Two Years of Short URLs Internet Measurement: Security Threats and Countermeasures

Federico Maggi, Alessandro Frossi
Stefano Zanero
Politecnico di Milano
{fmaggi, frossi, zanero}@elet.polimi.it

Gianluca Stringhini, Brett Stone-Gross
Christopher Kruegel, Giovanni Vigna
UC Santa Barbara
{gianluca, bstone, chris, vigna}@cs.ucsb.edu

ABSTRACT

URL shortening services have become extremely popular. However, it is still unclear whether they are an effective and reliable tool that can be leveraged to hide malicious URLs, and to what extent these abuses can impact the end users. With these questions in mind, we first analyzed existing countermeasures adopted by popular shortening services. Surprisingly, we found such countermeasures to be ineffective and trivial to bypass. This first measurement motivated us to proceed further with a large-scale collection of the HTTP interactions that originate when web users access live pages that contain short URLs. To this end, we monitored 622 distinct URL shortening services between March 2010 and April 2012, and collected 24,953,881 distinct short URLs. With this large dataset, we studied the abuse of short URLs. Despite short URLs are a significant, new security risk, in accordance with the reports resulting from the observation of the overall phishing and spamming activity, we found that only a relatively small fraction of users ever encountered malicious short URLs. Interestingly, during the second year of measurement, we noticed an increased percentage of short URLs being abused for drive-by download campaigns and a decreased percentage of short URLs being abused for spam campaigns. In addition to these security-related findings, our unique monitoring infrastructure and large dataset allowed us to complement previous research on short URLs and analyze these web services from the user's perspective.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Web-based services; C.2.0 [**Computer Communication Networks**]: General

Keywords

Security; Short URLs; Measurement; Crowdsourcing.

1. INTRODUCTION

Since 2001, a number of URL shortening services have made their appearance on the Web. Users can submit URLs to such services to create aliases (short URLs) that are easier to share than the original URLs. A shortening service will keep an association between the original URL and the alias, and will redirect accesses to the short URL to the original page. Short URLs are commonly used to save valuable characters on services that impose strict length limits (e.g., Twitter). Users have grown accustomed to following a URL that looks like `http://bit.ly/lhBa6k`, even when the mapped URL may be `http://evil.com/attack?id=31337`. If it is usually difficult for a user to determine whether a URL is legitimate or not just by looking at it, this is even harder in case of short URLs. As a result, shortening services have been abused by miscreants for masquerading the true URLs of phishing or drive-by-download pages [5, 10, 12]. Large services such as Twitter, Facebook or YouTube have started running their own shortening service, upon which their social networks rely (e.g., `t.co`, `fb.me`, `youtu.be`). Unfortunately, when the hyperlinks of an entire social network rely upon one, single URL "translator", speed and availability also become of concern[1] (similarly to what happens with the DNS service).

To the best of our knowledge, there has never been a large-scale and global measurement study of the *threats to users* introduced by short URLs and the *countermeasures adopted* by shortening services. Previous work highlighted the security and privacy issues related to the rise of short URLs [7, 11], whereas the effectiveness of existing protections adopted by the services was not analyzed. In this work, we first assess whether such countermeasures can substitute blacklist-based protections implemented in current browsers, so that users can actually trust URLs exposed by popular shortening services even when client-side defenses are not in place. According to our experiments, popular services react against attempts of shortening long URLs that expose malicious behavior at the time of submission—by either banning offending URLs or by displaying warning pages; however, from our preliminary experiments we noticed that shortening services do not check existing short URLs periodically (see Tab. 3). Such checks are useful in case the aliased landing pages turn malicious (e.g., after a timeout expiration). The only exception is tinyurl.com, which deleted 1,806 (spam) short URLs that became active *after* the short URLs were created.

In addition, previous work did not consider the *end users* and how they typically access pages containing short URLs. The research described in [2] analyzed the typical referrers, content popularity, geolocalization, and longevity of bit.ly and ow.ly short URLs, collected via Twitter and exhaustive enumeration (i.e., `ow.ly/[a-Z0-9]+`). We observe the short URL phenomenon from a completely different perspective,

[1] `http://www.theverge.com/2012/10/9/3477734/twitter-outage-phishing-complaint`

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

which allows us to obtain more detailed information about the usage of short URLs. More precisely, instead of directly crawling short URLs found on web pages, we "crowdsource" the collection to a large pool of real web users. To this end, we developed and deployed a publicly-available web service providing a much-needed feature, that is, a preview of a short URL's landing page. While browsing the Web, our users submitted 24,953,881 distinct short URLs to our servers automatically, via browser add-ons. Although the users in our dataset rarely stumbled upon malicious short URLs, we found some patterns that characterize malicious short URLs: The use of multiple short URLs that point to the same malicious URL is quite common; in contrast, benign URLs are typically aliased less frequently. Also, malicious short URLs remain active longer than benign short URLs. More precisely, we noticed that the difference in time between the first and latest appearance of a malicious short URL—regardless if it migrates intermittently across several pages—is longer than that of benign short URLs.

In summary, this paper makes the following contributions:

- We analyzed the countermeasures adopted by URL shortening services, and show how these are not adequate against pages that expose their malicious content later in time.
- We conduct the first user-centric collection of short URLs that includes the vast majority of shortening services, comprising more than 7,000 real web users.
- To the best of our knowledge, we are the first to broadly analyze the impact of malicious short URLs on users. Thus, our measurements are a complement toward the understanding of how short URLs are used.
- We assess whether there are typical usage patterns that cybercriminals may leverage to drive their campaigns via short URLs. For this, we use our dataset to calculate global rankings and statistics about usage patterns, along with a fine-grained categorization of websites that contain both short and long URLs.

2. CURRENT COUNTERMEASURES

Our first goal is to understand what (if any) measures are taken by shortening services to prevent malicious URLs from being shortened and, if they are shortened, the amount of time it takes for such URLs to be flagged and removed. To this end, we submitted three types of malicious URLs to the most popular short URL services that had a public API. More specifically, we submitted 10,000 URLs (2,000 for each of the five shortening services examined), picked randomly, among those that were recently submitted to Wepawet and that delivered drive-by-download exploits targeted at vulnerabilities in web browsers and browser plugins (e.g., Adobe Flash, Java). In addition, we submitted 10,000 phishing URLs that were online and tracked by PhishTank, and 10,000 URLs that were recently observed in spam emails that we obtained from Spamhaus. The purpose of examining three types of URLs was to determine whether URL shortening services block one or more classes of threats. After submitting the URLs, we first recorded whether the shortening service allowed us to shorten the URL. Then, if the service shortened the URL, we tracked whether the corresponding short URL could be expanded on a daily basis for a four week period.

