteratorMockup.sol	100	100	100	100	
MintHandlerMockup.sol	100	100	100	100	
MintListenerMockup.sol	100	100	100	100	
MintManagerExposure.sol	100	100	100	100	
PriceBandManagerMockup.sol	100	100	100	100	
PriceCalculatorExposure.sol	100	100	100	100	
PriceCalculatorMockup.sol	100	100	100	100	
RedButtonMockup.sol	100	100	100	100	
ReserveManagerMockup.sol	100	100	100	100	
SGAAuthorizationManagerMockup.sol	100	100	100	100	
SGATokenManagerMockup.sol	100	100	100	100	
SagaModelMockup.sol	100	100	100	100	
SagaModelStateMockup.sol	100	100	100	100	

```
  TimeManagerExposure.sol                  |     100 |      100 |     100 |     100 |                |
  TimeManagerMockup.sol                    |     100 |      100 |     100 |     100 |                |
  TradingClassesMockup.sol                 |     100 |      100 |     100 |     100 |                |
  TradingConverterMockup.sol               |     100 |      100 |     100 |     100 |                |
  TradingDataSourceMockup.sol              |     100 |      100 |     100 |     100 |                |
  TradingManagerMockup.sol                 |     100 |      100 |     100 |     100 |                |
  TransactionConverterMockup.sol           |     100 |      100 |     100 |     100 |                |
  TransactionLimiterMockup.sol             |     100 |      100 |     100 |     100 |                |
  TransactionManagerMockup.sol             |     100 |      100 |     100 |     100 |                |
 saga/interfaces/                          |     100 |      100 |     100 |     100 |                |
  IDataSource.sol                          |     100 |      100 |     100 |     100 |                |
  IDebtHandler.sol                         |     100 |      100 |     100 |     100 |                |
  IDebtManager.sol                         |     100 |      100 |     100 |     100 |                |
  IDebtQueue.sol                           |     100 |      100 |     100 |     100 |                |
  IInterestConverter.sol                   |     100 |      100 |     100 |     100 |                |
  IIntervalIterator.sol                    |     100 |      100 |     100 |     100 |                |
  IMintListener.sol                        |     100 |      100 |     100 |     100 |                |
  IPriceBandManager.sol                    |     100 |      100 |     100 |     100 |                |
  IPriceCalculator.sol                     |     100 |      100 |     100 |     100 |                |
  IRedButton.sol                           |     100 |      100 |     100 |     100 |                |
  IReserveManager.sol                      |     100 |      100 |     100 |     100 |                |
  ISGAAuthorizationManager.sol             |     100 |      100 |     100 |     100 |                |
  ISGATokenManager.sol                     |     100 |      100 |     100 |     100 |                |
  ISagaModel.sol                           |     100 |      100 |     100 |     100 |                |
  ISagaModelState.sol                      |     100 |      100 |     100 |     100 |                |
  ITimeManager.sol                         |     100 |      100 |     100 |     100 |                |
  ITradingClasses.sol                      |     100 |      100 |     100 |     100 |                |
  ITradingConverter.sol                    |     100 |      100 |     100 |     100 |                |
  ITradingDataSource.sol                   |     100 |      100 |     100 |     100 |                |
  ITradingManager.sol                      |     100 |      100 |     100 |     100 |                |
  ITransactionConverter.sol                |     100 |      100 |     100 |     100 |                |
  ITransactionLimiter.sol                  |     100 |      100 |     100 |     100 |                |
  ITransactionManager.sol                  |     100 |      100 |     100 |     100 |                |
 utils/                                    |   99.03 |    94.44 |     100 |   99.18 |                |
  Adminable.sol                            |     100 |      100 |     100 |     100 |                |
  MultiSigWallet.sol                       |   98.84 |    93.18 |     100 |   99.02 |            150 |
 utils/helpers/                            |     100 |      100 |     100 |     100 |                |
  MultiSigWalletExposure.sol               |     100 |      100 |     100 |     100 |                |
  MultiSigWalletTestCalls.sol              |     100 |      100 |     100 |     100 |                |
  MultiSigWalletTestToken.sol              |     100 |      100 |     100 |     100 |                |
-------------------------------------------|---------|----------|---------|---------|----------------|
All files                                  |   99.89 |     98.9 |     100 |   99.89 |                |
-------------------------------------------|---------|----------|---------|---------|----------------|
```

Tests output

```
  50740 passing (2h)
```

Compilation output

```
Compiling ./contracts/Migrations.sol...
