Tricky To unfURL:
The Risks Associated With URL Shortening Services

Thomas Hendrickson

1 Abstract

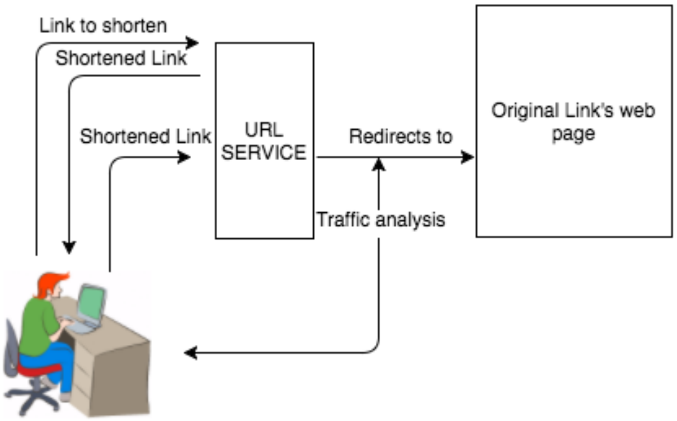
Often URL links that people commonly use are awkwardly long and seemingly obscure. This presents difficulties in situations when the link needs to be shared, such as a Google Document detailing a homework assignment given to the entire class. One solution is use a URL shortener that converts the lengthy URL into a much more manageable link. These shortened URLs generally take the form of *sho.rt*/TOKEN. The domain name is *sho.rt* and is dedicated to hosting shortened URLs, and the TOKEN is a unique identifier for this particular mapping.

This paper provides a brief overview of the technology behind URL shorteners and their overall architectures. Then it highlights some of the security problems that arise when using these URL shorteners. These consist of two main issues, leaking private information and the distribution of malware. Real examples of the problems are shown, including evidence that proves these attacks have already been used successfully in the wild.

2 Introduction

URL shorteners are starting to become more and more common. They allow a user to send a long URL link and receive a shortened one in return. When the shortened link is visited the web page is redirected to the longer one. Additionally most services provide tracking analytics on the shortened URL, this information can include the number of visits, where the link was clicked, device the link was clicked on, time of day and more [1]. This data is often sold for marketing purposes, *Bitly* is one company that uses this model as its entire business plan. Multiple services exist for shortening the URLs including: *bit.ly*, *goo.gl*, *tinyURL*, *tiny.cc*, *ow.ly*, and Twitter's *t.co* [2].

URL shortening systems have a fairly simple architecture. A client submits a long URL to be shortened: *https://www.mywebsite.com/pleaseclickonthis/buyme* to the service. Based off of the submitted link, the service computes a shortened URL. The service uses some method (cryptographic hash function, iterating through IDs, random IDs, or some combination [3]) to produce a token "XQ6953". The link sent back to the client consists of the service's domain name *sho.rt*, plus this ID token appended to the end: *http://sho.rt/XQ6953*. The service stores this associative mapping from long URL to shortened URL. When the service receives a shortened URL, it looks up the associated longer URL and redirects the web request to that link. Normally an HTTP 301 response code is used, meaning "Moved Permanently" [4]. During the redirection process the shortening service normally records data about the shortened URL query (the client's IP address, time of day, location, machine type etc.) and save this for analysis. The basic overview is illustrated in the below diagram [Figure 1].



[Figure 1]

A fundamental component of the URL shortening process is the 'token' associated with the shortened URL. To actually be a shorter URL, this token must consist of a low number of characters. *Bitly* links currently use four to seven characters per token [5]. *Bitly* URLs with six character tokens have an approximate 42% density rate (42% of all 6-character URL combinations map to valid *bit.ly* shortened URLs). This has some major privacy implications.