In addition to the URLs mentioned above, we also submitted 10 URLs of each type that we manually reviewed to ensure that they were actually still delivering live exploits at the time of submission, as it is common that a malicious URL, once discovered, is brought to the attention of a site administrator and removed. We ran the analyses discussed in the remainder of this section on both the large and small set of URLs; as the results we obtained were consistent, we present the results obtained for the larger dataset. An overview of the results of our measurements is shown in Tab. 1. Interestingly, the most popular service, bit.ly, accepted all the malicious URLs we submitted. Among the services that employs countermeasures, is.gd is particularly effective against spam, as it prevented the vast majority of spam URLs that we submitted from being shortened., while migre.me seems to perform some kind of phishing filtering on submitted URLs.

The situation changes significantly when looking at the warnings that are displayed when short URLs are accessed (expanded), as shown in Tab. 2. Overall 2,049 shortened malicious URLs were blacklisted after the submission by these services (about 21.45% of the 9,551 that passed the submission). Here, bit.ly covers a significant fraction of all malicious URLs: It indeed expands a short URL unless it believes the target is malicious. Overall, all services had quite effective spam URL detection systems. We were also rather surprised that goo.gl, in late 2010 when we were able to test it, was not as effective at blocking malware and phishing as (at least) Google's own blacklist.

Summary: bit.ly was the only one that flagged almost all malicious URLs that we shortened, although we recall that they were all accepted for shortening with no checks upon

Service	Malware		Phishing		Spam	
	#	%	#	%	#	%
bit.ly	2,000	100.0	2,000	100.0	2,000	100.0
durl.me	1,999	99.9	1,987	99.4	1,976	98.8
goo.gl*	2000	99.9	994	99.4	1,000	100.0
is.gd	1,854	92.7	1,834	91.7	364	18.2
migre.me	1,738	86.9	1,266	63.3	1,634	81.7
tinyurl.com	1,959	99.5	1,935	96.8	587	29.4
Overall	9,550	95.5	9,022	90.2	6,561	65.6

Table 1: Number and percentage of malicious URLs that were accepted for shortening by the top services. Overall, 25,133 malicious URLs were accepted for shortening, accounting for about 83.78% of the 30,000 submitted.

Service	Malware	Phishing	Spam
bit.ly	0.05	11.3	0.0
durl.me	0.0	0.0	0.0
goo.gl*	66.4	96.9	78.7
is.gd	1.08	2.27	0.8
migre.me	0.86	14.0	0.0
tinyurl.com	0.66	0.7	2.04
Overall	21.45	26.39	31.38

Table 2: Shortened malicious URLs expanded without warnings when accessed. (*) We tested goo.gl in late 2010, whereas the results for the remainder shorteners are up to date (late 2011), when Google introduced a CAPTCHA that prevented automated submissions.

submission. On the other hand, is.gd prevented the majority of malicious URLs from being shortened when submitted—probably using lightweight, blacklist-based checks.

2.1 Deferred Malicious URLs

We measured whether shortening services retroactively analyze malicious URLs, that is, we determined if these services perform any repeated verification of the safety of the long URLs to which their existing short URLs point to. Thus, we set up a web page that served benign HTML content for a period of three days; this page's URL contained 32 random characters to ensure that no legitimate users accidentally stumbled upon the URL. After the third day, we modified our page to redirect to a web site serving non–legitimate content (i.e., malicious, spam or phishing content). We discovered that all the shortening services that we tested did not detect the previously-benign page that we modified to redirect visitors to a malicious site. Surprised by this finding, we set up 3,000 distinct web pages hosted at random domains and URLs and fed each service with all of them, totaling 1,000 distinct short URLs per service for a total of 15,000 short URLs overall. After 72 hours we modified each page so it redirected to a malicious site. More precisely, we used 1,000 unique URLs serving drive-by exploits, 1,000 phishing pages, and 1,000 spam URLs. In other words, after 72 hours, the short URLs became active aliases of the set of 3,000 malicious URLs. We monitored the redirection chain of each short URL on a weekly basis to determine which shortening services displayed an alert or blocked the redirection—as a result of a security check performed *after* the shortening.

From our results in Tab. 3 we notice that only 20% of the malicious URLs were blocked by the shortening service when we accessed them after they became malicious—this 20% is actually due to the fact that durl.me *always* displays a preview for a short URL, regardless of whether the URL is benign or malicious, which is by not a very effective security mechanism. The other services, however, did not block any malicious short URL, neither at submission time nor after they were modified.

Summary: The most popular shortening services verify the URLs only upon submission, and an attacker can evade this check by shortening a benign URL that will begin to redirect to a malicious page a few moments later. As we further elaborate in §7, believe that URL shortening services should periodically sanitize past short URLs, so that benign pages turning malicious can be detected. Clearly, this is not an easy task as it presents the typical challenges of client-side threat analysis (e.g., cloaking, fingerprinting, evasion) [9].

Threat	Shortened	Blocked	Not Blocked
Malware	5,000	20%	80%
Phishing	5,000	20%	80%
Spam	5,000	20%	80%
Overall	15,000	20%	80%

Table 3: Deferred malicious short URLs submitted and percentage of blocked versus not blocked ones. The 20% is simply due to the fact that durl.me displays a default warning for any URL benign or malicious; as a result 1,000 out of 5,000 URLs per threat category are naïvely "blocked" by this mechanism.

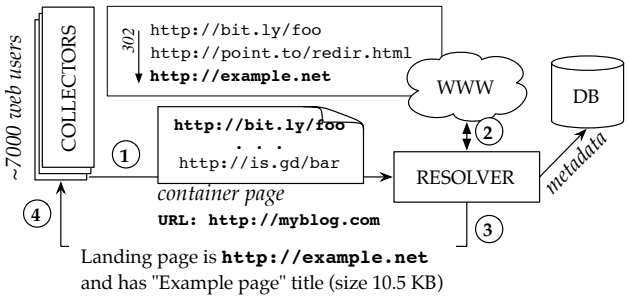


Figure 1: Overview of our collection approach.

We conducted the experiments described in this section in April 2011 and repeated them in April 2012. Alarmingly, we obtained statistically-similar results, showing that none of the shortening services changed their security measures against dangerous short URLs.

3. GOALS & MEASUREMENT APPROACH

Alarmed by our security assessment discussed in Section 2, we wanted to analyze short URLs at a larger scale, to understand how they are used with malicious intents. Unlike previous work, we concentrate on the clients' perspective, so that we can characterize also the usage habits: To this end, we collect short URLs while clients access web pages that contain short URLs. Moreover, we do not limit our analysis to a selection of shortening services (e.g., the most popular ones) nor narrow our attention on short URLs published on a few, specific online social networks and news aggregators (e.g., Twitter). Instead, we cover a wide variety of URL shortening services, up to 622, whose URLs appear in thousands distinct websites.

Our collection system comprises a browser add-on (named "collector") and a centralized short URL resolver. The *collector* analyzes container pages while the user browses the Web. The *container page* is a web page that, when rendered on a browser, displays or contains at least one short URL. Each collector submits short URLs to our resolver, along with a timestamp, URL of the container page, and client IP address. The *resolver* finds the respective *landing page*.

