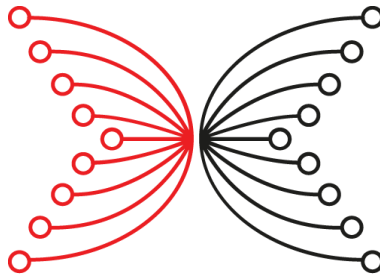


Response to Security Audit Report (Byron Reboot)



Cardano Delivery Team

April 17th 2020



Table of Contents


Table of Contents	1
Document History	2
Overview	3
Mitigations Applied	5
Issue 1: Insecure Genesis Key Generation	5
Issue 2: Code Practice – ReadFile	6
Issue 3: Code Practice and Potential Resource Usage – Async Read	7
Issue 4: Potential Resource Usage/Denial of Service (DoS)	7
Issue 5: Potential Protocol Incompletion – Static Node Set	7
Issue 6: Primitive Usage – Mock Crypto	8
Issue 7: Weakened Protections – CSP in Electron App Daedalus	9
Issue 8: Blake Hash Function Only Performed Once When Applying a Spending Password	10
Issue 9: Address Randomization Suggestion	11
Issue 10: Potential Future Issue - Payment URI	11
Issue 11: Theoretical DoS (Denial of Service) Vulnerability	12
Issue 12: Update Process	13
Issue 13: IOHK Monitoring Web Frontend Exposure	13



Document History

Version	Date	Activity
1.	2020-01-31	Phase 1 Audit Report Received from Root9B
2.	2020-02-03	Internal Discussion of Phase 1 Audit Issues
3.	2020-02-04	Initial Issue Resolution
4.	2020-03-05	Phase 2 Audit Report Received from Root9B
5.	2020-03-12	Further Issue Resolution
6.	2020-03-20	Initial Responses Prepared
7.	2020-04-16	Collation of Responses and Report Formatting
8.	2020-04-17	Final additions by Daedalus Team
9.	2020-04-17	Report Sign-Off by IOHK Director of Cybersecurity

Public distribution of this report authorised by:


.....

2020-04-17
.....

Charles Morgan
Director of Cybersecurity, Input Output HK

Date

While every attempt has been made to ensure the accuracy and correctness of this document, no warranty is provided, whether express or implied. The document is provided for information purposes only.

2020-04-17



Overview

This document responds to Phase 1 and Phase 2 of the security audit on the IOHK Byron Reboot Software that was commissioned from Root9B by IOHK in October 2019. In total, the report raised the 13 issues that are outlined in the tables below.

Issues Raised in the Phase 1 Report

Issue No.	Severity	Issue	Mitigated
1	Critical	Insecure Genesis Key Generation	Not Applicable
2	Minor	Code Practice – ReadFile	Yes
3	Minor	Code Practice and Potential Resource Usage – Async Read	Yes
4	Minor	Potential Resource Usage/Denial of Service (DoS)	Yes
5	Minor	Potential Resource Usage/Denial of Service (DoS)	Yes
6	Minor	Primitive Usage – Mock Crypto	Yes
7	Moderate	Weakened Protections – CSP in Electron App Daedalus	Yes



Issue No.	Severity	Issue	Mitigated
8	Minor	Blake Hash Function Only Performed Once When Applying a Spending Password	Yes
9	Potential	Address Randomization Suggestion	Not Applicable
10	Potential Future Issue	Payment URI (daedalus:// or similar)	Not Applicable
11	Theoretical	DoS Vulnerability	Yes

Issues Raised in the Phase 2 Report

Issue No.	Severity	Issue	Mitigation Applied
12	Minor	Update Process	Yes
13	Minor	IOHK Monitoring Web Front-end Exposure	Yes



Mitigations Applied

Issue 1: Insecure Genesis Key Generation

We have reviewed this issue. It applied only to our internal test systems, and not to any public systems. It therefore presented no real security risk.

