Smart Contract Code Review and Security Analysis Report

Customer: 5eth.io
Date: November 13, 2018

HACKEN

HACKEN CYBERSECURITY ECOSYSTEM.

This document contains confidential information about IT systems and intellectual property of the customer as well as information about potential vulnerabilities and methods of their exploitation.

This confidential information shall be used only internally by the customer and shall not be disclosed to third parties.

Document:

Name	Smart Contract Code Review and Security Analysis Report for 5eth.io
Platform	Ethereum / Solidity
Link	https://github.com/5ethio/Eth5io
Date	15.11.2018
Version	ef61a2406ad7aa78ce3ab61aa3c9cd1ab2203a75

Table of contents

Introduction

Hacken OÜ (Consultant) was contracted by 5eth.io (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of Customer`s smart contract and its code review conducted between November 11st, 2018 – November15th, 2018.

Scope

The scope of the project is 5eth.io smart contract, which can be found on github by link below:

https://github.com/5ethio/Eth5io

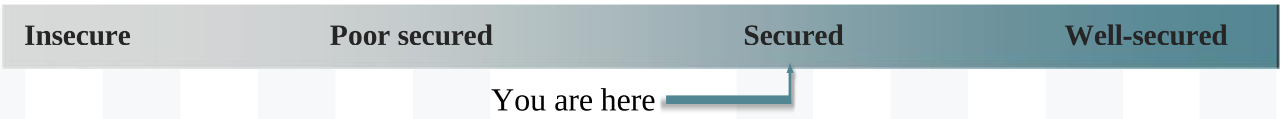
Commit version – ef61a2406ad7aa78ce3ab61aa3c9cd1ab2203a75

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are 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 Block Gas Limit
- Transaction-Ordering Dependence
- Byte array vulnerabilities
- Style guide violation
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Unchecked external call - Unchecked math
- Unsafe type inference
- Implicit visibility level

Executive Summary

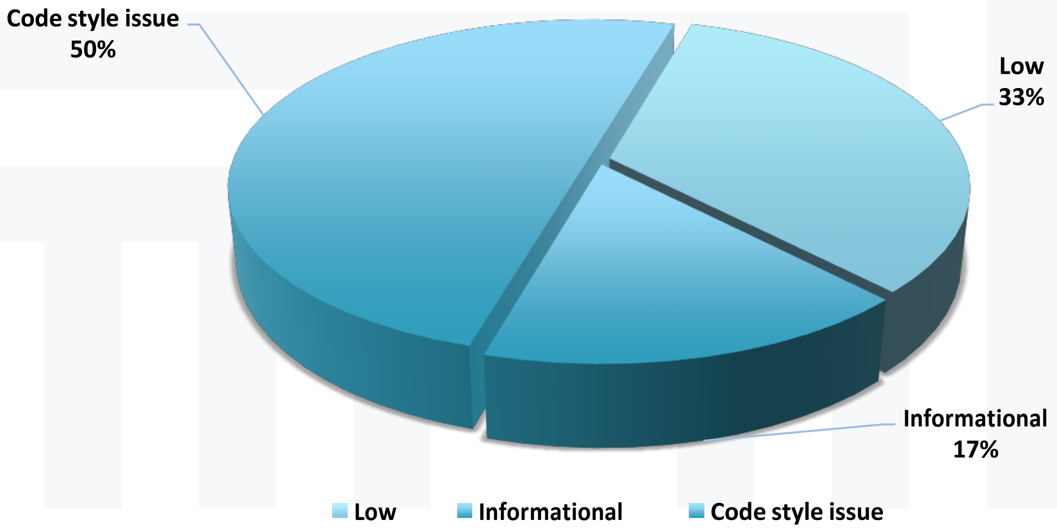
According to the assessment, Customer`s smart contracts are secure.



Our team performed analysis of code functionality, manual audit and automated checks with solc, Mythril, Slither and remix IDE (see Appendix B pic 1-13). All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in Audit overview section. General overview is presented in AS-IS section and all found issues can be found in Audit overview section.