Our add-on works on Google Chrome, Mozilla Firefox, Opera[2], and any browser that support JavaScript-based add-ons. When the browser renders a page, the add-on searches its DOM for short URLs and submits them to our resolver along with the container page URL. The add-on instantiates contextual tooltips associated to each short URL submitted. These tooltips are revealed whenever the mouse cursor hovers on a resolved short URL. The tooltip displays details about the landing page (e.g., URL, size, title, and content type). Users can also contribute by reporting suspicious short URLs by clicking on a "flag as suspicious" link on the tooltip. This action is recorded in our database.

For each short URL received, our resolver obtains the landing page URL, title, size, and content type (e.g., HTML, XML, image), by visiting each short URL with a mechanized browser that follows and tracks redirections. When the re-

[2]We released our add-on after approval from the Mozilla community. The code base is the same, although we had to adapt it to the various browsers with minor adjustments.

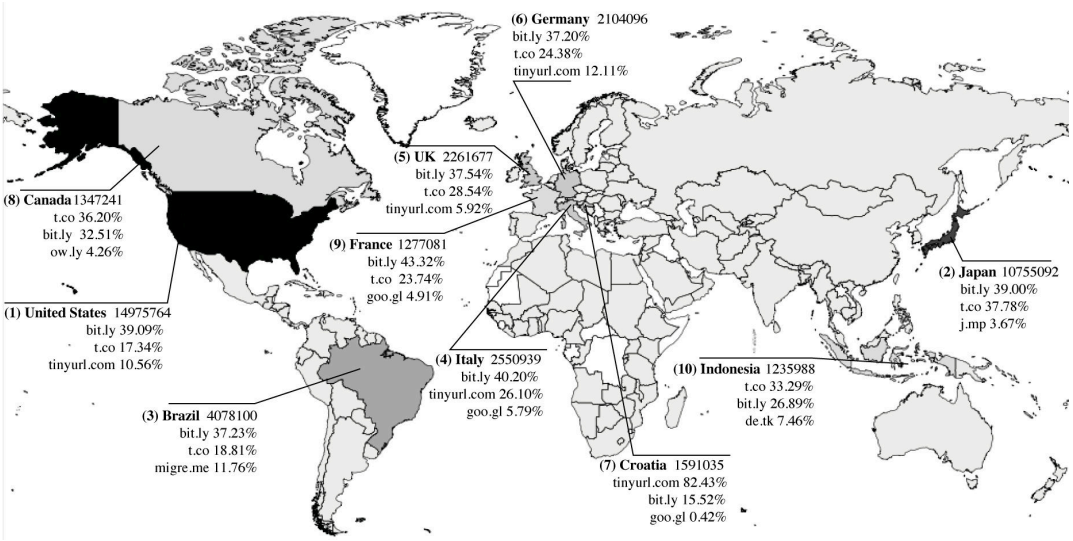


Figure 2: Contributors' geographical location.

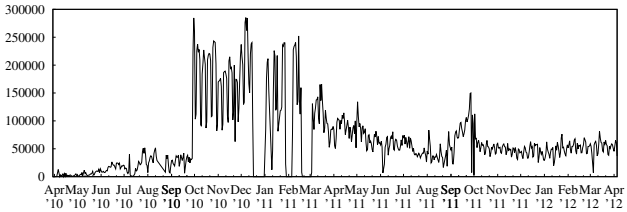


Figure 3: Log entries per day between late March 2010 and April 2012.

Distinct URLs		Log entries	
10,069,846	bit.ly	24,818,239	bit.ly
4,725,125	t.co	12,054,996	t.co
1,418,418	tinyurl.com	5,649,043	tinyurl.com
816,744	ow.ly	2,188,619	goo.gl
800,761	goo.gl	2,053,575	ow.ly
638,483	tumblr.com	1,214,705	j.mp
597,167	fb.me	1,159,536	fb.me
584,377	4sq.com	1,116,514	4sq.com
517,965	j.mp	1,066,325	tumblr.com
464,875	tl.gd	1,045,380	is.gd

Table 4: The 10 most popular services ranked by number of log entries in our database, and number of distinct short URLs collected. Highlighted rows indicate services at the same rank.

solver receives an HTTP 200 response, it assumes that the landing page has been reached and no further redirections follow. The resolver then extracts the relevant data from the landing page's source code and saves the redirection chain. In addition, we store the collectors' IP addresses for aggregation purposes. The completion of the whole procedure may take up to a few seconds, depending on network conditions and responsiveness of the servers that host the landing and intermediate pages. For this reason, we deployed 100 load-balanced, parallel resolvers along with a caching layer (that stores results for 30 minutes) that ensures short response times. According to the measurements reported in [2], a short URL random suffix, which is its identifier, takes much

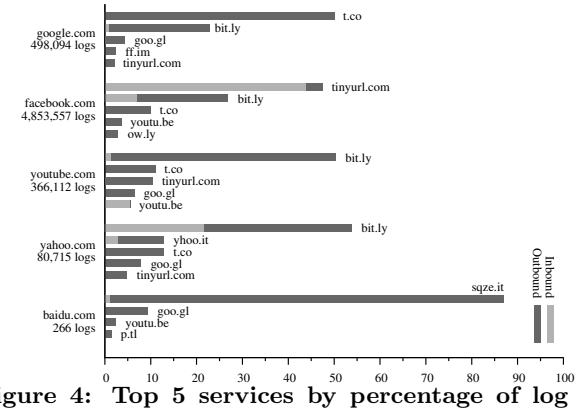


Figure 4: Top 5 services by percentage of log entries of outbound short URLs of Alexa top 5 domains. Light gray bars are the portion of logs of outbound short URLs contained that redirect to pages within the same domain (i.e., self loops).

longer to expire and get recycled. When the cache expires, we retain a snapshot of the log entry (i.e., the history of each short URL).

In summary, our service takes short URLs as input from clients and returns the aliased landing page. These "expansion" services have become useful for previewing the actual websites behind short URLs. The long time span of our measurement and the usefulness of our service—which is free of charge and publicly available through popular browser add-on marketplaces—allowed us to collect a unique, large dataset.

3.1 Measurement

l We deployed our data collection infrastructure in March 2010 and, as of April 2012, our database contained 24,953,881 unique short URLs. More than 7,000 web users downloaded and installed our browser add-on and submitted short URLs; some users also contacted us and requested that we add support for additional shortening services. Around 100 out of the 622 that are currently supported by our system were

suggested by users. Our collection infrastructure receives data from 500 to 1000 active users every day. We store a record in our database for each short URL submitted. We refer to such records as *log entries*. Each log entry contains the source (geo-localized) IP address, the container page and landing page URLs, and the timestamp. Thus, each log entry corresponds to one short URL found in a given container page, and represents the fact that a user viewed the container page at a given time. We never retain identifying information possibly related to the specific user who submitted a log entry.

