Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Summary-Report Standard Notes Pentest 10.2019

Cure53, Dr.-Ing. M. Heiderich, N. Hippert, BSc. T.-C. “Filedescriptor” Hong, M. Kinugawa.
Dr. N. Kobeissi, Dipl.-Ing. A. Aranguren (ext.)

Index

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Introduction

“Standard Notes is a safe place for your notes, thoughts, and life's work. A free, open-source, and completely encrypted notes app.”

From https://standardnotes.org/

This report summarizes the findings of a thorough and large-scale assessment targeting the Standard Notes software compound, which consists of various different items, including a desktop application, mobile applications, server-side code and other aspects. Beyond the components of the Standard Notes software compound, cryptographic design and implementations were audited as well. Carried out by Cure53 in October 2019, this evaluation included a penetration test and a dedicated code audit. The project yielded twenty-one security-relevant discoveries, including two marked as “*Critical*” in terms of severity and impact.

As for the timeline, resources and processes deployed during this project, it should be noted that the assessment was requested by Standard Notes and executed by five members of the Cure53 team together with one external tester. In order to address the goals of the project in a comprehensive and structured fashion, five work packages were proposed and subsequently completed. While WP1 was dedicated to Cure53 examining the Standard Notes Web Application using AngularJS, the WP2 work concentrated on Standard Notes Mobile Applications written in React Native. Next, in WP3, Standard Notes Desktop Application using Electron were evaluated. Moving on to a dedicated cryptographic assessment, WP4 looked at SFJS Cryptographic Library for key Handling & Item De-/Encryption. Last but not least, WP5 tackled Standard Notes Syncing Server written in RoR.

It has been agreed by the participating parties that a white-box approach is an optimal method here, especially since a lot of code is available as open source software. In addition, Cure53 was given access to documentation, custom builds that work against a test environment and test servers to use during testing. The team further took advantage of local builds for examining the scope. Cure53 managed to reach good coverage while at the same time avoiding coming to contact with actual PII from production.

The test was very well-prepared by the Standard Notes team and progressed efficiently. This is also due to effective communication enabled by a shared and dedicated private Slack channel that Cure53 created for this assessment. It needs to be noted that the overall budget dedicated to this project in October 2019 amounted to eighteen person-days.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Among the twenty-one findings, Cure53 classified eight as vulnerabilities and noted ten as general weaknesses with lower exploitation potential, and three as false alerts with intended behavior. As noted above, two findings received highest-possible "*Critical"* severity ratings. Both affect the Standard Notes desktop client software and make it possible for an attacker to cause Remote Code Execution. Both issues have been live-reported and discussed with the maintainer team. All other issues only exposed "*Medium"* or lower-scoring risks, which is a rather good sign despite the conversely extensive number of findings in total.

In the following sections, the report will first briefly reiterate the scope and then moves on to dedicated, chronologically discussed ticket stubs, which shed light on the discoveries' headline and fix status one-by-one.

Note: *This report was updated on October 28th 2019 to add fix notes to each ticket that was already addressed by the Standard Notes team. The fixes were all verified by the Cure53 team.*

Note: *A 2nd update to the report was added on September 2nd 2020, in which the remaining unaddressed issue, SN-01-014, was flagged as successfully fixed as well.*

Scope

- **Standard Notes Software**
  - **WP1**: Standard Notes Web Application using AngularJS
    - https://github.com/standardnotes/web
    - https://github.com/standardfile/ruby-server
    - https://github.com/standardfile/ruby-server/blob/master/docker/nginx.conf
  - **WP2**: Standard Notes Mobile Applications written in React Native
    - https://github.com/standardnotes/mobile
  - **WP3**: Standard Notes Desktop Application using Electron
    - https://github.com/standardnotes/desktop
  - **WP4**: SFJS Cryptographic Library for Key Handling & Item De-/Encryption
    - https://github.com/standardfile/sfjs
  - **WP5**: Standard Notes Syncing Server written in RoR
    - https://github.com/standardfile/rails-engine
- A Test Environment was made available
  - https://testapi.standardnotes.org
  - https://testapp.standardnotes.org
  - https://testsite.standardnotes.org
- Sources were available on GitHub; Standard Notes is an Open Source Software
- Additional documentation was made available to Cure53

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Identified Vulnerabilities

The following sections list both vulnerabilities and implementation issues spotted during the testing period. Note that findings are listed in chronological order rather than by their degree of severity and impact. The aforementioned severity rank is simply given in brackets following the title heading for each vulnerability. Each vulnerability is additionally given a unique identifier (e.g. *SN-01-001*) for the purpose of facilitating any future follow-up correspondence.

SN-01-001 General: Insufficient password complexity requirements *(Low)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-004 Desktop: RCE via protocol check bypass *(**Critical**)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-005 Desktop: RCE via unsanitized HTML in *Extensions* *(**Critical**)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-014 Crypto: Password-hashing not resistant to GPU attacks *(Low)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the updated codebase on Github.*

SN-01-017 iOS: JWT Token and PII leaks via missing *Data Protection* *(Medium)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the updated codebase on Github.*

SN-01-018 Android/iOS: No Pinning indicates possible account-takeover *(Medium)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the updated codebase on Github.*

SN-01-019 iOS: Keychain data accessible on locked devices *(Medium)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

SN-01-021 Extension: CSS injection via *postMessage* *(Low)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

Miscellaneous Issues

This section covers those noteworthy findings that did not lead to an exploit but might aid an attacker in achieving their malicious goals in the future. Most of these results are vulnerable code snippets that did not provide an easy way to be called. Conclusively, while a vulnerability is present, an exploit might not always be possible.

SN-01-002 Desktop: *nodeIntegration* enabled in renderer *(Info)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-003 Desktop: Missing *contextIsolation* mitigation *(Info)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-006 Web: Security vulnerabilities in outdated *Phusion* version *(Info)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-007 Web: Security vulnerabilities in outdated *nginx* version *(Info)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-008 Web: General HTTP security headers missing *(Low)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-009 Android: Usage of deprecated permission *(Info)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-010 Android/iOS: Multiple vulnerabilities in outdated modules *(Info)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the updated codebase on Github.*

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

SN-01-011 SCA: Hash verifications not in constant-time *(Low)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-012 SCA: Debug logs enabled on production *(Info)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-013 Crypto: No HMAC check in constant-time *(Info)*

Note: *This issue was addressed by the Standard Notes team and the fix was verified by Cure53 by inspecting the respecting PR/Diff on Github.*

SN-01-015 Android 5 and older: Vault access/passcode bypass via backups *(Info)*

Note: *The described behavior is in fact intended, the issue was therefore flagged to be a false alert an the severity was lowered to “Info”.*

SN-01-016 iOS: Limited PII access via backups *(Info)*

Note: *The described behavior is in fact intended, the issue was therefore flagged to be a false alert an the severity was lowered to “Info”.*

SN-01-020 Android/iOS: Data export leaks due to missing warnings *(Info)*

Note: *The described behavior is in fact intended, the issue was therefore flagged to be a false alert an the severity was lowered to “Info”.*