Overall, shortening URLs is very convenient. In many cases URLs become long and unwieldy. Someone who hopes to share a Google maps location needs to send: *https://www.google.com/maps/place/Halligan+Hall/@42.4081675,-71.1181983,17z/data=!3m1!4b1!4m5!3m4!1s0x89e376dc90ceec3f:0x775385274e764a96!8m2!3d42.4081636!4d-71.1160096* to a friend. After using Google's shortening service the link becomes: *https://goo.gl/maps/x4GzXyq7f862*. Clearly one of the two is preferable for emailing or texting to a friend. The data provided about the link's visits presents a great opportunity for marketing analysis. The data allows companies to compare marketing techniques and see which are more successful (based off of how many hits the link generated).

3 To the Community

The fundamental part of the shortening process is that the URL becomes shorter, by design losing information contained in the original URL. A safe link directing a user to *www.google.com/MYDOCUMENT123456789* now becomes *www.sho.rt/TOKEN1* and a malicious link pointing unsuspecting users to *www.evil.com/hackyourcomputer* is now *www.sho.rt/TOKEN2*. This obscurity is a huge security issue. Shortened URLs can be catalysts for malware distribution. It is impossible to tell which link *www.sho.rt/TOKEN1* or *www.sho.rt/TOKEN2* will lead to the desired document and which will lead to a site that hacks the client's computer. The URLs can be added into social engineering phishing attacks quite easily (and have been). Shortened URLs can be searched extremely quickly. Once a user submits a link to be shortened by *bit.ly*, it remains valid for as long as *bit.ly* remains working [6]. The information contained in the original link is now public, and inevitably private information will be leaked.

In 2014 Google's Safebrowsing service actually flagged all *bit.ly* links for hosting malware. All *bit.ly* access in Firefox and Chrome was temporarily suspended after Safebrowsing found many cases of trojans and other malicious software in *bit.ly* links. The ban was lifted quickly but the malware remains [7].

4.1 SEARCHING THE URLs

Because of the relatively dense URL space (42% for bit.ly six character tokens), information contained in the shortened URLs becomes public. An attacker searching for sensitive information and websites to compromise attempting to iterate over all possible domain names on the internet is out of luck. A site with a ten character URL (a normal length for a website) with characters in [a-z, A-Z, 0-9] resides in a namespace of eight hundred thirty nine quadrillion (8.39 * 10^{17}). After being compressed to one of *bit.ly*'s shortened URLs it is now in a namespace of 5.78 * 10^{10} with a high 42% density. This means that it is now feasible to scan the compressed URL space. Vitaly Shmatikov's paper demonstrated that by searching the URL namespace, it is easy to find Microsoft One-Drive account URLs. Approximately 7% of these accounts have write privileges, meaning an attacker can upload malware onto all of the account's devices [5]. The use of shortened URLs was built into this product by default without user's choice. This eventually exposed the users as easy targets. Microsoft said this behavior was not a security flaw, but eventually closed the loophole [8].

4.2 MALWARE HOSTING

The *Bitly* service allows the shortening of any valid URL. As the old identity of the URL is masked by the shortened link and that link is indistinguishable from other short links, opening the URL can lead anywhere. In 2009 Google's security blog posted a list of the current ten most popular malware sites [9]. Submitting these links to bit.ly gives some interesting results. Three of the links have already been shortened, and some statistics about the number of page hits the link has generated is available [Figure 2].

Malware Site	Compressed URL	Number of Hits
38zu.cn	http://bit.ly/PNA2ih	75
lousecn.cn	http://bit.ly/1uwqcD8	1
gumblar.cn	http://bit.ly/1mbeg55	25

[Figure 2]

An attacker has already decided to "shorten" the malware hosting site 38zu.cn, and *bit.ly* informs us that 75 hits occurred through the malicious *bit.ly* link.

Shortened URLs present an opportunity for phishing and social engineering. Often normal links are too long and unwieldy to be sent in email, as they can be chopped off in the message by line breaks. *Bitly* has shortened over 29 billion links, encountering them in an email is not unusual anymore. Someone who clicks a *bit.ly* link and then goes to *www.sis.uit.tults.edu/psp/paprd/EMPLOYEE/EMPL* versus *www.sis.uit.tufts.edu/psp/paprd/EMPLOYEE/EMPL* and sees an attacker's website that looks exactly like the one they expect is not going to notice a difference (a security minded user paranoid enough to check and verify the URL would probably not have clicked the shortened link in the first place).