3.1.1 Overall Dataset Statistics

Fig. 3 shows the daily number of log entries whereas Fig. 2 shows the contributors' geographical location. Albeit we deployed the system in March 2010, the vast majority of users became active contributors starting from Oct 2010. However, at its humble beginnings our system received 20,000 to 50,000 log entries per day. At steady usage rates, we store an average of 90,000 log entries per day. Each of the 1,649,071 distinct IPs contributed around 37 requests on average, with a standard deviation of 7,157. Distinct IPs may not correspond to distinct users, either because multiple users could share the same sets of IPs (e.g., via NATs) or because of dynamic IP-assignment policies employed by ISPs. Our system experienced three outages due to database failures throughout one year: in late December 2010, in late January 2011, and between late February and March 2011. Nevertheless, we collected a large amount of short URLs useful to conduct the analysis described in the remainder of this paper.

Before analyzing security aspects of short URLs, we describe our dataset through four aggregated statistics: (1) distinct short URLs, (2) log entries in our database, (3) log entries of *inbound* short URLs (distinct short URLs pointing to the sites' pages), and (4) *outbound* short URLs (short URLs that are found in their container pages and that point to both external and internal pages). Shortening services with many distinct short URLs are more popular (i.e., they have become the "shortener of choice" for several users), whereas those characterized by many log entries have their short URLs posted on many popular container pages. As shown in Tab. 3 the top most popular services in our dataset are bit.ly, t.co and tinyurl.com, respectively. As expected, popular shortening services hold a steadily large number of short URLs, whereas site-specific shortening services exhibit a behavior that is typical of content shared through social networks. Fig. 4 shows the ranking of the top websites with respect to inbound and outbound short URLs.

4. RESULTS AND DISCUSSION

Our objective is to assess if malicious short URLs have distinctive features (§4.1) and typical usage patterns (§4.2) that criminals may leverage to target their campaigns.

4.1 Malicious Short URLs

First, we wanted to understand how frequently the users in our database encounter malicious short URLs in a page. For this, we leveraged four datasets: the Spamhaus DBL, a list of DNS domains that are known to host spam pages, Wepawet, a service able to detect drive-by-download exploit pages, Google Safe Browsing, a list of domains known for hosting malware or phishing sites, and PhishTank, a blacklist of URLs that are involved in phishing operations. For Spamhaus, we checked the domain against the database. For the other three blacklists, we checked the full URL. We break the landing URLs into three classes: spam, phishing, and malware, according to the dataset they were flagged in: URLs detected by Wepawet are flagged as malware, domains found in Spamhaus are marked as spam, and URLs from PhishTank as phishing. Google Safe Browsing classifies both phishing and malware sites. Tab. 4.1 summarizes the breakdown of malicious short URLs.

Category	Short URLs	Long URLs	Ratio
Phishing	3,806	920	4.1
Malware	27,203	8,462	3.2
Spam	13,184	10,306	1.2

Blacklist	Phishing	Malware	Spam
Spamhaus	-	-	10,306
PhishTank	7	-	-
Wepawet	-	6,057	-
Safe Browsing	913	2,405	-

Table 5: Number of short and long URLs, respectively, classified as Phishing, Malware, and Spam. The dash '-' indicates that the blacklist in question provides no data about that threat.

We observed 44,932 unique short URLs pointing to 19,216 malicious landing pages. By looking at the referrer, these URLs were hosted on 1,213 different domains. We provide a more detailed analysis on the container pages of malicious URLs in the next section. In total, the malicious URLs in our dataset have been rendered by 1,747 users in their container pages via our browser add-ons: 378 users (about 21.6%) were located in South Korea, 282 (about 16.1%) in the United States, and 98 (about 5.6%) in Germany.

Unsurprisingly, bit.ly is the top most common service, serving 10,392 malicious short URLs, followed by tinyurl.com with 1,389, and ow.ly with 1,327. As a side result, we also measured whether users perceive and report malicious short URLs, and found out that only 2,577 distinct short URLs have been signaled as malicious through our browser add-ons. Only 2 of these URLs were actually malicious according to at least one of the aforementioned blacklists.

4.1.1 Dissemination of Malicious Short URLs

We then analyzed how malicious pages are aliased through short URLs, and whether this trend changed over time. During the first year of our analysis, multiple short URLs were sometimes used to point to the same malicious page, although the average ratio was low. About 2.01 distinct short URLs were used to alias a malicious landing page, whereas we observed an average of 1.3 distinct short URLs per distinct *benign* landing page. Looking at a 2-year span period, however, those average numbers became very similar: about 1.17 unique short URLs per malicious page versus 1.14 unique short URLs per benign page. This comparison is better explained in Fig. 5(a) and Fig. 5(b), which show the empirical cumulative distribution function for the ratio of short URLs per landing page URL of legitimate vs. malicious pages of the two periods. The pattern here is that benign URLs used to have, in general, less short URLs pointing to them when compared to malicious URLs. Interestingly, in the second year of our measurement, the situation changed slightly. In

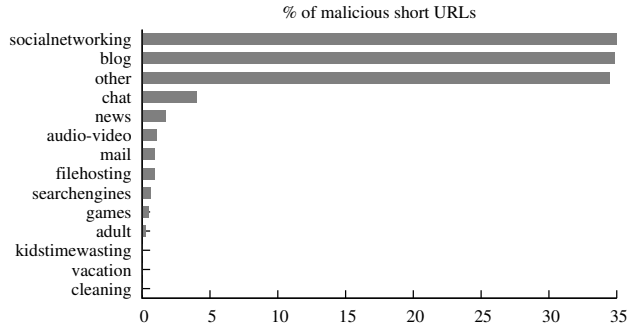


Figure 6: Malicious short URLs: Categories of container page ranked by the amount of short URLs they held. We assigned pages to categories as described in §4.2.

particular, as shown in Fig. 5(b), the practice of using multiple short URLs pointing to the same spamming long URL is less used than in the past (Fig. 5(a)), where the aliasing of spam URLs was more evident.

We then analyzed the frequent container pages abused to publish malicious short URLs through the HTTP requests' `referer` issued while expanding the URLs (9,056 of these had a referrer specified).

As summarized in Fig. 6, the majority of those URLs were found on social networks. More precisely Twitter accounted for 5,881 URLs, 30% of the total. In second position there is Facebook—228 requests, accounting for 1.18% of the total. The third most common referrer is a Belgian news site with 137 requests, accounting for 0.7% of the total. We suspect that this website was victim of massive comment spam. It is also interesting to look at which sites, among those containing malicious short URLs, attracted the most number of users. Twitter is in first position, with 104 potential victims, followed by Facebook with 31, and by a hacking forum with 27 distinct IP addresses visiting it. This forum is probably another example of comment spam. However, these container pages, which are the most targeted ones, do not contain many short URLs, as detailed in Fig. 8: We can argue that the cyber criminals are not considering the "density" of short URLs per container page, but rather its popularity.

