BLAZE

SMART CONTRACT SECURITY REVIEW

JURY.ONLINE SMART CONTRACT

CLIENT
JURY.ONLINE

DATE
03/29/2018

WWW.BLAZEINFOSEC.COM

SUMMARY

1.0 DOCUMENT CONTROL

1.1 DOCUMENT CONTROL

Authors	Delivery Date	Pages	Version	Status
Victor Farias and Julio Fort	16/03/2018	10	0.9	Draft for Q&A
Victor Farias and Julio Fort	18/03/2018	22	1.0	Final report
Victor Farias and Julio Fort	29/03/2018	22	1.1	Final report – fixes added

1.2 DOCUMENT DISTRIBUTION

Name	Title	Organisation
Julio Fort	Director of Professional Services	Blaze Information Security
Alexander Shevtsov	Founder	Jury.Online

2.0 INTRODUCTION

DISCLAIMER: **This document presents the findings of a security review of the smart contracts under scope of the audit. As a time-boxed and best effort exercise, it does not guarantee there are no other security issues in the smart contract. The results of this audit should not be read as an investment advice.**

Jury.Online aims to create a platform to facilitate deals between different parties. The platform puts itself in the middle of a deal and works as a escrow service; it mediates a given transaction and if all parties are satisfied with the outcome of the deal, the transaction is completed successfully. As a escrow service, it also mediates disputes between the parties of a deal. Jury.Online provides interaction between judges, arbitrators and parties of a deal for dispute resolutions.

Code patterns non-compliant with the Ethereum token standard or to the contract specification, deviation of best practices and vulnerability discovered during the assessment, Blaze Information Security attributed a risk severity rating and, whenever possible, validated the existence of the vulnerability with a working exploit code.

For each vulnerability discovered during the assessment, Blaze Information Security attributed a risk severity rating and, whenever possible, validated the existence of the vulnerability with a working exploit code.

The main objectives of the assessment were the following:

- Identify the main security-related issues present in the smart contract
- Assess the level of secure coding practices present in the project
- Obtain evidences for each vulnerability and, if possible, develop a working exploit
- Document, in a clear and easy to reproduce manner, all procedures used to replicate the issue
- Recommend mitigation factors and fixes for each defect identified in the analysis
- Provide context with a real risk scenario based on a realistic threat model

3.0 EXECUTIVE SUMMARY

The engagement was performed in a total period of 6 business days, including report writing. The smart contract security review commenced part-time on 05/03/2018 and ended on 16/03/2018, finishing with the preliminary version of this report.

On 23/03/2018 all findings reported by Blaze Information Security were fixed accordingly by Jury.Online. The issues are no longer present in the code of the contracts and were fixed in the commit 3f5f707cfeec36e174702b46be0c8f6850e6a12b.

The audit was done with the assistance of automated tools as well as subjected to manual review. The generated EVM code was not inspected in this assessment.

There was only one minor issue discovered in the contracts audited in this engagement. This issue was believed to not bring an immediate risk to the contracts, but should be taken as an advice to improve its security and make it future-proof. The review of the the contracts under scope did not reveal vulnerabilities that could lead to loss of tokens, bias of jurors, nor problems that had the potential to cause a significant impact to the intended operations of Jury.Online.

Jury.Online had defensive security coding patterns and followed many recommended Solidity programming good practices. Overall the code quality was considered very good, as it was clear, well commented and easy to understand.

The following table sumarizes the issues found in the smart contracts under scope for this audit.

POINT	TITLE	SEVERITY
1	Absence of arithmetic underflow and overflow checks in parts of the contract	LOW

4.0 SCOPE

The scope of this security review is comprised of smart contracts written in Solidity.

- Project name: **juryonline**
- Commit: **986aca6ca9c666a34632e4e0ed10d2c78d1fa245**

Filename	Lines of code
ERC20Token.sol	237
JuryOnlineExchanger.sol	28
JuryOnlineICOContract.sol	204
JuryOnlineInvestContract.sol	226
Migrations.sol	23
Pullable.sol	38

The code audited is open source and can be found at https://github.com/juryonline/contracts/tree/playground (Playground branch)

5.0 METHODOLOGY - SMART CONTRACT SECURITY REVIEW

Our security-oriented smart contract review follows an organized methodology with the intent to identify the largest number of vulnerabilities in the contracts under scope from the perspective of a motivated, technically capable and persistent adversary.

Special attention is directed towards critical areas of the smart contract such as burning of tokens and functioning of the multi-signature. Our process also looks into other common implementation issues that lead to problems like reentrancy, mathematical overflows and underflows, gas-related denial of service, etc.

