Measuring the Effectiveness of Twitter's URL Shortener (t.co) at Protecting Users from Phishing and Malware Attacks

Simon Bell
Royal Holloway, University of London
simon.bell.2014@rhul.ac.uk

Peter Komisarczuk
Royal Holloway, University of London
peter.komisarczuk@rhul.ac.uk

ABSTRACT

In this paper we investigate how effective Twitter's URL shortening service (*t.co*) is at protecting users from phishing and malware attacks. We show that over 10,000 unique blacklisted phishing and malware URLs were posted to Twitter during a 2-month timeframe in 2017. This lead to over 1.6 million clicks which came directly from Twitter users – therefore exposing people to potentially harmful cyber attacks. However, existing research does not explore if blacklisted URLs are blocked by Twitter at time of click.

Our study investigates Twitter's URL shortening service to examine the impact of filtering blacklisted URLs that are posted to the social network. We show an overall reduction in the number of blacklisted phishing and malware URLs posted to Twitter in 2018-19 compared to 2017, suggesting an improvement in Twitter's effectiveness at blocking blacklisted URLs at time of tweet. However, only about 12% of these tweeted blacklisted URLs – which were not blocked at time of tweet and therefore posted to the platform – were blocked by Twitter in 2018-19. Our results indicate that, despite a reduction in the number of blacklisted URLs at time of tweet, Twitter's URL shortener is not particularly effective at filtering phishing and malware URLs - therefore people are still exposed to these cyber attacks on Twitter.

CCS CONCEPTS

• **Security and privacy → Malware and its mitigation**; **Phishing**; • **General and reference** → *Empirical studies*; *Measurement*;

KEYWORDS

Security, Malware, Phishing, Blacklists, Measurement Study

ACM Reference format:
Simon Bell and Peter Komisarczuk. 2020. Measuring the Effectiveness of Twitter's URL Shortener (t.co) at Protecting Users from Phishing and Malware Attacks. In *Proceedings of the Australasian Computer Science Week Multiconference, Melbourne, VIC, Australia, February 4–6, 2020 (ACSW 2020),* 11 pages.
https://doi.org/10.1145/3373017.3373019

ACSW 2020, February 4–6, 2020, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-7697-6/20/02…$15.00
https://doi.org/10.1145/3373017.3373019

1 INTRODUCTION

Since its creation in 2006, Twitter has gained over 974 million users with 330 million active users per month posting 500 million tweets per day[2]. Among these Twitter users are many high profile celebrities, politicians, heads of states and societal influencers whom attract large numbers of followers [38]. Due to this large user base, Twitter makes an attractive target for malicious users aiming to carry out phishing and malware attacks to exploit people. One of the main ways attackers carry this out is by leading victims to a malicious site, by including one or more URLs in a tweet, whereby the attack can occur.

Phishing attacks on Twitter have been known to lure victims in by offering verification on the social network but instead take them to a fake login page to steal their Twitter username and password [4] while malware attacks have included drive-by-download links contained within tweets, trojans, cross-site scripting attacks [1] and Android malware that is controlled by tweets [9].

Twitter has come under increasing pressure to protect its users against these attacks, such as, in 2010 when the company settled a case with the FTC in which Twitter agreed to strengthen security throughout the platform and to carry out an independently assessed bi-annual information security audit [11].

One of the ways in which Twitter is improving its security for users is by implementing numerous rules [37] that govern what type of content users of the platform can and cannot send. In 2009, it was reported [24] that Twitter had started to use the phishing and malware blacklist, Google Safe Browsing (GSB) – used by popular web browsers – to filter out and protect users from attack URLs. In June 2010 Twitter announced[1] that it was working on its own URL shortening service (*t.co*) – this service launched a year later in June 2011[2]. All URLs posted to Twitter (i.e. tweeted) are shortened via Twitter's URL shortening service (*t.co*). If a URL has already been shortened with an existing URL shortener(s) (such as Bitly, Goo.gl, etc) then a redirection chain is formed, starting with *t.co*. This gives Twitter full control over and monitoring of URLs posted to, and clicks leaving, its network.