4.1.2 Lifespan of Malicious Short URLs

In the previous section we analyzed whether the dissemination of short URLs exhibits different characteristics between malicious and benign content, whereas in this section we compare them by means of timing patterns. We derived the maximum lifespan of each collected URL based on historical access logs to their container pages. We calculated the *maximum lifespan* (or simply lifespan) as the delta time between the first and last occurrence of each short URL in our database. More specifically, our definition of lifespan accounts for the fact that short URLs may disappear from some container pages and reappear after a while on the same or other container pages. Fig. 7 shows the empirical cumulative distribution frequency of the lifespan of malicious versus benign short URLs. About 95% of the benign short URLs have a lifetime around 20 days, whereas 95% of the malicious short URLs lasted about 4 months. For example, we observed a spam campaign spanning between April 1st and June 30th 2010 that involved 1,806 malicious short URLs redirecting to junk landing pages; this campaign lasted about three months until removed by tinyurl.com administrators. The Message-

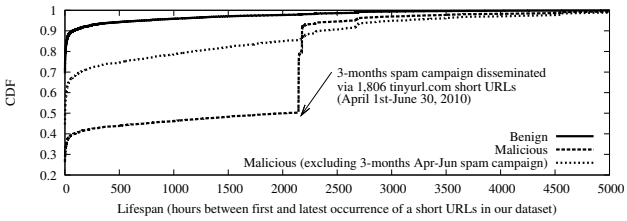


Figure 7: Delta time between first and latest occurrence of malicious versus benign short URLs. The "peak" indicates a high, about 50 %, amount of spam short URLs that lasted about three months. Malicious URLs are usually found on multiple different container pages even for extended periods of time, whereas benign short URLs follow the "one-day-of-fame effect" pattern.

Labs Intelligence Annual Security Report [1] for that year corroborates our findings: The Storm botnet, which made a significant reappearance in April 2010, seems to be the culprit of this massive spam campaign that contains several shortened URLs.

For the sake of clarity, we removed short URLs involved in such spam campaign from the second dashed curve in Fig. 7; nevertheless, we notice that malicious short URLs last longer than benign URLs, in general. Recall that each short URL may have different container pages at the same point in time, and these can vary over time. Also recall that the longevity of short URLs on each container pages is quite low, as observed in [2] by Antoniades et al. A short URL can make its first appearance on a certain page, disappear to make room for new pages, and reappear a few moments later (even) on different container pages. From this observation, we can argue that, from the miscreants' point of view, the lifespan as we calculate it—across different container pages—seems to be of more importance than the lifespan on a single container page. In fact, short URLs that have a longer lifespan—regardless if they migrate intermittently across several pages—have higher chances of receiving visits from a large audience while remaining stealthy even months after publication. However, those URLs that survive on very popular pages only for a few hours may have their one day of fame before disappearing or being deleted by the container page administrators.

As detailed in §5, we do not track clicks on short URLs. Nevertheless, our collection method ensures that short URLs are tracked as soon as they appear the web pages visited by the users in our large pool. This ensures us good visibility

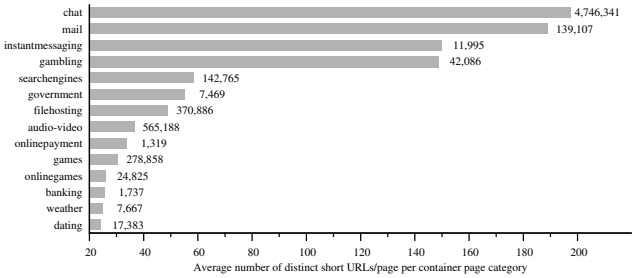
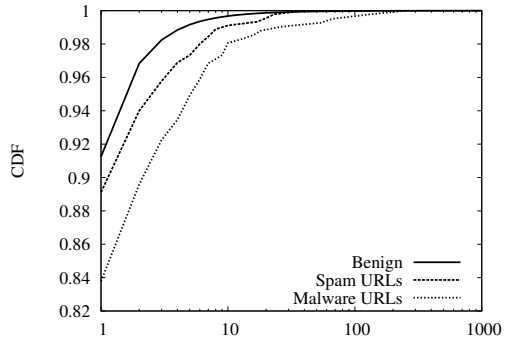
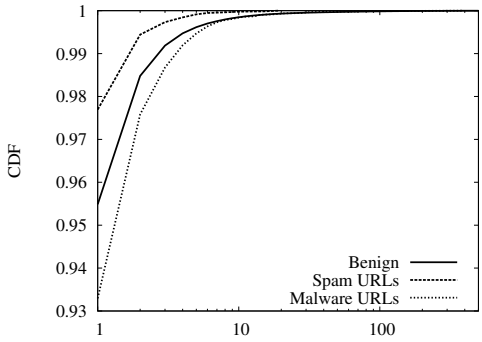


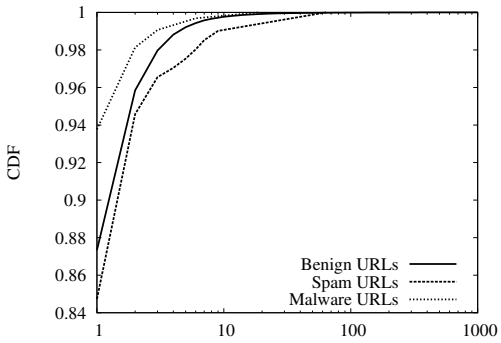
Figure 8: Categories of container page ranked by the average number of short URLs/page they held. The total number of distinct short URLs is also shown.



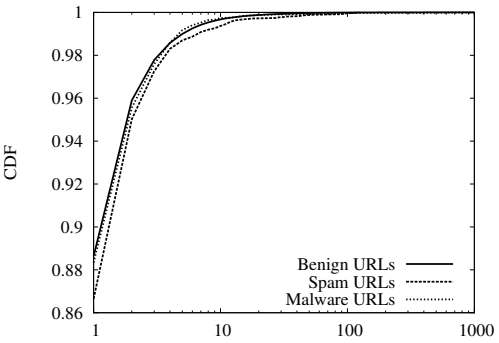
(a) Distinct short URLs per distinct malicious or benign landing URL from April 2010 to April 2011.



(b) Distinct short URLs per distinct malicious or benign landing URL from April 2010 to April 2012.



(c) Distinct containers per distinct malicious or benign short URL from April 2010 to April 2011.



(d) Distinct containers per distinct malicious or benign short URL from April 2010 to April 2012.

Figure 5: Comparison of the number of distinct short URLs per unique landing page (a, c) and distinct container page per unique short URL (b, d) after 1 year (a, b) and after 2 years (c, d). The distribution have changed over time: (a) spam pages were generally aliased with a larger number of short URLs than benign pages; (b) in the following year spammers did not to alias their pages as much as before. Also, (c) spam short URLs used to be spread over a larger number of container pages than benign short URLs; however, (c) they now exhibit the same distribution as their benign counterpart. We may argue that short URLs are not seen anymore as a valuable mean of aliasing spam pages.

over their evolution. This is corroborated by the statistics about the abuse of short URLs found in latest three APWG reports [12–14]: After a growing trend, at the beginning of our measurement (2010), the subsequent reports highlight a stable (2011) and decreasing (2012) trend.