We found 2 low vulnerabilities in smart contract; We also outline 1 informational statement and 3 code style issues, that can't have any security effect, but should be presented in the report.

Graph 1. The distribution of vulnerabilities.



Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens lose etc.
High	High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose
Low	Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution
Lowest / Code Style / Info	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored.

AS-IS overview

5eth.io contract overview

Eth5iov2 contract manages investment system for 1 year.

Eth5iov2 contract does not inherit other contracts. It does'n imports SafeMath, Percent, Zero, ToAddress libraries and InvestorsStorage contract.

Eth5iov2 contract constructor sets:

- admin to 0xb34a732Eb42A02ca5b72e79594fFfC10F55C33bd
- advertising to 0x63EA308eF23F3E098f8C1CE2D24A7b6141C55497
- roundStart to roundStart(resTriger)

Eth5iov2 has 4 modifiers:

- newDayDepositLimit – sets the daily fund limit.
- notOnPause – checks whether pause on next wave is not expired.

Eth5iov2 has 20 functions:

- getInvestorCount is a public view function – returns number of investors.
- investor is a public view function – returns all information from investor.
- newNumerator is a public view function – returns numerator denominator for percent.

- waiver is a private function – delete the contract owner.
- adminPercent is a public function – returns numerator and denominator for admin.
- addInvestorFrom_v1 is a public function – manual addition of investors with 0 balance.
- getInvestorDividendsAmount is a public view function – returns value devidebts for specified address.
- latestPayout is a public view function – returns latestTime for the payout.
- getMyDividends is a public function – sends dividends to the msg.sender. Has notOnPause, atPaymode and balanceChanged modifiers.
- doInvest is a public payable function – makes an investment. Has notOnPause and balanceChanged modifiers.
- payout is a public function – makes a payout for all addresses in investors storage. Has notOnPause, onlyAdmin, atPaymode and balanceChanged modifiers.
- setAdminAddr is a public function – sets admin address. Has onlyAdmin modifier.
- setPayerAddr is a public function – sets payer address. Has onlyAdmin modifier.

Audit overview

Critical

No critical vulnerabilities were found.

High

No high severity vulnerabilities were found.

Medium

No high severity vulnerabilities were found.

Low

1. doInvest function emits LogNewReferral only ones. It accepts 3 referral addresses, but doesn't log all of them (See Appendix A pic 1).
2. Compiler version is not locked. Consider locking the compiler version with latest one (See Appendix A pic. 2 for evidence).

```
pragma solidity ^0.4.25; // good: compiles w 0.4.25 only
```

Lowest / Code style / Info

Informational statements

Informational statements are audit team findings that doesn't have any security issues. However, they are presented in report to clarify and outline functionality and business requirements.

3. Percent contract doesn't use SafeMath library for math operations.

Code style issues

4. Accessibility contract:
   - Space should be deleted after coma on line 16
5. Eth5iov2 contract:
   - Space should be deleted after coma on line 284
   - Spaces should be added between expressions on line 208

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. For the contract high level description of functionality was presented in As-is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Overall quality of reviewed contracts is good; however, it contains 2 low vulnerabilities.

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed in accordance with industry practices at the date of this report, in relation to: cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

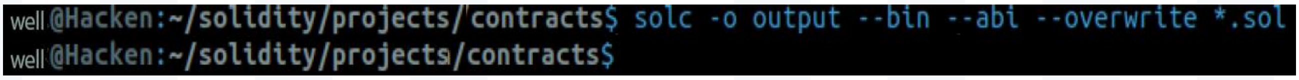
The audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only - we recommend proceeding with several independent audits and a public bug bounty program to ensure security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on blockchain platform. The platform, its programming language, and other software related to the smart contract can have own vulnerabilities that can lead to hacks. Thus, the audit can’t guarantee explicit security of the audited smart contracts