Blaze's smart contract review methodology involves automated and manual audit techniques.

The applications are subjected to a round of dynamic analysis using tools like linters, program profilers and source code security scanners.

The contracts have their source code manually inspected for security flaws. This type of analysis has the ability to detect issues that are missed by automated scanners and static analyzers, as it can discover edge-cases and business logic-related problems.

6.0 TECHNICAL SUMMARY

6.1 DESCRIPTION OF THE SMART CONTRACTS

- ERC20Token.sol

Contract with the ERC20 standard token, modified with security enhancements such as *SafeMath* and *approve_fixed*, the latter created to prevent a well-known ERC20 race condition that may cause double withdraw.

- JuryOnlineICOContract.sol

Responsible for fund raising. This contract defines funding goals for each milestone and total effort to be spent on and duration of the project.

- JuryOnlineInvestContract.sol

This contract is responsible for managing potential dispute of interests among parties. In this contract an investor can open a dispute case against a developer about a project milestone, for example, and it will be voted by the jurors to decide whether or not to allocate resources and funds for the continuation of the project.

- Pullable.sol

This contract has auxiliary methods used in **InvestContract** to make asynchronous transfers.

7.0 VULNERABILITIES

7.1 ABSENCE OF ARITHMETIC UNDERFLOW AND OVERFLOW CHECKS IN PARTS OF THE CONTRACT

SEVERITY: LOW	CVSS SCORE:
AFFECTED POINTS	JuryOnlineICOContract.sol and JuryOnlineInvestContract.sol

DESCRIPTION

Fixed	Comments
Yes	Fixed in commit 3f5f707cfeec36e174702b46be0c8f6850e6a12b

During the audit it was observed that the contract implement a series of measures regarding to mathematical operations to prevent and effectively mitigate arithmetic underflow and overflows of *uint32* variables.

However, Blaze Information Security noticed some parts of the contract as in **JuryOnlineICOContract.sol** and **JuryOnlineInvestContract.sol** did not apply these countermeasures, making the variables and functions that perform mathematical operations in these contracts potentially susceptible to this kind of attack.

The code below illustrates the absence of SafeMath or other functions and libraries to prevent arithmetic overflows and underflows:

```
    /// @dev Adds a milestone.
    /// @param _etherAmount amount of Ether needed for the added milestone
    /// @param _tokenAmount amount of tokens which will be released for added milestone
    /// @param _startTime field for start timestamp of added milestone
    /// @param _duration assumed duration of the milestone
    /// @param _description description of added milestone
    function addMilestone(uint _etherAmount, uint _tokenAmount, uint _startTime, uint _duration, string _description) public
notSealed only(operator) returns(uint) {
        totalEther += _etherAmount;
        totalToken += _tokenAmount;
        return milestones.push(Milestone(_etherAmount, _tokenAmount, _startTime, 0, _duration, _description, ""));
    }
```

Despite no evidence of a viable exploitation scenario using underflow or overflow on those contracts were found during this assessment, it is important to implement the mitigation in advance in order to turn this kind of attack impossible, even if they are somehow discovered in the code or be triggered by corner cases.

REFERENCE

- https://ethereumdev.io/safemath-protect-overflows
- https://openzeppelin.org/api/docs/math_SafeMath.html

SOLUTION

As a general good practice and be overall consistent with the same security countermeasures already present in the project, it is recommend to apply the same mitigation implemented in other parts of the contract to the aforementioned Solidity files.

Consider using OpenZeppelin's SafeMath, as it is the most popular library with enhanced security checks for safe mathematical operations.

It is understood, however, that adding those arithmetical safeguards to the contract may increase its gas usage.

8.0 CONCLUSION

The ultimate goal of a security assessment is to bring the opportunity to better illustrate the risk of an organization and help make it understand and validate its security posture against potential threats to its business.

With that in mind, Blaze Information Security provides the following recommendations that we believe should be adopted as next steps to further enhance the security posture of the smart contracts:

- Fix the only outstanding issue presented in the report, taking into consideration future development of the project;
- Engage another third party IT security provider for a second round of audit;
- Consider establishing a bug bounty program, as it is becoming increasingly common among companies in the smart contract and blockchain field.

Blaze Information Security would like to thank the team of Jury.Online for their support and assistance during the entire engagement.

