Preliminary security report for Vanish

MLC Consulting

2017 September 2

Disclaimer and Third Party Sharing

MLC Consulting attests that this is a comprehensive reporting of the work performed in relation to the security audit.

However, MLC Consulting does not certify that the software analyzed is free of defects or vulnerabilities. This report should not be used as proof or advertisement of any aspect of the software's security.

As a preliminary report, this is missing many standard analyses (e.g. protocol analysis, static analysis). The report also has incomplete component coverage (the Android client is not analyzed).

This document should ***not*** ordinarily be shared with third parties. However, if necessary to share, the document ***must*** be presented to third parties in whole (particularly, ***with this page attached***).

Services Performed

The source was preliminarily analyzed to determine compilation and usage characteristics. A local testing environment was set up with dummy keys and internet addresses. The back-end (server) was pen-tested (mostly via fuzzing) without any further knowledge of source. The front-end (chat client) and back-end was tested in concert, with packet capture and subsequent analysis to determine any leaks of unencrypted information.

Then, a cursory source code analysis was performed to (1) determine promising targets for pen-testing and (2) provide preliminary suggestions. Further pen-testing informed by this analysis was performed.

Fuzzing report

Server

Random fuzzing with stock tools is insufficient to recover secret information.

After a source review, targeted fuzzing was similarly unsuccessful in recovering secret information.

In this case, secret information means plaintext (binary), hex-encoded, or base64-encoded data at the packet level.

Notably, metadata exfiltration was not analyzed. This requires a protocol review to determine which metadata is intended to be transmitted as-is over wire.

Crashes and hangs were observed on multiple fuzzed requests.

iOS

Similarly, fuzzing the iOS client over network is insufficient to recover secret information. Same caveat about metadata.

Crashes and hangs were observed on multiple fuzzed requests.

Packet analysis

No plaintext messages were intercepted over the wire during normal operation of the application. No base64 or hex-encoded messages were intercepted.

Further pentesting

Informed by source code analysis, more advanced fuzzing (protocol-consistent) and connection failure modes were tested on client and server. No message data (plain or encoded) was recovered.

Suggestions from preliminary source audit

In the course of preliminary source analysis, the following potential action items were uncovered. This is *very* shallow and a comprehensive source audit (to include static analysis) should be performed at a later date.

Restrict SSL cipher suite

There is a laundry list of SSL cipher suites in `cluster.coffee:ciphersExplicit`. Although these suites are all probably necessary for a general-purpose web application, the restriction of Vanish to mobile (and iOS in particular) suggests a restriction of this cipher suite list to the following:

- `ECDHE-ECDSA-AES256-GCM-SHA384`
- `ECDHE-ECDSA-AES256-SHA384`

This is sufficient to support iOS versions 6-10 and Android versions 4.4/7 (not sure about Android 5 and 6, and of course, further testing should be done to verify compatibility in all settings). Other ciphers may be re-added as necessary to support new client types.

Move encryption keys out of the repository

It is encouraging that the `sslLocal/` directory was not stored in the source repository. However, `private/`, `apnLocal/` and `pushKitLocal/` contain keys that should not be trusted to a general-purpose repository host such as GitHub.

Remove unused packages

Many packages listed in `package.json` are not actually used by the server source code (e.g. `chai`). Although there should be negligible impact since these packages are never `require`d, it is still good practice to keep dependency lists pruned to keep attack surface area low.