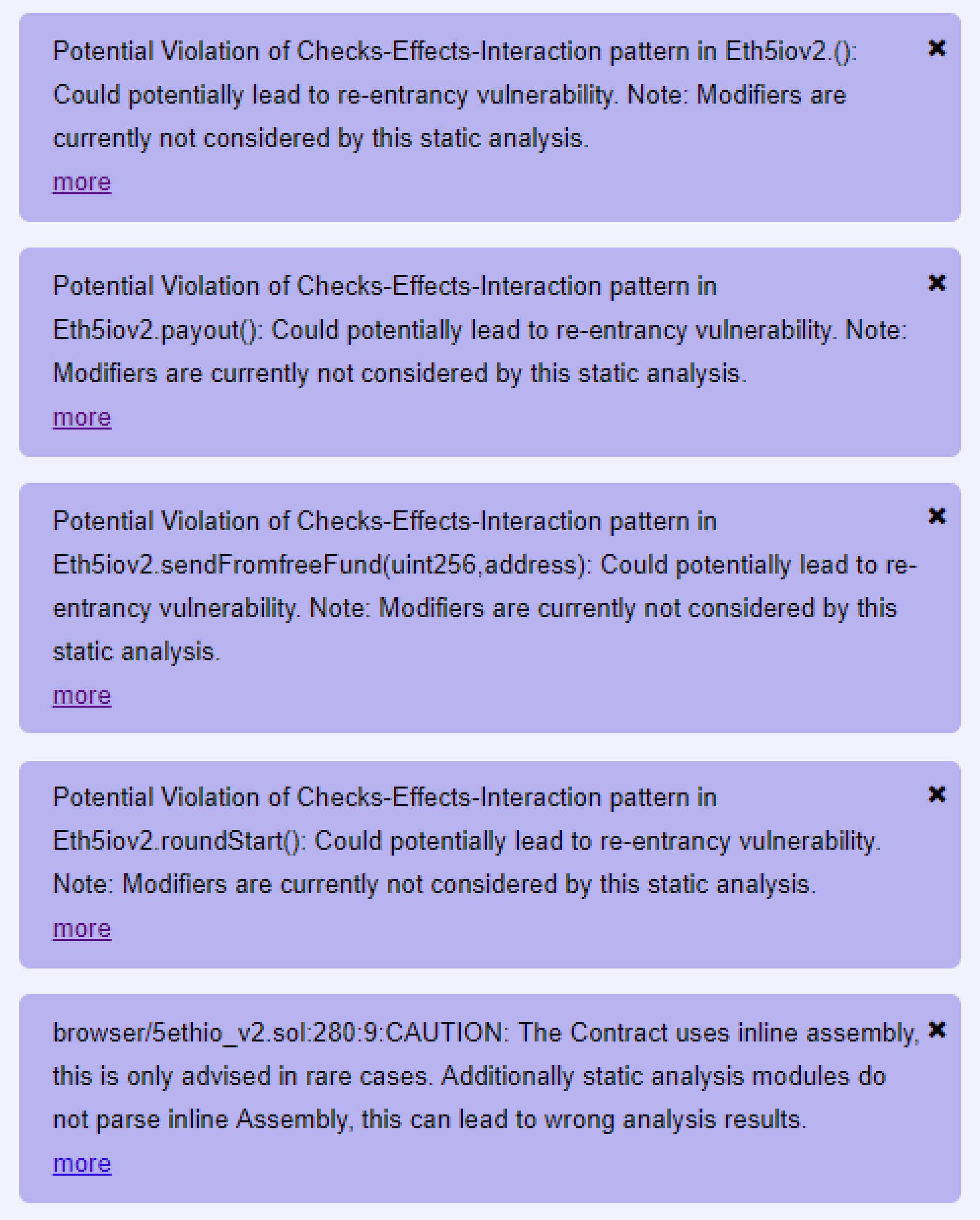
Appendix B. Automated tools reports

Pic 1. Solc automated report:



```
well @Hacken:~/solidity/projects/contracts$ solc -o output --bin --abi --overwrite *.sol
well @Hacken:~/solidity/projects/contracts$
```

Pic 2. Remix automated report Eth5iov2:



Potential Violation of Checks-Effects-Interaction pattern in Eth5iov2.(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more