Compiling ./contracts/authorization/AuthorizationActionRoles
.sol...
Compiling ./contracts/authorization/AuthorizationDataSource.
sol...
Compiling ./contracts/authorization/helpers/AuthorizationDat
aSourceMockup.sol...
Compiling ./contracts/authorization/interfaces/IAuthorizatio
nDataSource.sol...
Compiling ./contracts/registry/ContractAddressLocator.sol...
Compiling ./contracts/registry/ContractAddressLocatorHolder.
sol...
```

```
Compiling ./contracts/registry/ContractAddressLocatorProxy.s
ol...
Compiling ./contracts/registry/helpers/ContractAddressLocato
rHolderExposure.sol...
Compiling ./contracts/registry/helpers/ContractAddressLocato
rMockup.sol...
Compiling ./contracts/registry/helpers/ContractAddressLocato
rProxyMockup.sol...
Compiling ./contracts/registry/interfaces/IContractAddressLo
cator.sol...
Compiling ./contracts/saga-genesis/ConversionManager.sol...
Compiling ./contracts/saga-genesis/SGNAuthorizationManager.s
ol...
Compiling ./contracts/saga-genesis/SGNToken.sol...
Compiling ./contracts/saga-genesis/SGNTokenManager.sol...
Compiling ./contracts/saga-genesis/helpers/ConversionManager
Mockup.sol...
Compiling ./contracts/saga-genesis/helpers/MintManagerMockup
.sol...
Compiling ./contracts/saga-genesis/helpers/SGNAuthorizationM
anagerMockup.sol...
Compiling ./contracts/saga-genesis/helpers/SGNTokenManagerMo
ckup.sol...
Compiling ./contracts/saga-genesis/helpers/SagaExchangerMock
up.sol...
Compiling ./contracts/saga-genesis/interfaces/IConversionMan
ager.sol...
Compiling ./contracts/saga-genesis/interfaces/IMintHandler.s
ol...
Compiling ./contracts/saga-genesis/interfaces/IMintManager.s
ol...
Compiling ./contracts/saga-genesis/interfaces/ISGNAuthorizat
ionManager.sol...
Compiling ./contracts/saga-genesis/interfaces/ISGNTokenManag
er.sol...
Compiling ./contracts/saga-genesis/interfaces/ISagaExchanger
.sol...
Compiling ./contracts/saga/DataSource.sol...
Compiling ./contracts/saga/DebtManager.sol...
Compiling ./contracts/saga/DebtQueue.sol...
Compiling ./contracts/saga/InterestConverter.sol...
Compiling ./contracts/saga/IntervalIterator.sol...
Compiling ./contracts/saga/MintManager.sol...
Compiling ./contracts/saga/PriceBandManager.sol...
```

```
Compiling ./contracts/saga/PriceCalculator.sol...
Compiling ./contracts/saga/RedButton.sol...
Compiling ./contracts/saga/ReserveManager.sol...
Compiling ./contracts/saga/SGAAuthorizationManager.sol...
Compiling ./contracts/saga/SGAToken.sol...
Compiling ./contracts/saga/SGATokenManager.sol...
Compiling ./contracts/saga/SagaModel.sol...
Compiling ./contracts/saga/SagaModelState.sol...
Compiling ./contracts/saga/TimeManager.sol...
Compiling ./contracts/saga/TradingClasses.sol...
Compiling ./contracts/saga/TradingConverter.sol...
Compiling ./contracts/saga/TradingDataSource.sol...
Compiling ./contracts/saga/TradingManager.sol...
Compiling ./contracts/saga/TransactionConverter.sol...
Compiling ./contracts/saga/TransactionLimiter.sol...
Compiling ./contracts/saga/TransactionManager.sol...
Compiling ./contracts/saga/helpers/DataSourceMockup.sol...
Compiling ./contracts/saga/helpers/DebtManagerMockup.sol...
Compiling ./contracts/saga/helpers/DebtManagerUser.sol...
Compiling ./contracts/saga/helpers/DebtQueueMockup.sol...
Compiling ./contracts/saga/helpers/InterestConverterMockup.s
ol...
Compiling ./contracts/saga/helpers/IntervalIteratorExposure.
sol...
Compiling ./contracts/saga/helpers/IntervalIteratorMockup.so
l...
Compiling ./contracts/saga/helpers/MintHandlerMockup.sol...
Compiling ./contracts/saga/helpers/MintListenerMockup.sol...
