Twitter Steganography using manual annotation and codebooks.

gfoun@protonmail.com

Abstract

A simple steganographic system is presented that can embed shortened goo.gl URLs inside the text of a tweet. We use two keys to construct a synchronized dictionary that maps words to goo.gl URLs (codebook). The mapping and the data encoding is arbitrary and can be used to embed larger capacities and different payload types. The system can also be implemented for other social media platforms besides Twitter. It does not use formatting methods, "invisible" artifacts or grammars.

1 Introduction

This is the paper version of my project developed in the final year of my distance-learning BSc CIS degree in University of London, Goldsmiths (2014). Since then lots of things have changed, besides me losing interest. Google shutdown the goo.gl shortening service in 2018, while Twitter doubled its character limit to 280. The Twitter Search API was also changed from the ground-up and as such the proof-of-concept Python code I've written 6 years ago almost certainly won't work.
Since it took me only 6 years to write this simplified paper version, updating the code is unlikely at this point. The good news is that its fairly trivial to implement. If you do, please drop me an e-mail, I'll be happy to chat.

I got the initial inspiration from Muñoz et al[1] and investigated whether the idea can be applied in severely constrained environments such as Twitter. Different payload sizes and encodings beside goo.gl URLs are certainly possible. First, it depends on the size of your dictionary and how many words are you able to fit in without breaking synchronization. Second, mappings could not necessarily point to lexical items. You can map raw bits in order to embed a binary blob and use compression, or encode a domain-specific symbol-set such as GPS coordinates and international maritime signal flags or botnet commands.
The main idea behind the initial design is that low payload capacity in steganography is largely ignored and unexplored, and could potentially deliver more, particularly in cyberwar. Most research in steganography seems to be focused on maximizing a channel's capacity while trying to avoid detection.
On the other hand, using payloads of the size of bits or a few kilobytes guarantees invisibility.

2 System Design

1. Alice and Bob exchange two keys: the first is used as a search term, the second to randomize the order of the dictionary and the mapping of words to goo.gl URL characters.

2. Alice uses the first key to build a list of matching tweets. She removes stop words, and other artifacts, and produces a final list that contains only words.

3. Alice uses the second key to either select a random subset from that list, or use the whole list to randomly map the payload. For goo.gl URLs we need to map 62 different characters: [A-Z][a-z][0-9]. If we use a one-to-one mapping, then we need a list of 62 words. Usually the results from a Twitter search contain more.

4. To embed a URL, Alice looks-up the dictionary using the URL characters as keys and retrieves the matching words. She then manually annotates them to produce a passable looking tweet. For example if the payload is mapped to `['lay', 'barbara', 'assistant', 'monica', 'off', 'tickets']`, the annotated result could be the following:

   I hope I won't have to lay barbara. She's the assistant of monica, my wife's accountant. Anyway I'm off for tickets! #games2014

5. Bob looks up his own copy of the dictionary using the words from Alice's tweet as keys and retrieves the hidden goo.gl URL characters.

3 Synchronization

Due to Twitter's ephemeral nature, not two search operations will return exactly the same results, even if using the same criteria. This will break the synchronization and produce different codebook dictionaries between Alice and Bob. One way to fix this is to use the `until` parameter in the Twitter Search API and narrow down the results within a smaller time-window. A period of 2 days is usually enough and won't break synchronization, but this needs to be tested thoroughly. It also means that Bob will only have 2 days to extract the payload after which it will become irretrievable.
Alice will also need to exercise care and not change the order of the words she annotates. If she does, then the goo.gl URL will be garbled and Bob will not be able to access it. A similar issue will arise if Alice uses words in her annotated tweet,besides those of used in the payload, that exist in the synchronized dictionary. As a result, Bob will retrieve extra URL characters that are not valid. There is certainly ways to add extra positional information in the form of some error-detection, or implement a more robust synchronization method, by looking deeper into the Twitter API.

4 Security

There is a risk of having the codebooks become 'stale', if the same search term/key is used multiple times per day to embed information. This can cause the words in the tweets to repeat, that might trigger some alarm. The danger here is not for someone to discover that you are using steganography, but to flag you as a bot and shut-down your account.
It would seem natural to use this system as a Command & Control center for botnets but if you don't mix 'normal' tweets with embedded ones and try to use it programmatically and in high frequency, all bets are off.

References

[1] Muñoz, A., Carracedo, J., & Alvarez, I. A. (2010, June). Hiding Short Secret Messages based on Linguistic Steganography and Manual Annotation. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on (pp. 960-964). IEEE.