4.2 The Short URLs Ecosystem

As part of their research, Antoniades and colleagues in [2] have analyzed the category of the pages to which bit.ly and ow.ly short URLs typically point to, along with the category of the container page, that they had available for bit.ly URLs only. They assigned categories to a selection of URLs. We did a similar yet more comprehensive analysis by characterizing all the short URLs that we collected by means of the categories described in the following.

Categorizing an arbitrary-large number of websites automatically is a problem that has no solution. However, our goal was to obtain a coarse-grained categorization. To this end, we relied on community-maintained directories and blacklists. More precisely, we classified the container pages (about 25,000,000 distinct URLs) and landing pages (about 22,000,000 distinct URLs) using the DMOZ Open Directory Project (`http://www.dmoz.org`) and URLBlacklist.com. The former is organized in a tree structure and includes 3,883,992 URLs: URLs are associated to nodes, each with localized, regional mirrors. We expanded these nodes by recursively merging the URLs found in these mirrors. The latter complements the DMOZ database with about 1,607,998 URLs and domains metadata. URLBlacklist.com is used by web-filtering tools such as SquidGuard (`http://www.squidguard.org`) and contains URLs belonging to clean categories (e.g., gardening, news), possibly undesired subjects (e.g., adult sites), and also malicious pages (i.e., 22.6% of the sites categorized as "anti-spyware", 18.15% of those categorized as "hacking", 8.29% of pages falling within "searchengine" domains, and 5.7% of the sites classified as "onlinepayment" are in this order, the most rogue categories according to an analysis that we run through McAfee SiteAdvisor [3]). Overall, we ended up with 74 categories. For clearer visualization, we selected the 48 most frequent categories. These include, for example, "socialnetworking," "adult," "abortion," "contraception," "chat," etc. We reserved the word "other" for URLs belonging to the less meaningful categories that we removed, or for URLs

[3] `http://siteadvisor.com/sites/`

Service	Most freq.	%	Least frequent	%
bit.ly	News	23.56	Naturism	$8.3 \cdot 10^{-4}$
	Audio-video	10.62	Contraception	$7.7 \cdot 10^{-4}$
	Socialnet	9	Astrology	$1.6 \cdot 10^{-4}$
t.co	Audio-video	29.42	Naturism	$1.07 \cdot 10^{-3}$
	File-hosting	27.43	Anti-spyware	$8.89 \cdot 10^{-4}$
	News	17.48	Contraception	$1.78 \cdot 10^{-4}$
tinyurl	News	24.08	Contraception	$4.5 \cdot 10^{-3}$
	Audio-video	10.61	Naturism	$6.29 \cdot 10^{-4}$
	File-hosting	9.36	Childcare	$2.51 \cdot 10^{-4}$
goo.gl	News	19.10	Gardening	$3.34 \cdot 10^{-3}$
	Audio-video	12.23	Weapons	$1.69 \cdot 10^{-3}$
	Socialnet	11.65	Naturism	$1.69 \cdot 10^{-3}$
ow.ly	News	23.38	Contraception	$2.5 \cdot 10^{-3}$
	Socialnet	12.84	Childcare	$1.32 \cdot 10^{-3}$
	Audio-video	10.03	Naturism	$1.32 \cdot 10^{-3}$

Table 6: Most- and least-popular landing page categories for the top 5 shortening services. There is an overlap between categories, so percentages do not necessarily add up to 100%.

that remained unclassified. Note that each URL can belong to multiple categories.

4.2.1 Frequent and Infrequent Categories

Tab. 4.2.1 details the most and least frequent categories of the landing pages pointed to by short URLs of the top services. We notice that the five most popular services are used to refer to various categories including news, audio-video content, blog, and online social networks. However, the majority of short URLs collected come from user-authored content (e.g., online social networks, blog posts), mainly because these sites are very popular (e.g., Facebook). We also provide a different viewpoint by plotting the number of short URLs *per page* (Fig. 8).

We notice that short URLs are seldom used as aliases of reference, science, and health-related pages. A possible explanation could be that users may have somehow perceived that, as short URLs shall expire sooner or later, they are not reliable for spreading really important content (e.g., health). Secondly, and more importantly, users post short URLs in email and chat messages, weather sites, search-engine pages (including all `*.google.com` pages), do-it-yourself sites, and news pages. In summary, the majority of short URLs point to content that expires quickly. Therefore, from a security viewpoint, real-time countermeasures against malicious short URLs such as WarningBird [8] are of paramount importance and much more effective than blacklists. As detailed in Fig. 6, however, the categories that were most targeted by malicious short URLs in 2010–2012 are social networks and blogs.

4.2.2 Content-category Change

To understand how web pages are interconnected through short URLs, we analyzed whether clicking on a short URL brings the user to a landing page of a category that differs from the category of the container page (e.g., from a news website to a file-hosting website).

In Fig. 9, the top 50 shortening services are ranked by the median frequency of category change (plotted as a dot). More precisely, for each service and for each category, we calculated the fraction of short URLs that result in a "change of content category"—such fraction is then normalized by the

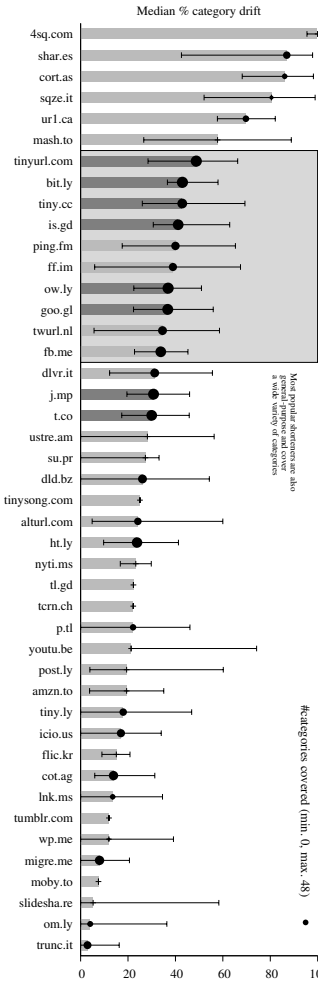


Figure 9: Frequency of change of category (median with 25- and 75-percent quantiles) and number of categories covered (size of black dot) of the top 50 services. The most popular, general-purpose shortening services highlighted are characterized by an ample set of categories (close to 48, which is the maximum) and short URLs that, in 32–48% of the cases, are published on pages having categories different from the landing page category.

total number of unique short URLs. Then, we derived the 25- and 75-percent quantiles to define a confidence interval around the median; this is useful to visually highlight how frequencies are distributed. Values close to 100% are not plotted for the sake of clarity. Services with 0–30% change frequency typically deal with a small set of categories and have short URLs often posted on websites of similar subjects. For example, flic.kr is used exclusively within the Flickr ecosystem; therefore, it covers very few categories and exhibits a very low change frequency, meaning that its short URLs are posted on websites that regard the same subjects, or even on Flickr directly. Another popular example is nyti.ms. On the opposite side, services with values above 50% also cover a small set of categories. However, differently from the first tier (i.e., 0–30%), we notice that the categories of the containers of these short URLs rarely match the categories of their landing pages. This is the case, for example, of 4sq.com