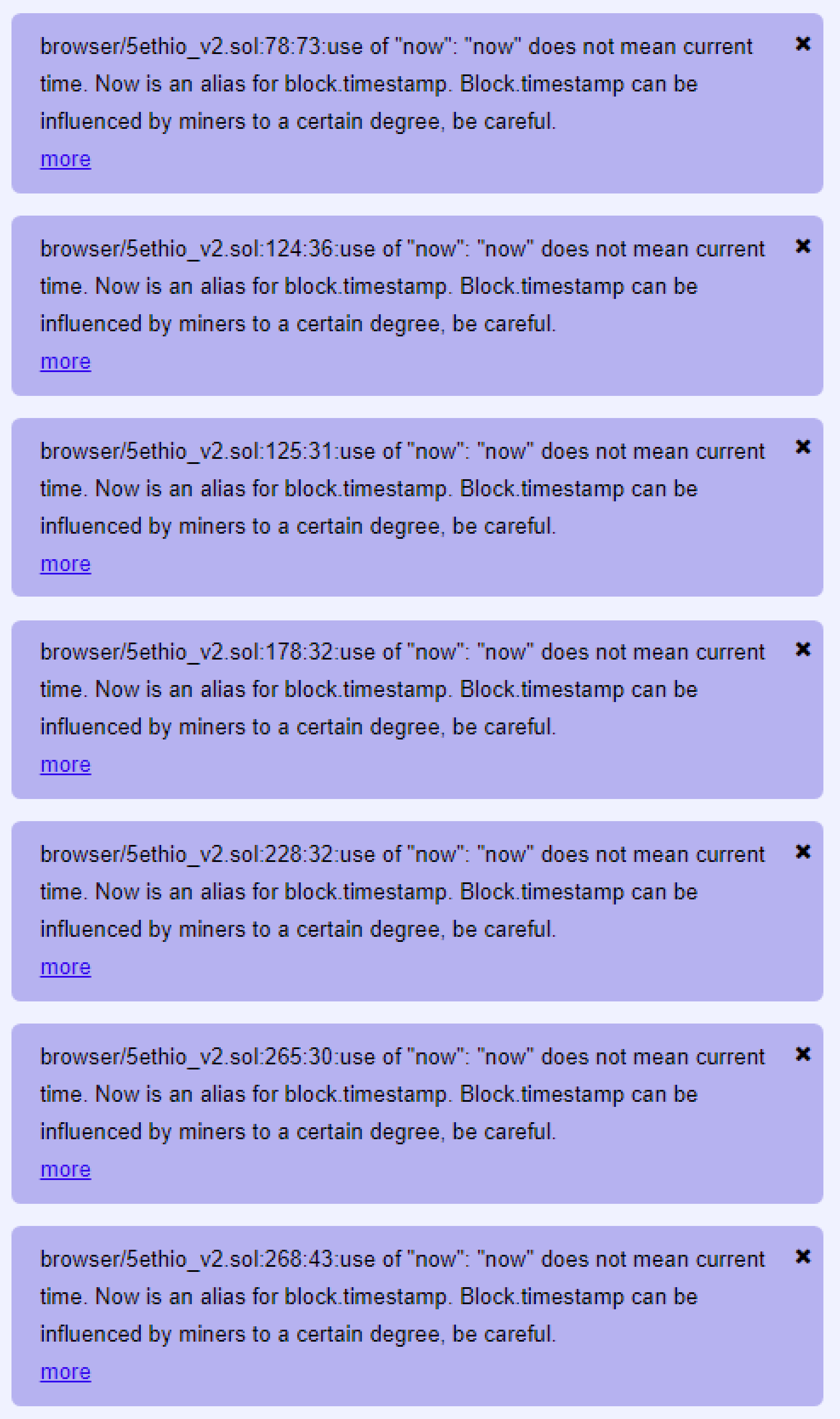
Potential Violation of Checks-Effects-Interaction pattern in Eth5iov2.payout(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more

Potential Violation of Checks-Effects-Interaction pattern in Eth5iov2.sendFromfreeFund(uint256,address): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more

Potential Violation of Checks-Effects-Interaction pattern in Eth5iov2.roundStart(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more

browser/5ethio_v2.sol:280:9:CAUTION: The Contract uses inline assembly, this is only advised in rare cases. Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.
more