The aim of our paper is to explore how effective Twitter's URL shortening service, (*t.co*) is at protecting Twitter users from phishing and malware attacks. A 2010 study [15] provided evidence to suggest that Twitter is not using GSB effectively to protect users from such attacks. The study showed that 100,000 phishing and malware URLs were tweeted to the social network – therefore endangering users. Some of these blacklisted URLs accumulated 1.6 million clicks from Twitter users – although the total number

[1] https://blog.twitter.com/official/en_us/a/2010/links-and-twitter-length-shouldn-t-matter.html
[2] https://www.geek.com/news/twitter-finally-adds-its-own-url-shortening-service-1388467/

of URL clicks in the study includes phishing, malware, and scam URLs. These specific results from the 2010 paper were part of a much larger study aiming to characterise spam and analyse features unique to Twitter.

In 2017 [3] we investigated the delay times between phishing/-malware URLs being tweeted to appearing in blacklists. In this follow-up paper to our 2017 work, we build on our existing research to investigate Twitter's URL shortener.

Existing research suggests that Twitter is not effectively using the GSB blacklist and therefore relying on web browsers built-in protection to prevent users from visiting dangerous URLs. Twitter's "time of click" URL filtering (via *t.co*) has not previously been studied. The main motivation for our study is to determine if blacklisted URLs, tweeted to the social network, are filtered at time of click – via Twitter's URL shortener (*t.co*) – and, if they are, how effective this filtering process is.

In our paper we investigate if, since the 2010 study, the same volume of phishing and malware URLs are still being posted to Twitter. We also investigate if Twitter has implemented a filter to block blacklisted phishing and malware URLs on their platform at time of click – therefore preventing users from visiting these potentially harmful websites.

Our main research questions are:

- Are phishing and malware URLs still being posted to Twitter (i.e. tweeted)?
- Has Twitter moved to a "time of click" defence strategy - via their *t.co* URL shortener - rather than a "time of post"?
- If so: does Twitter's filtering (via *t.co*) complement GSB blacklisting?
- How many tweeted URLs that are blacklisted are **also** filtered by *t.co*?

We organise the remainder of this paper as follows. Section 2 provides a background to the three main blacklists we use in this study and also explores previous studies related to our work. Section 3 presents an overview of our infrastructure, explains our data collection process and methodology, and provides an overview of our experiments. Section 4 gives technical details of our infrastructure implementation. Section 5 presents our key measurement results and interpretations. Section 6 discusses our findings in this paper, followed by our conclusion in Section 7.

2 BACKGROUND AND RELATED WORK

2.1 Blacklists

A blacklist is defined as a set of elements to be blocked; an access control list. Our study looks at phishing and malware blacklists that are used to block access to URLs posted to Twitter. We focus on 3 blacklists in this study: Google Safe Browsing (GSB) [13], Phish Tank (PT) [29], and Open Phish (OP) [25].

Google Safe Browsing: launched in 2007, Google Safe Browsing (GSB) is a URL blacklist that contains both malicious and phishing URLs and is used by the web browsers Google Chrome, Safari, Firefox, Opera, and Vivaldi to protect users from dangerous websites. We focus on GSB in our study because of its prominence in popular web browsers: already in 2012 GSB was protecting 600 million users from dangerous websites [40]. In 2015 GSB began using the term "Social Engineering" to categorise phishing websites which also encompass additional types of deceptive content. Google defines a social engineering web attack as occurring when either: "the content pretends to act, or looks and feels, like a trusted entity - like a bank or government" or "the content tries to trick you into doing something you would only do for a trusted entity - like sharing a password or calling tech support" [12]. During the week commencing 3rd September 2017 the total number of sites deemed dangerous by GSB was 573,433 phishing and 500,245 malicious. During that week GSB detected 24,756 new phishing sites and 6,312 new malware sites. GSB defines malware websites in its blacklist as being either *compromised* or *attack*. A compromised website is a legitimate website that has been hijacked to either include, or direct users to, malicious content. An attack site is a website that has intentionally been set up to host and distribute malware [14]. During the week commencing 3rd September 2017, GSB identified 5,981 new compromised websites and 335 new attack websites.

GSB provides two APIs for accessing its blacklist: Lookup and Update. The Lookup API provides a remote service whereby URLs to be checked are sent to Google's servers and a response is returned for each URL stating if the URL is in the blacklist. The Update API provides the user with a local copy of the blacklist, this local copy is stored as a database of SHA-256 URL hash prefixes, the majority of the hash prefixes being 4 bytes. To perform a URL blacklist lookup, the URL hash prefix is checked in the local database and, if there is a prefix match, then the full URL hash is retrieved from Google's servers to determine if there is a match on the full hash.