The report identifies a flaw in a CLI tool used for QA & integration testing of node clusters. This CLI tool can generate all the necessary test keys and a test genesis file to start a cluster of nodes locally. This is used in integration tests and in benchmarking scripts where a local cluster is needed. This is unrealistic compared to a real public cluster because all the keys are generated by one person (or script) together in one place, which would not be a secure thing to do for a real public cluster. Nevertheless it is extremely helpful to have such a command for QA purposes.

Indeed this mode of the tool **has never been used for creating a real genesis file and associated keys used for a real network.**

Setting up a real public network is a much more elaborate process and uses a different tool to securely generate keys offline. For the Cardano mainnet, each genesis key was generated (with a purpose-built tool) by individuals from each of the three organisations in the Cardano federation, on separate computers never connected to the internet. Each genesis key was used to sign an operational key certificate that is used to run the Byron core nodes. Only the operational key certificates were moved from the offline computers, and the genesis keys remain offline. The purpose-designed key generation tool follows proper code security practices, including getting the key generation entropy from the system's `/dev/random`.



The report explains the flaw in the CLI tool for creating test genesis keys (and a genesis file) which would mean that these test keys would be insecure because they are derived from only 32bits of entropy.

This is correct, but it is fortunately of no consequence since **such keys would never and have never been used for any purpose other than convenient testing.**

Nevertheless, it is wise to follow secure code practices even in testing tools, so code changes were made to use a proper source of entropy for generating the test keys and to clarify in the code comments that it can only be used for test purposes because it generates all the keys in one go and in once place. (<https://github.com/input-output-hk/cardano-ledger/pull/732>). These changes have been independently reviewed by IOHK Director of Cybersecurity and by suitably qualified internal experts.

Issue 2: Code Practice – ReadFile

We have reviewed this issue, and have fully addressed it.

Suitable code changes were made in line with the recommendations of the report (<https://github.com/input-output-hk/ouroboros-network/issues/1573>).



Issue 3: Code Practice and Potential Resource Usage – Async Read

We have reviewed this issue, and have fully addressed it.

Suitable code changes were made in line with the recommendations of the report (<https://github.com/input-output-hk/ouroboros-network/issues/1574>).

Issue 4: Potential Resource Usage/Denial of Service (DoS)

This issue has been fully addressed.

Timeouts had already been included in our development plan, but were not present at the time the snapshot of the code was sent to Root9B for review. This development was completed prior to deploying nodes for production use on mainnet.

(<https://github.com/input-output-hk/ouroboros-network/issues/1575>).



Issue 5: Potential Protocol Incompletion – Static Node Set

We have reviewed this issue, and concluded that it does not apply. We have clarified this in the source code repository.

This appears to be a misunderstanding of the purpose of the Chairman program, which is just for a Continuous Integration test. It is not expected to work in a dynamic environment. The node does re-establish connections, it is just the chairman monitor that does not, because such failures count as Continuous Integration test failures. The purpose of the chairman program has been clarified in order to avoid future confusion on this issue.

(<https://github.com/input-output-hk/cardano-node/issues/671>)

Issue 6: Primitive Usage – Mock Crypto

We have reviewed this issue. It does not apply at this point in time, and will be addressed through future planned development work.

During early stages of Ouroboros Praos development for Shelley, and for the purpose of certain automated tests, the KES and VRF cryptographic primitives use mock implementations. In parallel, the real KES and VRF implementations have been in development. At the time of the snapshot that was used for the Root9B audit, the mock cryptographic implementations were still in use. Mock cryptographic primitives will not be used in any mainnet deployment, and so there will be no risk to the user.

(<https://github.com/input-output-hk/ouroboros-network/issues/261>)



Issue 7: Weakened Protections – CSP in Electron App Daedalus

We have reviewed this issue. There is no risk at this point in time, and planned development work will eliminate any potential future risk.

