Project Plan

Aidan Fray

May 29, 2019

1 Title

"Security evaluation of pEp's TrustWord implementation"

2 Motivation

- Use-case for TrustWords: Users are often deterred by difficulty in key authentication and end-to-end encryption (e.g. PGP web-of-trust).
- Other word lists result in a higher number of words to compare. Trustwords mapping of a word to 2 bytes results in a lower number of words. Meaning the possibility of increased usability
- However, the mapping of words to 16-bits is yet to be proved as secure as this is the highest number of bits per word seen in the literature.

3 Goals

Overall the research will aim to investigate the strength of pEp's Trustword fingerprint mappings, and the ease in which partial collisions can be obtained for keys and how this will ultimately affect the end user(s). This will be accompanied by recommendations into how the TrustWord system can be altered to provide maximum security alongside research into what makes an effective wordlist.

4 Possible research questions

- Is the recommended minimum number of Trustwords enough to provide a basic level of security?
- What attributes make a strong general wordlist for fingerprint mapping and does the Trustword implementation exhibit these features?
- What are some of the most effective ways of measuring linguistic distance or similarity?
- How easy is it to generate similar keys that attack a targeted key pair?
- How can the search for similar keys be assisted? Could weighting them like "*Fuzzy Fingerprints Attacking Vulnerabilities in the Human Brain*" help to find partial matches?
- How can similarity be quantified in terms of words? Does this include pronunciation or visual aspects?
- As usability is the main justification for the use of Trustwords are there alternatives that provide the same usability but with more security?

5 High Level View

Wordlist characteristic analysis

This area will look into characteristics that are attractive when designing a wordlist. These will be used later to possibility improve the trustword implementation.

Analysis into effective "similarity" metrics

TODO

Development of a tool used to find partial collisions for wordlist mappings

This tool would be fed two keys, one static and one the "target". The tool would then generate keys for the "target" that create a similar overall wordlist mapping. Here metrics for linguistic distance will be used to define "similarity". The similarity metric will generate a list of keys that provide similar matches, the tool will then hash large number of PGP keys to generate partial matches.

Security of Trustwords

This section will evaluate Trustwords using the previously defined metrics. Alongside this, the feasibility of generating partial collisions will be assessed. The security could be quantified by testing the current implementation out on real users and determining the False acceptance rate of near attacks. This could be used as a baseline.

Security of alternative wordlists:

In the same way as Trustwords were assessed, the same will be performed on already available wordlists. Such as the PGP wordlist and the wordlist generated by SafeSlinger.

Security of implemented recommendations

This section will then amalgamate all the research performed prior and use it to improve and update the implementation of Trustwords. This would then be assessed by experimentation and the improvements could be quantified with the same experimentation.

6 In-depth breakdown

6.1 Wordlist characteristic analysis

- Investigation into characteristics of a strong wordlist
  - Distinctiveness
  - Homonyms (Visual)
  - Homophones (Pronunciation)
  - Length
  - Current ways to define linguistic distance:
    - **Visual:**
      - Levenshtein (Edit distance)
    - **Auditory**:
      - Soundex
      - Metaphone
      - NYSIIS
      - Match Rating
- Analysis into the strength of the TrustWords using previously obtained metrics.
- Investigation into the strength of other wordlists using defined metrics.
  - PGP word list
  - SafeSlinger

6.2 Security of Trustwords

Here we'd look at the words used in the dictionary and attempt to generate lists of keys that would generate partial collisions these would then be fed into the created tool

6.3 Security of alternative wordlists

6.4 Security of implemented recommendations