Phish Tank: launched in October 2006, Phish Tank (PT) provides a community based phishing website reporting and verification system. Users of the website can submit URLs of suspected phishing websites; the Phish Tank community then vote as to whether these URLs are phishing or not. Phish Tank is used by the web browser Opera, online reputation and internet safety service web browser plugin Web Of Trust, email provider Yahoo! Mail, and antivirus providers McAfee and Kaspersky [28]. The Phish Tank blacklist of approved phishing URLs can be downloaded as a JSON file and typically contains around 23,000 unique URLs.

Open Phish: launched in 2014, Open Phish (OP) is the result of a 3 year research project on phishing detection that uses autonomous algorithms to detect zero day phishing websites. Our study has access to the academic feed. Open Phish is used by the antivirus companies Virus Total and Strong Arm. The Open Phish blacklist can be downloaded as a JSON file which typically contains around 5,000 unique URLs.

2.2 Related Work

Existing literature has explored the effectiveness of malware blacklists [17, 18] and also phishing attacks in areas such as why they work [7], the effectiveness of toolbars in protecting users [41, 42], detection of phishing websites [43], the effectiveness of web browser warnings [8], demographic analysis of phishing susceptibility and effectiveness of interventions [31], and a study to determine a baseline for phishing campaign success [16]. There are also various techniques to prevent phishing attacks including Dynamic Security Skins [6], Trusted Devices [27] along with educational aspects of

phishing training including PhishGuru [19] and the game Anti-Phishing Phil [32]; the effectiveness of these two educational approaches were analysed [20]. Previous studies have also developed techniques to detect spam, phishing and malware on Twitter, such as looking at redirection chains to detect suspicious URLs [21], analysing suspended accounts [35], and using social graph models [39]. Phishers that use URL shortening services to masquerade phishing URLs on Twitter have been studied [5] and short URLs have been measured to explore threats and assess the effectiveness of countermeasures [23].

Two key studies, carried out in 2007 [22] and 2009 [33] focused on phishing blacklists and how effective they are at protecting users from phishing email attacks, paying particular attention to the delay from an email containing a phishing URL being received to that URL appearing in a blacklist. We focus on Twitter as a delivery platform rather than e-mail.

Whilst these previous studies have looked at the landscape in terms of detecting and preventing phishing and malware attacks, they have not focused specifically on the relationship between blacklists and phishing and malware attacks on Twitter. In addition, Twitter's URL shortener, *t.co*, has not previously been studied to determine its effectiveness at blocking phishing and malware attacks. A 2010 study [15] characterised phishing, malware, and scam URLs posted to Twitter. Part of the study analysed blacklist delays on Twitter and produced evidence showing 100,000 phishing and malware URLs were posted to the social network during a 1 month period. The study also provided evidence showing that spam URLs received 1.6 million clicks from Twitter users. In June 2010 Twitter announced[3] that it was working on its own URL shortening service – this service launched a year later in June 2011[4]. Therefore the 2010 study was conducted before Twitter implemented its URL shortener, *t.co*.

To our knowledge, this is the first research study to measure the effectiveness of Twitter's URL shortener at protecting users from phishing and malware attacks.

3 DESIGN

For our measurement study we built an infrastructure that is designed to work around some limitations of the data sources we use. These limitations include: Twitter's data feed is a "small sample" of all global tweets, the GSB API blacklist is limited to 10,000 daily lookup per day, and our Twitter URL filter checking system cannot send too many HTTP requests per second because this would flood Twitter's servers. The methodologies described in this section ensure that, despite these limitations of the data sources we use, our measurement framework, and experiments we run on this framework, produce accurate results to answer our research questions. The methodologies described in this section are explained in more technical details in Section 4: Implementation.

In our 2017 study [3] we used the same infrastructure to investigate delay times between phishing/malware URLs being tweeted and appearing in blacklists. During our 2017 study, the GSB API was not rate limited to 10,000 requests per day - therefore we were able to check higher volumes of URLs per day for blacklist membership. Since 2017, the GSB rate limit has come into effect, which makes it infeasible to exactly repeat some of the experiments in our 2017 study. Therefore, in our 2018-19 experiments, we alter our GSB lookup methodology and build on our existing research to investigate Twitter's URL shortener. We include some of the key results from our 2017 study is this paper.