Compiling ./contracts/saga/helpers/MintManagerExposure.sol..
.
Compiling ./contracts/saga/helpers/PriceBandManagerMockup.so
l...
Compiling ./contracts/saga/helpers/PriceCalculatorExposure.s
ol...
Compiling ./contracts/saga/helpers/PriceCalculatorMockup.sol
...
Compiling ./contracts/saga/helpers/RedButtonMockup.sol...
Compiling ./contracts/saga/helpers/ReserveManagerMockup.sol.
..
Compiling ./contracts/saga/helpers/SGAAuthorizationManagerMo
ckup.sol...
Compiling ./contracts/saga/helpers/SGATokenManagerMockup.sol
...
Compiling ./contracts/saga/helpers/SagaModelMockup.sol...
```

```
Compiling ./contracts/saga/helpers/SagaModelStateMockup.sol.
..
Compiling ./contracts/saga/helpers/TimeManagerExposure.sol..
.
Compiling ./contracts/saga/helpers/TimeManagerMockup.sol...
Compiling ./contracts/saga/helpers/TradingClassesMockup.sol.
..
Compiling ./contracts/saga/helpers/TradingConverterMockup.so
l...
Compiling ./contracts/saga/helpers/TradingDataSourceMockup.s
ol...
Compiling ./contracts/saga/helpers/TradingManagerMockup.sol.
..
Compiling ./contracts/saga/helpers/TransactionConverterMocku
p.sol...
Compiling ./contracts/saga/helpers/TransactionLimiterMockup.
sol...
Compiling ./contracts/saga/helpers/TransactionManagerMockup.
sol...
Compiling ./contracts/saga/interfaces/IDataSource.sol...
Compiling ./contracts/saga/interfaces/IDebtHandler.sol...
Compiling ./contracts/saga/interfaces/IDebtManager.sol...
Compiling ./contracts/saga/interfaces/IDebtQueue.sol...
Compiling ./contracts/saga/interfaces/IInterestConverter.sol
...
Compiling ./contracts/saga/interfaces/IIntervalIterator.sol.
..
Compiling ./contracts/saga/interfaces/IMintListener.sol...
Compiling ./contracts/saga/interfaces/IPriceBandManager.sol.
..
Compiling ./contracts/saga/interfaces/IPriceCalculator.sol..
.
Compiling ./contracts/saga/interfaces/IRedButton.sol...
Compiling ./contracts/saga/interfaces/IReserveManager.sol...
Compiling ./contracts/saga/interfaces/ISGAAuthorizationManag
er.sol...
Compiling ./contracts/saga/interfaces/ISGATokenManager.sol..
.
Compiling ./contracts/saga/interfaces/ISagaModel.sol...
Compiling ./contracts/saga/interfaces/ISagaModelState.sol...
Compiling ./contracts/saga/interfaces/ITimeManager.sol...
Compiling ./contracts/saga/interfaces/ITradingClasses.sol...
Compiling ./contracts/saga/interfaces/ITradingConverter.sol.
..
```

```
Compiling ./contracts/saga/interfaces/ITradingDataSource.sol
...
Compiling ./contracts/saga/interfaces/ITradingManager.sol...
Compiling ./contracts/saga/interfaces/ITransactionConverter.
sol...
Compiling ./contracts/saga/interfaces/ITransactionLimiter.so
l...
Compiling ./contracts/saga/interfaces/ITransactionManager.so
l...
Compiling ./contracts/utils/Adminable.sol...
Compiling ./contracts/utils/MultiSigWallet.sol...
Compiling ./contracts/utils/helpers/MultiSigWalletExposure.s
ol...
Compiling ./contracts/utils/helpers/MultiSigWalletTestCalls.
sol...
Compiling ./contracts/utils/helpers/MultiSigWalletTestToken.
sol...
Compiling openzeppelin-solidity-v1.12.0/contracts/ownership/
Claimable.sol...
Compiling openzeppelin-solidity-v1.12.0/contracts/ownership/
Ownable.sol...
Compiling openzeppelin-solidity/contracts/math/SafeMath.sol.
..
Compiling openzeppelin-solidity/contracts/token/ERC20/ERC20.
sol...
Compiling openzeppelin-solidity/contracts/token/ERC20/IERC20
.sol...
Writing artifacts to ./build/contracts
```

Solhint output

```
774 problems (733 errors, 41 warnings)
```

Solium output

```
55 errors, 260 warnings found.
```