9.0 APPENDIX A - VULNERABILITY CRITERIA CLASSIFICATION

Below is the risk rating criteria used to classify the vulnerabilities discussed in this report:

Severity	Description
CRITICAL	Leads to the compromise of the system and the data it handles. Can be exploited by an unskilled attacker using publicly available tools and exploits. Must be addressed immediately.
HIGH	Usually leads to the compromise of the system and the data it handles.
MEDIUM	Does not lead to the immediate compromise of the system but when chained with other issues can bring serious security risks. Nevertheless, it is advisable to fix them accordingly.
LOW	Do not pose an immediate risk and even when chained with other vulnerabilities are less likely to cause serious impact.

10.0 APPENDIX B - OUTPUT OF AUTOMATED SCANNERS

10.1 SOL-FUNCTION-PROFILER

/home/crashburn/ethereum/audits/jury.online/contracts/JuryOnlineExchanger.sol

Contract	Function	Visibility	Constant	Returns	Modifiers
JuryOnlineTokenExchanger	JuryOnlineTokenExchanger(address,address)	public	false		
JuryOnlineTokenExchanger	exchange(uint)	public	false		

/home/crashburn/ethereum/audits/jury.online/contracts/Migrations.sol

Contract	Function	Visibility	Constant	Returns	Modifiers
Migrations	Migrations()	public	false		
Migrations	setCompleted(uint)	public	false		restricted
Migrations	upgrade(address)	public	false		restricted

/home/crashburn/ethereum/audits/jury.online/contracts/Pullable.sol

Contract	Function	Visibility	Constant	Returns	Modifiers
Pullable	withdrawPayment()	public	false		
Pullable	asyncSend(address,uint256)	internal	false		

/home/crashburn/ethereum/audits/jury.online/contracts/JuryOnlineICOContract.sol

Contract	Function	Visibility	Constant	Returns	Modifiers
ICOContract	ICOContract(address,address,uint,uint,uint)	public	false		
ICOContract	addMilestone(uint,uint,uint,uint,string,string)	public	false	uint	notSealed,only
ICOContract	editMilestone(uint,uint,uint,uint,uint,string,string)	public	false		notSealed,only
ICOContract	seal()	public	false		only,notSealed
ICOContract	finishMilestone(string)	public	false		only
ICOContract	startNextMilestone()	public	false		only
ICOContract	getCurrentMilestone()	public	true	uint	
ICOContract	milestonesLength()	public	false	uint	view
ICOContract	createInvestContract(address,uint,uint)	public	false	address	sealed,only
ICOContract	investContractDeposited()	public	false		
ICOContract	returnTokens()	public	false		only

/home/crashburn/ethereum/audits/jury.online/contracts/ERC20Token.sol

Contract	Function	Visibility	Constant	Returns	Modifiers
Owned	Owned()	public	false		
Owned	transferOwnership(address)	public	false		only
Owned	acceptOwnership()	public	false		only
ERC20	transfer(address,uint)	public	false	success	isStartedOnly
ERC20	transferFrom(address,address,uint)	public	false	success	isStartedOnly
ERC20	balanceOf(address)	public	true	balance	
ERC20	approve_fixed(address,uint,uint)	public	false	success	isStartedOnly
ERC20	approve(address,uint)	public	false	success	isStartedOnly
ERC20	allowance(address,address)	public	true	remaining	
Token	Token(string,string,uint8)	public	false		
Token	start()	public	false		only,isNotStartedOnly
Token	mint(address,uint)	public	false	bool	only,isNotStartedOnly
Token	multimint(address,uint)	public	false	uint	only,isNotStartedOnly
TokenWithoutStart	TokenWithoutStart(string,string,uint8)	public	false		
TokenWithoutStart	transfer(address,uint)	public	false	success	
TokenWithoutStart	transferFrom(address,address,uint)	public	false	success	
TokenWithoutStart	balanceOf(address)	public	true	balance	
TokenWithoutStart	approve_fixed(address,uint,uint)	public	false	success	
TokenWithoutStart	approve(address,uint)	public	false	success	
TokenWithoutStart	allowance(address,address)	public	true	remaining	
TokenWithoutStart	mint(address,uint)	public	false	bool	only
TokenWithoutStart	multimint(address,uint)	public	false	uint	only

