Arachnid / **chainlink-audit-draft.md** Secret

Last active Mar 10, 2019 • Report abuse

Embed ▾ `<script src="https://gi` Download ZIP

chainlink-audit-draft.md

Introduction

On 2019-02-01, Nick Johnson performed an audit of the Chain Link smart contracts. My findings are detailed below.

I, Nick Johnson have no stake or vested interest in Chain Link. This audit was performed under a contracted hourly rate with no other compensation.

Authenticity

This document should have an attached cryptographic signature to ensure it has not been tampered with. The signature can be verified using the public key from Nick Johnson's keybase.io record.

Audit Goals and Focus

Smart Contract Best Practices

This audit will evaluate whether the codebase follows the current established best practices for smart contract development.

Code Correctness

This audit will evaluate whether the code does what it is intended to do.

Code Quality

This audit will evaluate whether the code has been written in a way that ensures readability and maintainability.

Security

This audit will look for any exploitable security vulnerabilities, or other potential threats to either the operators of ChainLink or its users.

Testing and testability

This audit will examine how easily tested the code is, and review how thoroughly tested the code is.

About Chain Link

Chain Link is middleware to simplify communication with blockchains. It provides a system for interacting with offhcain oracles from EVM smart contracts.

Terminology

This audit uses the following terminology.

Likelihood

How likely a bug is to be encountered or exploited in the wild, as specified by the OWASP risk rating methodology.

Impact

The impact a bug would have if exploited, as specified by the OWASP risk rating methodology.

Severity

How serious the issue is, derived from Likelihood and Impact as specified by the OWASP risk rating methodology.

Overview

Source Code

The Chain Link smart contract source code was made available in the smartcontractkit/chainlink Github repository.

The code was audited as of commit `5327f9d694a5ce2ad0a5688ee06241d7df12a97c` .

The following files were audited:

```
SHA1(solidity/contracts/Chainlink.sol)= 0d0850151963e728ee92d9cfa4313b002179deea
SHA1(solidity/contracts/Chainlinked.sol)= ad001d0d4f69db6821eaed8e0349e677e8f09a12
SHA1(solidity/contracts/Oracle.sol)= f7acf51aa47a7b2732d558c843d72aa98c9446be
# SHA1(solidity/contracts/examples/BasicConsumer.sol)= 4c4f41591c0186d1e6d84ea33219f07e8652b5f8
SHA1(solidity/contracts/examples/Consumer.sol)= 794c469149d241754cbeb2c647750fa9c1c12ed0
SHA1(solidity/contracts/examples/UpdatableConsumer.sol)= c0b4ad0c71aa3a519a867f8d4984e21cf8bfe843
```

Other files in this repository were not audited.

General Notes

Contracts

Testing

An automated build is configured, and shows passing tests and good coverage.

Findings

Findings without security impact

An earlier version of this report found a number of minor bugs, all of which have since been remedied by the ChainLink team.

Note Issues

None found.

Low Issues

None found.

Medium Issues

None found.

High Issues

None found.

Critical Issues

None found.