CryptoSec

SECURITY FOR BLOCKCHAIN

REPORT
SMART CONTRACTS AUDIT RESULTS
FOR CRYPTOINDEX

CryptoSec

SECURITY FOR BLOCKCHAIN

Change history

Version	Date	Author	Changes
1.0	05.30.2018	CryptoSec	Report created
2.0	06.01.2018	CryptoSec	Report updated
3.0	06.05.2018	CryptoSec	Report updated

Confirmation

Name	Company	Position	Location	Email

Contacts

Company name	Tomahawk Technologies Inc.
Address	1801 Wedemeyer, San Francisco, 94129
PGP	CCFD 364F 0180 287E 4DD6 A0AB 6CDF F9EC 1BAE D618
Email	info@cryptosec.us
Web	https://www.cryptosec.us

CryptoSec
SECURITY FOR BLOCKCHAIN

Table of Contents

1 Limitations on disclosure and usage of this report

This report has been developed by the company CryptoSec (the Service Provider) based on the result of security audit of Ethereum Smart Contracts defined by CRYPTOINDEX (the Client). The document contains information on discovered vulnerabilities, their severity and methods of exploiting those vulnerabilities, discovered in the process of the audit.

The information, presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of the Client.

If you are not the intended receiver of this document, remember that any disclosure, copying or dissemination of it is forbidden.

2 Acronyms and Abbreviations

ETHEREUM – An open source platform for creating decentralized online services (called DApps or Decentralized applications) based on the blockchain that use smart contracts.

ETH (ether) – The cryptocurrency and token in the Ethereum blockchain that is used for payment of transactions and computing services on the Ethereum network.

SOLIDITY – An object-oriented programming language for creating smart contracts defined for use on Ethereum platform.

SMART CONTRACT – A computer algorithm designed to create and support contracts recorded on a blockchain.

ERC20 – The Ethereum token standard used for Ethereum smart contracts. It is a set of rules for the implementation of Ethereum tokens.

SAFEMATH – A Solidity library created for secure mathematical operations.

SOLC – A compiler for Solidity.

3 Introduction

3.1 Vulnerability Level

- Low severity – A vulnerability that does not have a significant impact on the use of the contract and is probably subjective.
- Medium severity – A vulnerability that could affect the desired outcome of executing the contract in certain scenarios.
- High severity – A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
- Critical severity – A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates the risk that the contract may be broken.

4 Main Results

Source code: https://gitlab.com/smartcontracts/cryptoindex
Last commit: 5aecb81dab61322afde237ff7902324547747b1c

4.1 Vulnerability table for all smart contracts

Contract name	Vulnerability level			
	Critical	High	Medium	Low
SafeMath	-	-	-	-
ERC20Basic	-	-	-	-
ERC20	-	-	-	-
BasicToken	-	-	-	-
StandardToken	-	-	-	-
Ownable	-	-	-	-
Cryptoindex100	-	-	-	2
Genaral comments	-	-	-	1
Totals	-	-	-	3

4.2 Evaluation of SafeMath library

This library meets modern security requirements and does not require changes.

4.3 Evaluation of ERC20Basic contract

This contract meets modern security requirements and does not require changes.

4.4 Evaluation of ERC20 contract

This contract meets modern security requirements and does not require changes.

4.5 Evaluation of BasicToken contract

This contract meets modern security requirements and does not require changes.

4.6 Evaluation of StandardToken contract

This contract meets modern security requirements and does not require changes.

4.7 Evaluation of Ownable contract

This contract meets modern security requirements and does not require changes.

4.8 Evaluation of Cryptoindex100 contract

4.8.1 Low Severity

- × For consistency in calculations, the "SafeMath" library should be used everywhere instead of "*", "/" and "-".
- ✓ *Recommendation*: Update "SafeMath" library and use for all arithmetic actions.

- × This contract contains "now" (alias for "block.timestamp") thus miners can perform some manipulation. In this case miner manipulation risk is really low.
- ✓ *Recommendation*: Consider the potential risk and use "block.number" if necessary.

4.9 General Comments

4.9.1 Low Severity

- × Coding style of the project has several observations.
- ✓ *Recommendation*: Improve code quality. Even better, use a coding style from solidity doc.

4.10 Code coverage

File name	Statements		Branches		Functions		Lines	
	Percent	Amount	Percent	Amount	Percent	Amount	Percent	Amount
Cryptoindex100.sol	84.62	99/117	50	34/68	77.78	21/27	85.25	96/114
Total	84.62	99/117	50	34/68	77.78	21/27	85.25	96/114

Result: The code coverage is more than 80% thus the contract has good code coverage.
Recommendation: add tests for branches and functions.

5 Conclusion

Serious vulnerabilities were not detected. However this contract has minor issues that may affect certain functions or cause inconvenience when using it. However, they will not prevent the contract from functioning as intended; the smart contract has been tested.

6 Tools Used

The audit of the contract code was conducted using:

- the Remix IDE (http://remix.ethereum.org);
- the Solidity-coverage (https://github.com/sc-forks/solidity-coverage).

The Solidity compiler version used was 0.4.24 (the most recent stable version).