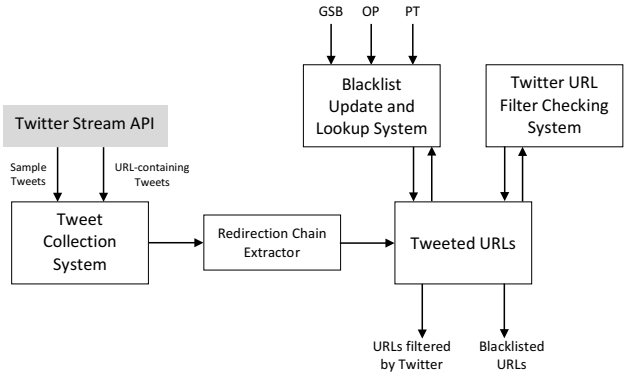


Figure 1: Infrastructure design architecture.

3.1 Overview

The infrastructure used to carry out experiments in this study consists of a tweet collection system that receives both sample tweets and tweets containing URLs from Twitter; a database to store these collected tweets; a URL redirection chain extractor; a blacklist system to store, update and perform lookups against 3 popular blacklists; a database to store tweeted URLs which have appeared in a blacklist; and a system to determine if tweeted URLs are blocked by Twitter. The overall design architecture of our infrastructure system can be seen in Figure 1.

3.2 Data Collection

The first data requirement for our study is a source of live tweets from Twitter. To achieve this we use Twitter's Stream API, set with a filter to only retrieve URL-containing tweets. Our dataset contains both the original tweeted URLs along with their shortened *t.co* aliases. It is important to note that our 2017 and 2019 datasets were collected using the same sampling strategy.

The second data requirement for our study is a way to store and search various blacklists. For our study 3 blacklists are used: Google Safe Browsing (GSB), Open Phish (OP), and Phish Tank (PT). The GSB blacklist includes both social engineering and malicious URLs. Our system regularly obtains the latest copies of these blacklists and saves them locally in a database on our system. Tweeted URLs, from our collection of tweets, can then be searched for in these 3 blacklists.

The third data requirement for our study is the ability to check all tweeted URLs, collected from Twitter's Stream API, to determine if they have been blocked by Twitter. All URLs tweeted on Twitter's platform are shortened via Twitter's URL shortener, *t.co*. These *t.co* shortened URLs are stored in our database alongside the corresponding full URLs. We retrieve the redirection chain of each

[3]https://blog.twitter.com/official/en_us/a/2010/links-and-twitter-length-shouldn-t-matter.html

[4]https://www.geek.com/news/twitter-finally-adds-its-own-url-shortening-service-1388467/

t.co URL in our dataset to determine which URLs have been blocked by Twitter.

3.3 Methodology

As described in the previous subsection, we collect URL-containing tweets from Twitter's Filter Stream API and store them in a local database. We also store 3 blacklists in the local database. In order to determine which tweeted URLs appear in the 3 blacklists two main systems are used: a GSB blacklist lookup system and a Phish Tank and Open Phish blacklist lookup system. Our GSB blacklist system uses GSB's API to check tweeted URLs for blacklist membership. This API is rate limited to 10,000 lookups per 24-hour period, therefore we check all tweeted URLs from the past 24-hours to 7-days for GSB membership approximately every 7 hours. The GSB blacklist lookup system updates its list of URLs every 10 minutes and the Phish Tank and Open Phish blacklist lookup system updates its URLs once every hour. Timestamps for when URLs are added and removed from the Phish Tank and Open Phish blacklists are used to determine URL membership times.

For a tweeted URL, there could be a number of hops or redirections before arriving at the final landing page. For this reason a redirection chain extractor system is used to check if each URL, contained within the redirection chain, exists in any of the blacklists. The technicalities of this redirection chain extractor system are explained in more detail in the implementation section.

To determine if a tweeted URL has been blocked by Twitter we analyse the redirection chair for each *t.co* URL collected via Twitter's Filter Stream API. If a tweeted URL (shortened via *t.co*) has been blocked by Twitter then a warning page is displayed to the user and there is no redirection chain. If a URL is not blocked by Twitter then *t.co* forwards the user directly to the URL that was shortened, thereby creating a redirection chain. By analysing which *t.co* URLs have redirection chains and which do not then we can determine which tweeted URLs have been blocked by Twitter. This system is carefully rate limited to ensure we do not flood Twitter's servers.