The Skype Goo.gl virus, a virus that hit Skype in mid-2016, spread through shortened URLs. A compromised user's account would send direct messages to other users on the account's contact list. The receivers of the messages would see a shortened URL from a contact. The link contained malware, and if clicked the receiver of the link would also be hacked [10].

Malicious attackers have also created fake shortened *bit.ly* links with the domain name of *bt.ly*. Theses links generally have a token that consists of a video game name instead of the normal short random appearing token. The links were distributed with the promise of cracked games available for download, but hosted malware instead [11].

4.3 Twitter's Approach

The simplest approach to fixing the issues associated with shortened URLs is to not use them. Unfortunately this is too much to ask of people. Another approach is to build a browser extension that pauses before letting a user visit a shortened URL, scans the link for malware, and then reports back to the user. This prevents an issue with user experience, as people who click on the shortened links will have to wait for the results of the scan (might be a long time), providing a negative browsing experience. Often the scan will say the link is fine, and then the user could get into the habit of just skipping the scan or disabling the extension. When the choice comes between usability and security, most people drop security. A different approach to fixing the problem is the shortening services themselves check the submitted links for malware, and block malware sites from having a shortened URL. However this places a burden on the service provider as extra computation is now required and the shortened link will not be generated as quickly (potential loss of revenue).

Twitter has a URL shortening service of its own with an interesting approach to security. Tweets are limited to 140 characters, so to allow users to tweet URLs Twitter uses the *http://t.co* domain for shortening. This shortener is private to Twitter, the only way links can be shortened are by sharing the link with Twitter. Twitter acknowledges the fact that anyone with the *t.co* link can visit the URL and provides no solution for the privacy issue of URL shortening. Upon receiving a link Twitter's service scans it for malware and if any is found the link is flagged. When users try to access the link a warning appears [12]. Because all sites get scanned, this approach helps solve the problem of hosting malware on shortened URLs.

5 Summary

URL shorteners leak information and pose a security risk by potentially hiding malware. Scraping the compressed URL space is entirely feasible and requires an insignificant amount of computation [5], making the information associated with the URLs publicly accessible. Some compressed URLs contain malware; these are indistinguishable from the safe links.

The best course of action is to never click on the shortened URLs. However if one has to click on a shortened URL, always unshorten the link first and confirm the website is as expected (*bitly* lets one do this by adding a '+' to the end of a link). Additional assurance can be gained by scanning the decompressed URL with a virus scanner (such as VirusTotal) before visiting the site. Never use a URL shortener with sensitive information (family photos, private documents) as these can be easily found. Finally, when designing URL shorteners for use in a production system use an approach like Twitter's. These recommendations presented could help fix an issue that is getting larger every day.

References

[1] https://bitly.com/pages/enterprise
[2] http://vanityurlshorteners.com, 2015
[3] NEUMANN, A., BARNICKEL, J., AND MEYER, U. Security and Privacy Implications of URL Shortening Services. In W2SP (2011).
[4] https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
[5] http://www.cs.cornell.edu/~shmat/shmat_urls.pdf
[6] http://support.bitly.com/customer/en/portal/articles/1765837-does-bitly-ever-re-use-links-?b_id=5612
[7] http://thenextweb.com/insider/2014/10/25/chrome-firefox-flag-bit-ly-links-malware/
[8] https://freedom-to-tinker.com/2016/04/14/gone-in-six-characters-short-urls-considered-harmful-for-cloud-services/
[9] https://security.googleblog.com/2009/06/top-10-malware-sites.html
[10] http://www.securityspyware.com/skype-goo-gl-virus-remove/
[11] https://blog.malwarebytes.com/cybercrime/2015/05/bitly-imitation-leads-to-malware-download/
[12] https://support.twitter.com/articles/109623