10.2 OYENTE

```
WARNING:root:You are using an untested version of z3. 4.5.0 is
the officially tested version

WARNING:root:You are using evm version 1.8.1. The supported
version is 1.6.6

WARNING:root:You are using solc version 0.4.19, The latest
supported version is 0.4.17

INFO:root:Contract ERC20Token.sol:Base:

INFO:oyente.symExec:Running, please wait…

INFO:oyente.symExec:        ==== Results ===

INFO:oyente.symExec:          EVM code coverage:          100.0%

INFO:oyente.symExec:          Callstack bug:          False

INFO:oyente.symExec:          Money concurrency bug: False

INFO:oyente.symExec:          Time dependency bug:          False

INFO:oyente.symExec:          Reentrancy bug:          False

INFO:root:Contract ERC20Token.sol:ERC20:

INFO:oyente.symExec:Running, please wait…

INFO:oyente.symExec:        ==== Results ===

INFO:oyente.symExec:          EVM code coverage:          99.9%

INFO:oyente.symExec:          Callstack bug:          False

INFO:oyente.symExec:          Money concurrency bug: False

INFO:oyente.symExec:          Time dependency bug:          False

INFO:oyente.symExec:          Reentrancy bug:          False

INFO:root:Contract ERC20Token.sol:Owned:
```

```
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          99.6%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract ERC20Token.sol:SafeMath:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          100.0%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract ERC20Token.sol:Token:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          86.7%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
```

```
INFO:root:Contract ERC20Token.sol:TokenWithoutStart:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          86.1%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract
JuryOnlineExchanger.sol:JuryOnlineTokenExchanger:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          97.2%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
```

```
INFO:oyente.symExec:        == Analysis Completed ==
```

```
WARNING:root:You are using an untested version of z3. 4.5.0 is
the officially tested version

WARNING:root:You are using evm version 1.8.1. The supported
version is 1.6.6

WARNING:root:You are using solc version 0.4.19, The latest
supported version is 0.4.17

INFO:root:Contract ERC20Token.sol:Base:

INFO:oyente.symExec:Running, please wait…

INFO:oyente.symExec:        ==== Results ===

INFO:oyente.symExec:          EVM code coverage:          100.0%

INFO:oyente.symExec:          Callstack bug:          False

INFO:oyente.symExec:          Money concurrency bug: False

INFO:oyente.symExec:          Time dependency bug:          False

INFO:oyente.symExec:          Reentrancy bug:          False

INFO:root:Contract ERC20Token.sol:ERC20:

INFO:oyente.symExec:Running, please wait…

INFO:oyente.symExec:        ==== Results ===

INFO:oyente.symExec:          EVM code coverage:          99.9%

INFO:oyente.symExec:          Callstack bug:          False

INFO:oyente.symExec:          Money concurrency bug: False

INFO:oyente.symExec:          Time dependency bug:          False

INFO:oyente.symExec:          Reentrancy bug:          False

INFO:root:Contract ERC20Token.sol:Owned:
```

```
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          99.6%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract ERC20Token.sol:SafeMath:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          100.0%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract ERC20Token.sol:Token:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          86.7%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
```

```
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract ERC20Token.sol:TokenWithoutStart:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          86.1%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract JuryOnlineICOContract.sol:ICOContract:
incomplete push instruction at 17331
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          31.9%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract JuryOnlineInvestContract.sol:InvestContract:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          82.5%
INFO:oyente.symExec:          Callstack bug:          False
```

```
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract JuryOnlineInvestContract.sol:TokenPullable:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          97.4%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract Pullable.sol:Pullable:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          99.1%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
```

```
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:oyente.symExec:        == Analysis Completed ==
INFO:root:Contract JuryOnlineICOContract.sol:ICOContract:
incomplete push instruction at 17331
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          31.9%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract JuryOnlineInvestContract.sol:InvestContract:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          82.5%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
```

```
INFO:oyente.symExec:          Reentrancy bug:          False
```

```
INFO:root:Contract JuryOnlineInvestContract.sol:TokenPullable:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          97.4%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:root:Contract Pullable.sol:Pullable:
INFO:oyente.symExec:Running, please wait…
INFO:oyente.symExec:        ==== Results ===
INFO:oyente.symExec:          EVM code coverage:          99.1%
INFO:oyente.symExec:          Callstack bug:          False
INFO:oyente.symExec:          Money concurrency bug: False
INFO:oyente.symExec:          Time dependency bug:          False
INFO:oyente.symExec:          Reentrancy bug:          False
INFO:oyente.symExec:        == Analysis Completed ==
```

BLAZE
INFORMATION SECURITY

BRAZIL

+55 81 3071.7148
R. VISCONDE DE JEQUITINHONHA, 279,
EMPRESARIAL TANCREDO NEVES, ROOM
701, RECIFE, PERNAMBUCO

PORTUGAL

+351 22 120 1335
PRAÇA MOUZINHO DE ALBUQUERQUE,
113, 5TH FLOOR, PORTO

WWW.BLAZEINFOSEC.COM
INFO@BLAZEINFOSEC.COM