Unsafe-eval is required to run WASM code

(<https://github.com/WebAssembly/content-security-policy/issues/7>).

This will not be necessary in the future when `wasm-eval` is implemented in Chromium. With one exception, Daedalus does not load any external content, so the risk of weakened protection is not a general issue. The only content that is currently loaded is that for the newsfeed. This is safe since it is loaded over HTTPS with domain SSL certificate verification and it is also protected by hash verification of the content. Content and verification hashes are loaded from different HTTPS sources, providing additional protection.

Issue 8: Blake Hash Function Only Performed Once When Applying a Spending Password

We have reviewed this issue. It has already been addressed in the latest versions of Daedalus through changes to the algorithms and inter-component communication processes that we are using. Blake hashing will be completely phased out in all versions of Daedalus in the near future.

Blake hashing has been removed in Daedalus Rewards (and subsequently Daedalus Flight), prior to the receipt of this report. It is still present in Daedalus Classic, which will be phased out in the near future. Spending passwords are sent in plain-text format using a secure TLS connection between the Daedalus frontend and the Cardano wallet backend.



Issue 9: Address Randomization Suggestion

We have reviewed this issue, and concluded that there is no security risk.

As hypothesized in the report, the reason is to avoid port conflicts. This is detailed in the option description for the command line:

```
--random-port
```

```
serve wallet API on any available port (conflicts with --port)
```

This option is used by the Daedalus launcher since we have frequently encountered situations where our chosen default port (8090) is already in use for some other purpose. This option works around this issue by picking a port that is available for use. This has nothing to do with cross-site request forgery or DNS rebinding.

Issue 10: Potential Future Issue - Payment URI

We have reviewed this issue. It does not apply at this point in time, and we do not anticipate it being a future concern. However, we will take it into consideration during relevant design and implementation phases.

This form of payment URI is not currently used and there is therefore no current risk. If we do decide to use this form of payment URI in future, then we will address this risk during the initial design and subsequent implementation phases.



Issue 11: Theoretical DoS (Denial of Service) Vulnerability

We have reviewed this issue. This is a theoretical issue that applies to any system that operates over a public network. The current Cardano system mitigates against a Distributed-Denial-of-Service attack by cleanly separating the block-producing nodes from the public network, and by using independent public relays to transmit blocks and transactions to the public internet. This means that a DDoS attack will not disrupt the block producing nodes. The future decentralised development will provide resilience through its highly distributed nature and also through its use of the Ouroboros Praos private slot-leader schedule. The decentralised blockchain will not be dependent on any individual operator or specific group of operators to maintain its overall integrity. The risk is therefore minimised.

This issue is currently mitigated by the federated design of the existing Cardano system (Byron). As the report states, the use of the Ouroboros Praos private slot-leader schedule will mitigate this in the decentralised setting that we are moving towards (Shelley). Moreover, in the decentralised setting, an attack on any one user/operator will have limited effect on the operation of the blockchain as a whole, since the system is precisely designed so that one stakepool may be substituted for another that becomes unavailable, and the network is designed to dynamically adapt to changes in connectivity and availability. Defence in depth is ensured by engaging and maintaining sufficient numbers of stake pools, through the use of built-in blockchain protocol parameters that can be adjusted as required to meet overall resilience and reliability needs.



Issue 12: Update Process

We have reviewed this issue. It has been addressed through a new update process that forms part of the Byron Reboot production system.

A revised update process has been designed and released in recent versions of the Daedalus Flight wallet. The update system mentioned in the report will no longer be used with the first release of the Byron reboot production version of Daedalus (Daedalus 1.0.0), which will be released to all Daedalus users in April 2020.

Issue 13: IOHK Monitoring Web Frontend Exposure

We have reviewed this issue and fully addressed it through changes to the source code.

The issue has been considered and suitable code changes have been applied (<https://github.com/input-output-hk/cardano-node/issues/672>).