Security assessment and code review

Initial Delivery: November 23, 2021
Most Recent: December 10, 2021

Prepared for:
Remy Roy | Wagyu
Colfax Selby | Wagyu
Inan | Wagyu

Prepared by:
Mikerah Quintyne-Collins | HashCloak Inc
Peter Lai | HashCloak Inc

Table Of Contents

Executive Summary

Stake-house engaged HashCloak Inc for an audit of their Wagyu Key-Gen software, an Electron-based desktop application for generating Eth2.0 keys. The audit was done with two auditors over a 2 week period, from November 8, 2021 to November 22, 2021. The Stake-house codebase was assessed at commit 2d526027a501ecbb12cbe67efd2546bb701fccbe.
From December 3, 2021 to December 10, 2021, the Stake-house codebase was reassessed at commit d3e108fa378e6295a6826c97811d283cd10caf38. All the issues brought up during the initial audit were fixed and no new issues have been identified.

The files within scope were all files in the **src** directory and all files ending in **.py**, **.sh** and **.bat**.

During the first week we familiarized ourselves with the Stake-house code base and started our manual analysis of the Key-Gen software. In the second week we further investigated the code base and an automated analysis was performed with off-the-shelf automated tools.

We found a variety of issues ranging from Medium to Informational and provide some general guidance to improve the code quality.

Severity	Number of Findings
Critical	0
High	0
Medium	1
Low	2
Informational	5

Overview

The Stake-house Wagyu Key-Gen software is a GUI application providing functionality to the eth2.0-deposit-cli. The deposit-cli was audited by Trail of Bits, is a tool for creating EIP-2335 format BLS12-381 keystores and a corresponding deposit_data*.json file for Ethereum 2.0 Launchpad.

The Wagyu Key-Gen code base can be separated into several logical units. The main ones are:

- src/electron	The electron GUI application that runs react.
- src/react	The react application that handles generating keys and validating keys.
- src/scripts	The python script that proxy commands to eth2 deposit cli.

Assumptions

Throughout the audit, the following assumptions were made:

- User assumptions
    - The user cares about the privacy of information (deposit data file, validator keys, passwords, mnemonic) generated using Wagyu Key Gen
    - The user keeps such private information offline, in cold storage.
    - The user will generate their own keys but may use their keys on an external service for staking which they may or may not control. For example, they may use a VPS that they control or use a third party staking service.
- Adversary assumptions
    - An adversary can have physical access to the machine on which the Wagyu Key Gen application is running
    - An adversary may be nearby a user as they are generating keys through the Wagyu Key Gen application.
    - An adversary may attempt to install malware on a user's device in order to extra the information generated by the Wagyu Key Gen tool.
    - An adversary may set up a website in order to fool users to download the adversary's version of the Wagyu Key Gen tool.

- Wagyu developer assumptions
    - The Wagyu developers should not have access to any keys generated by the Wagyu Key Gen tool
    - The Wagyu developers sign every release of the Wagyu Key Gen tool.
    - The PGP keys used for signing will either be kept secret forever or will expire.
- Other assumptions
    - The Developers, authors or distributors of the programming languages used by the Wagyu Key Gen tool should not be able to inject code into the bundled application giving them access to private information generated by the tool.
    - The Developers, authors or distributors of the operating systems used by a Wagyu Key Gen user should not
        - Be able to inject code into the application executable to extra private information generated by the tool.
        - Be able to have file level access to the resulting private information generated by the tool.

Findings

Clipboard is not cleared upon exiting the application

Type: Medium Severity
Files affected: MnemonicGenerationWizard.tsx

When exiting the Wagyu Key Gen application, the clipboard is not cleared. As such, any passwords or mnemonic that the user may have copied to their clipboard is still accessible to anyone with physical access to the air-gapped machine.

Impact: An adversary may be able to retrieve the password and mnemonic necessary for creating an Eth2 validator.
Suggestion: Clear the clipboard upon exiting the Wagyu Key Gen Application.
Status: The Wagyu team issued a commit to fix this issue. They cleared the clipboard when the user quit the application.

Out of date dependencies

Type: Low Severity
Files affected: package.json

As per Electron guidelines, all dependencies, including Electron itself, should be most of to date at the time at which the Electron app is released. This is to ensure that any patches that are pushed to dependencies are applied to your application.

Impact: An unpatched vulnerability in a dependency used in Wagyu Key Gen may cause unexpected behavior.
Suggestion: Run npm audit and carefully look through all the dependencies that are not up to date and that have a known vulnerability.
Status: The Wagyu team issued a commit to fix this issue. They upgraded dependencies to the latest version.

Unused dependencies

Type: Low Severity
Files affected: package.json

Having unused dependencies conflicts with the principle of least authority which states that applications should operate with the minimum necessary privileges on a given system. As such, if Wagyu Key Gen doesn't need a particular dependency, it should be removed from the package.json file.

The following packages are unused:

- @rauschma/stringio
- @types/git-revision-webpack-plugin
- Js-yaml

The following devDependencies are unused:

- @babel/core
- @types/js-yaml
- Babel-loader
- Webpack-cli

Impact: May introduce vulnerabilities into the Wagyu Key Gen application.
Suggestion: Remove unused dependencies.
Status: The Wagyu team issued a commit to fix this issue. They removed these unused dependencies. Further, as per discussions with the development team, the devDependency webpack-cli is used for building with yarn and as such is not unused.

nodeIntergration is set to true

Type: Informational
Files affected: index.ts

By default, nodeIntegration should be set to false as a security precaution. This is due to the fact that when in a renderer that may handle remote content, it is possible for an attacker to execute Javascript within the renderer. Due to the assumption that Wagyu Key Gen is ran within an air-gapped computer, this problem is minimized though someone with physical access to the device can still potentially run untrusted Javascript.

Suggestion: Set nodeIntegration to false. We recommend that the Wagyu team consider using a preload.ts file in which all needed node.js dependencies are loaded with the Wagyu Key Gen application in order to remove reliance on node.js. Further, we

also recommend that contextIsolation be set to true in order to ensure that both renderer and main components of the Wagyu Key Gen application are communicating through IPC.
Status: The Wagyu team issued a commit to fix this issue. They used preload.ts to load all nodejs dependencies, and updated settings so that the application and main components can communicate through IPC.

setPermissionRequestHandler is not set and by default allows external permissions

Type: Informational
Files affected: index.ts

By default, Electron approves all permissions from external sources. Since Wagyu Key Gen is a security conscious application, this is not a reasonable default.

Suggestion: Disallow permissions by default for Wagyu Key Gen.
Status: The Wagyu team issued a commit to fix this issue. They disallowed external permissions.

Mnemonic password is the empty string by default

Type: Informational
Files affected: eth2deposit_proxy.py

By default, the mnemonic password is set to the empty string with no prompt for the user to also set a password for their mnemonic as well. Although, as long as the mnemonic is generated securely, is sufficiently random and stored safely, it should provide similar guarantees as stated in the BIP32 specification.

Github and Discord links to do not render under current assumptions

Type: Informational
Files affected:
There are links to Github and Discord on the main page of the Wagyu Key Gen application. Since it is assumed that a user will be using an air-gapped machine, they will not be able to navigate to these sites.

Suggestion: Remove links to Github and Discord.

Status: The Wagyu team issued a commit to fix this issue. They showed Github and Discord url instead of a navigating link.

Keys maybe able to be extracted through side channel attacks

Type: Informational

Due to the assumptions used throughout the audit, we simply include this finding for completeness.

It is possible for an adversary to extract private keys from an air-gapped using malware, and a variety of techniques based on physical access, electromagnetism, optical signals, etc.