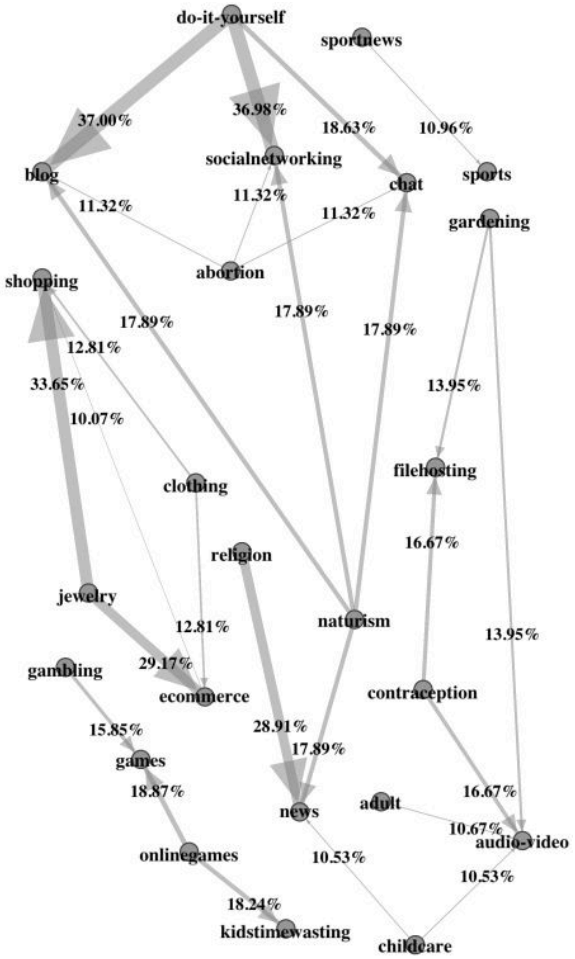


Figure 10: Digraph showing the connections between container- and landing-page categories. The edges' thickness expresses the frequency of finding short URLs between each nodes pair. For the sake of visualization, we removed edges with weight below 10 % and isolated nodes.

(about 100%), whose short URLs always bring from online social-networking sites to pages categorized as "other". The most popular shortening services (e.g., bit.ly, goo.gl, ow.ly) fall into the second tier (i.e., 32–48%), together with those services that cover a wide variety of categories, and typically interconnect pages of different categories. The most general-purpose services are those that are more abused to create aliases of malicious URLs: Here is indeed where we found the vast majority of malicious short URLs. Unfortunately, as we argument in §2, general-purpose shortening services rely on ineffective countermeasures.

4.2.3 Non-obvious Uses of Short URLs

We also analyzed how short URLs interconnect together pages of different categories, to understand whether some categories have a majority of container or landing pages. To this end, we calculated the average frequency of category change from the perspective of the container page and landing page. With this data we created a weighted digraph with 48 nodes, each corresponding to a category. The weights are the frequencies of change, calculated between each pair of categories—and normalized over all the short URLs and pages within each category. We then calculated the average weight of incoming, In(cat), and outgoing, Out(cat), edges for each category cat, and finally derive the ratio $\rho(cat) = \frac{\text{In}(cat)}{\text{In}(cat)+\text{Out}(cat)}$. When $\rho \to 0$, the category has a majority of outgoing short URLs (i.e., many container pages of such category), whereas $\rho \to 1$ indicates that the category has a majority of incoming short URLs (i.e., many landing pages of such categories). The digraph is shown on Fig. 10.

ρ	Category		
0.00	abortion	0.07	religion
0.00	antispyware	0.08	personalfinance
0.00	cellphones	0.10	gambling
0.00	childcare	0.14	government
0.00	contraception	0.16	medical
0.00	do-it-yourself	0.16	vacation
0.00	naturism	0.18	onlinegames
0.01	gardening	0.22	onlinepayment
0.01	hacking	0.22	sportnews
0.01	instantmessaging	0.30	searchengines
0.01	jobsearch	0.33	dating
0.01	pets	0.47	kidstimewasting
0.01	weapons	0.55	sports
0.02	artnudes	0.59	adult
0.02	drugs	0.60	games
0.02	jewelry	0.73	ecommerce
0.02	onlineauctions	0.78	shopping
0.02	weather	0.82	blog
0.03	mail	0.82	socialnetworking
0.04	banking	0.83	chat
0.04	cleaning	0.88	news
0.04	clothing	0.90	filehosting
0.06	drinks	0.92	audio-video
0.07	culinary	1.00	astrology

Table 7: Ranking of categories by the ratio of incoming and outgoing connections via short URLs.

As summarized in Tab. 7, there are clearly categories that exhibit a container-like usage, that is, they typically contain more outgoing short URLs than incoming short URLs. Besides a few extreme cases, which are mostly due to the scarcity of short URLs, container-like categories include, for instance, "onlineauctions," "mail" (web based emails contain outgoing short URLs more often than being referred to by short URLs), and "hacking."

Summary: Categories that we would anecdotally consider as aggregators (i.e., containers) of short URLs are actually more often used as landing pages. The most notable example is "socialnetworking" ($\rho = 0.82$), which we would expect to have many outgoing links as people share lots of resources through them. Instead, it turns out that, from a global viewpoint, this is no longer true. As expected, landing pages of a category with a high ρ (e.g., "socialnetworking", "blog", "audio-video") are the most obvious target of attacks: We indeed found that many short URLs that point to malicious resources have their landing page within these categories.

5. LIMITATIONS AND FUTURE WORK

Some research questions that need further answers include, for instance, to what extent shortening services have a vantage point in predicting trending topics. Indeed, shortening services are the very first services that receive important URLs from users, a few instants before a tweet is actually created.

Another limitation is that we collect short URLs when container pages are visited rather than when short URLs are visited. In addition to privacy concerns, in this initial work we decided to collect short URLs from visited container pages in order to collect a large amount of short URLs. Indeed,

we can realistically assume that users follow a subset of the short URLs contained in each page. Therefore, although we do not know exactly what short URLs are clicked by users, such short URLs are always considered in our measurement.

A technical limitation of our current implementation is that the mechanized browser used to resolve the landing page's URL acts as a normal browser would do, except for redirections implemented via (timed) JavaScript or Adobe Flash; these redirection mechanisms are not used by popular shortening services—except for ad-supported ones (e.g., `adf.ly`)—and are not reliably and efficiently supported by any of the publicly-available, mechanizable browsers. We experimented with scripting libraries such as Watir/webdriver, but they need about ten times the memory resources that our headless solution, which is faster and more reliable than Watir/webdriver.

6. RELATED WORK

We already mentioned the main points of [2] throughout this paper. The authors collected about 8.5M distinct short URLs by periodically crawling for bit.ly URLs on Twitter and by brute-forcing the key space of ow.ly. Although these services are the most popular, part of the statistics analyzed were actually calculated from the data offered by bit.ly, and thus may represent a service-centric view of the overall usage.