Pic 3. Remix automated report Eth5iov2:



browser/5ethio_v2.sol:78:73:use of "now": "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.
more

browser/5ethio_v2.sol:124:36:use of "now": "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.
more

browser/5ethio_v2.sol:125:31:use of "now": "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.
more

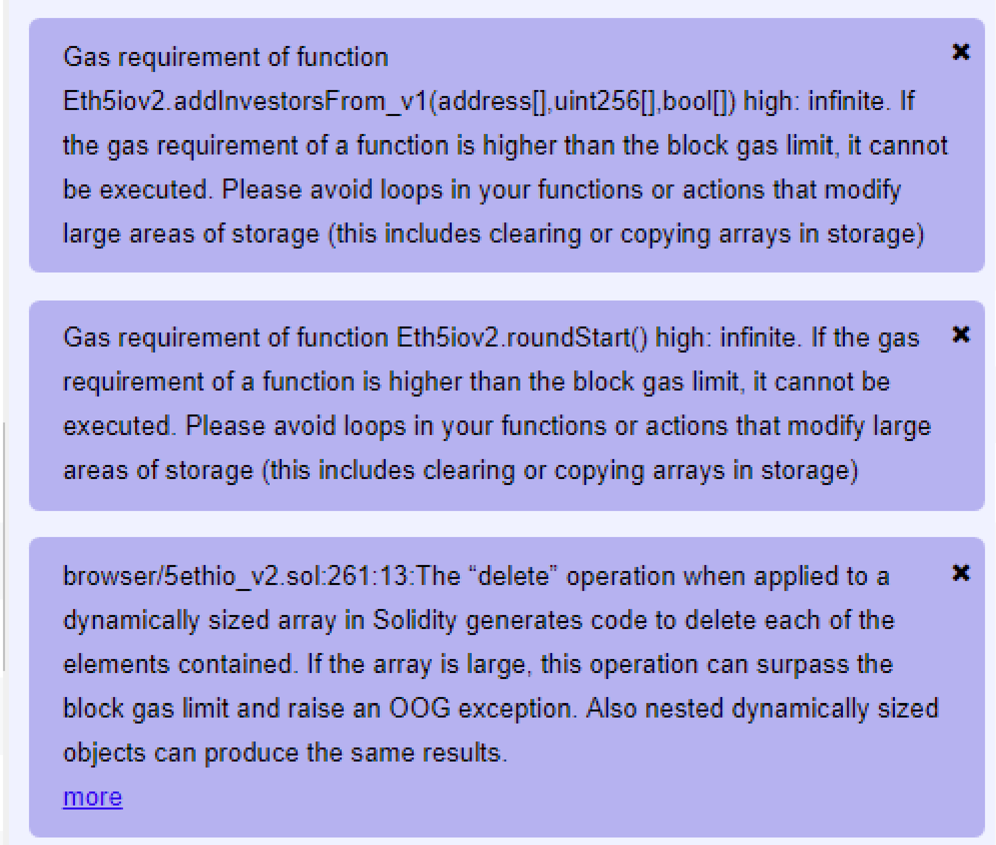
browser/5ethio_v2.sol:178:32:use of "now": "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.
more

browser/5ethio_v2.sol:228:32:use of "now": "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.
more

browser/5ethio_v2.sol:265:30:use of "now": "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.
more

browser/5ethio_v2.sol:268:43:use of "now": "now" does not mean current time. Now is an alias for block.timestamp. Block.timestamp can be influenced by miners to a certain degree, be careful.
more

Pic 4. Remix automated report Eth5iov2:



Gas requirement of function Eth5iov2.addInvestorsFrom_v1(address[],uint256[],bool[]) high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

Gas requirement of function Eth5iov2.roundStart() high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)

browser/5ethio_v2.sol:261:13:The “delete” operation when applied to a dynamically sized array in Solidity generates code to delete each of the elements contained. If the array is large, this operation can surpass the block gas limit and raise an OOG exception. Also nested dynamically sized objects can produce the same results.
more