In [15] the authors analyzed a corpus of Twitter data (2006–2009) to discover patterns of word-of-mouth propagation of URLs among Internet and social network users. Short URLs were nearly 75% of the URLs on Twitter in 2009, when TinyURL and ow.ly were the top services. Back then, services linked to major social networks (e.g., t.co, fb.me) did not exist yet. Our distribution is consequently very different (see Tab. 3). In addition, they do not enumerate the shortening services, but rely on heuristics to identify them.

In [3] the authors checked how many phishing scams are posted on Twitter, and hidden behind short URLs. They leveraged PhishTank and the bit.ly API to perform their analysis. For this reason, their analysis is limited to a single service (although the biggest one). They found out that most phishing campaigns of this kind target social-network credentials rather than other services.

Klien and Strohmaier in [7] did a geographical analysis of short URLs, yet taking into account only a specific shortening service (qr.cx). Their database, however, as stated by the authors themselves may be biased in terms of location, user preferences and URLs content. Our dataset allowed us to calculate these and other types of aggregated statistics. An advantage of our approach is that we monitor all known shortening services, obtaining therefore less biased results.

In [11] the authors started from a list of shortening services widely used on Twitter and demonstrated that short URLs have implications both in terms of information disclosure (secrecy of the URL being shortened) and security. They assessed the possibility and easiness of enumerating short URLs from widespread shortening services, thus exposing URLs that the users may have wanted to keep secret. The authors also demonstrated that short URLs may also expose the user to security risks, such as hacking of the shortening services to change the redirection chain, or the possibility for the shortening service provider to leverage cookies to track the user. This analysis, however, mainly deals with implications inherent to shortening services design. Our work, instead, is entirely based on the user perspective and analyzes how the short URLs are used and what real security risks they pose.

On a dataset comprising 35 million distinct bit.ly URLs extracted from Twitter in Oct 2009, the authors of [6] observe that the quality of the landing pages is either high or very low. Frequently-tweeted URLs tend to be of very low quality. Although the authors do not detail the method used to detect spam short URLs, their conclusions disagree with our discussion in §4.1.2: According to their measurements, spam short URLs have shorter lifespan than those that point to clean content, whereas we observed the opposite. Recently, the Twitter stream was used as a source of URLs to develop and evaluate WarningBird [8], which analyzes the redirection chain of URLs and finds common "join points" (i.e., URLs that are shared by many redirection chains). The frequency of these join points is leveraged along with other features to train a supervised classifier that can tell malicious and benign URLs apart. The intuition is that these join points are limited in number and thus are easy to spot. This work is related to ours because it concentrates on short URLs, which cover the majority of URLs on Twitter. However, they focus on detecting malicious URLs in general, whereas we focus on assessing the threat level in the short URLs ecosystem. Similarly, [17] also focuses on detection of spam campaigns spread on social networks via (short) URLs. Among the 13 URL features extracted, the redirection chain is one of them (cfr., WarningBird).

Previous work also showed evidence about the abuse of short URLs. In [16] show that most spam campaigns leverage short URLs. Gao et al. in [4] focused on Facebook wall posts and analyzed the activity of about 3.5 million users and detected approximately 200,000 rogue posts (out of 187 million posts) with embedded URLs. The authors found short URLs in 10,041 posts, about 0.005% of the total posts analyzed. Although the authors collected a large amount of data, they concentrated on social networks, whereas our work has a broader viewpoint.

7. CONCLUSIONS

We have observed that, on a global scale, users are seldom exposed, while browsing, to threats spread via short URLs, or at least no more than they are exposed to the same threats spread via long URLs. Although we came across a relatively small number of malicious short URLs in the wild, we were able to evade the security measures currently adopted by the top shortening services to filter dangerous URLs, with a simple time-of-check to time-of-use attack. However, shortening services are not—and should not be—a definitive protection layer between users and malicious resources. In-the-browser defense tools such as blacklists can alert users before visiting malicious URLs, regardless of whether they are short or long URLs. Since it is very inefficient for shortening providers to monitor all their aliases periodically, we believe that this is not necessary when modern browsers are already prepared for counteracting known malicious URLs.

Acknowledgments. The authors are thankful to the reviewers, proofreaders, and to bit.ly, who provided a fast-track access to their API when needed. This work has been supported by the EU Commission through IST-216026-WOMBAT and FP7-ICT-257007 funded by the 7th FP. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the EU Commission.

8. REFERENCES

[1] P. W. e. al. MessageLabs Intelligence: 2010 Annual Security Report. Technical report, Symantec, 2010.

[2] D. Antoniades, E. Athanasopoulos, I. Polakis, S. Ioannidis, T. Karagiannis, G. Kontaxis, and E. P. Markatos. we.b: The web of short URLs. In *WWW '11*, 2011.

[3] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru. Phi.sh/$oCiaL: the phishing landscape through short URLs. In *CEAS '11*. ACM Request Permissions, Sept. 2011.

[4] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *IMC '10*, pages 35–47, New York, NY, USA, 2010. ACM.

[5] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *CCS '10*, pages 27–37, New York, NY, USA, 2010. ACM.

[6] V. Kandylas and A. Dasdan. The utility of tweeted URLs for web search. In *WWW '10*, pages 1127–1128, New York, NY, USA, 2010. ACM.

[7] F. Klien and M. Strohmaier. Short links under attack: geographical analysis of spam in a URL shortener network. In *HT '12*. ACM Request Permissions, June 2012.

[8] S. Lee and J. Kim. WarningBird: Detecting Suspicious URLs in Twitter Stream. In *NDSS '12*, 2012.

[9] B. Livshits. Finding malware on a web scale. *Computer Network Security*, 2012.

[10] D. K. McGrath and M. Gupta. Behind phishing: an examination of phisher modi operandi. In *LEET '08*, pages 4:1–4:8, Berkeley, CA, USA, 2008. USENIX Association.

[11] A. Neumann, J. Barnickel, and U. Meyer. Security and Privacy Implications of URL Shortening Services. In *W2SP '11*, 2011.

[12] R. Rasmussen and G. Aaron. Global Phishing Survey: Trends and Domain Name Use in 1H2010. Technical report, APWG, Oct. 2010.

[13] R. Rasmussen and G. Aaron. Global Phishing Survey: Trends and Domain Name Use in 1H2011. Technical report, APWG, Nov. 2011.

[14] R. Rasmussen and G. Aaron. Global Phishing Survey: Trends and Domain Name Use in 1H2012. Technical report, APWG, Oct. 2012.

[15] T. Rodrigues, F. Benevenuto, M. Cha, K. Gummadi, and V. Almeida. On word-of-mouth based discovery of the web. In *Internet Measurement Conference*. ACM Request Permissions, Nov. 2011.

[16] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Annual Computer Security Applications Conference*, pages 1–9, Austin, TX, USA, Dec. 2010. ACM Request Permissions.

[17] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *SSP '11*, pages 447–462. IEEE, 2011.