3.4 Overview of Experiments

The key experiments we carry out in this paper are:

(1) Measure number of phishing and malware URLs posted to Twitter. Also, to assess impact, measure clicks from Twitter users to these blacklisted URLs
(2) Measure Twitter filtering: does Twitter filter URLs (via its *t.co* URL shortener)? If so, does Twitter block blacklisted URLs?
    - 24-hour measurement period
    - 7-day measurement period

Measure number of phishing and malware URLs posted to Twitter. The aim of this experiment is to understand how many phishing and malware URLs are posted to Twitter. We also want to understand the impact of these tweeted URLs by investigating how many clicks they receive from Twitter users. We use Twitter's Stream API to collect URL-containing tweets, along with the GSB API and the OP and PT phishing blacklists. Tweeted URLs are checked for membership in GSB every 10 minutes, and membership in PT and OP every hour, throughout the experiment. We then use Bitly's API to measure how many clicks blacklisted Bitly URLs receive that were posted to Twitter, in our dataset.

Measure Twitter Filtering: This section is split into two experiments: 24-hour measurements and 7-day measurements. Both experiments have 3 key aims:

(1) determine how many tweeted URLs are blocked by Twitter (via the *t.co* URL shortener)
(2) determine how many tweeted URLs are blacklisted by either GSB, OP, or PT
(3) determine how many tweeted URLs are blocked by Twitter **and** blacklisted - by either GSB, OP, or PT

Both experiments involve collecting URL containing tweets, via Twitter's Stream API, then checking these URLs to determine if they are blocked by Twitter and/or blacklisted by GSB, OP or PT. For the 24-hour measurements this check is carried out once for Twitter filtering and GSB membership. For the 7-day experiment this check is carried out 25 times over the duration of 7 days. The specific methodologies for each experiment are described below.

24-hour experiments: This measurement experiment analyses all tweeted URLs we collect from Twitter's Stream API over the past 24 hours. We determine how many tweeted URLs are blocked by Twitter's URL shortener (*t.co*), by checking all tweeted URLs from the past 24 hours - it takes approximately 7 hours to complete this check due to rate limiting. We collect approximately 3 million tweeted URLs from Twitter's Stream API per 24 hours, of which approximately 1.5 million are unique. Therefore each 7 hour cycle checks approximately 1.5 million tweeted URLs to determine if they have been blocked by Twitter. The second part of this experiment, which determines how many tweeted URLs are included in the 3 main blacklists (GSB, OP, and PT), checks all tweeted URLs from the past 24 hours for blacklist membership and takes approximately 10 minutes to complete this lookup. Each tweeted URL is only checked for GSB membership once in this experiment due to the GSB API rate limit. This experiment is carried out on a rolling timeline, i.e. at the start of each lookup cycle the most recent tweeted URLs from the past 24-hours are checked.

7-day experiments: This experiment is similar to the previous, however, the methodology in this experiment is altered to check tweeted URLs for GSB membership multiple times over a 7-day period. In this experiment the two parts are **combined** so that all tweeted URLs are checked to determine if they have been blocked by Twitter's URL shortener *t.co* **and** for GSB membership **at the same time**. This means both parts of this experiment take approximately 7 hours to check 1.5 million tweeted URLs. The main advantage to this methodology is that we can determine if tweeted URLs appear in the GSB blacklist and also check if they are blocked by Twitter – at the same time. We also check tweeted URLs for GSB membership multiple times, over a 7-day period, which means we can determine if URLs are removed from and possibly re-appear in GSB. This experiment includes OP and PT blacklist lookups, as before.

4 IMPLEMENTATION

Our entire system is implemented on a virtual machine running Ubuntu operating system, 8 core CPU, 24 GB RAM. The measurement framework is written in Python programming language.

4.1 Tweet Collection System

Our Twitter collection system uses Twitter's Stream API, implemented via the *Tweepy* [36] library. After authorising *Tweepy* to access Twitter, the *sample()* and *filter()* methods are used to collect sample and URL containing tweets. The filter method uses keywords *"http"* and *"https"* to filter out tweets containing URLs. All data received from Twitter's Stream API, using these two methods, is stored in a MySQL [26] database in two tables for sample tweets and URL-containing tweets, respectively.

4.2 URL Redirection Chain Extraction System

The URL redirection chain extraction system uses Python's *Requests* library [30] to send an HTTP request for each URL using a Macintosh Safari user agent header so the request appears to come from a regular user via the Safari web browser. The reason for setting this header is so the request extracts the same redirection chain that a legitimate user would see and not a redirection chain that a bot would see – therefore reducing bias in our results. The *Request* library's *Response* object contains a *History* property which consists of a list of *Response* objects that were created to complete the HTTP request. This list is then used to extract the redirection chain for a given URL in our system.

4.3 Blacklist Update and Lookup System

We use 3 blacklists in our system: GSB, OP, and PT. To implement our GSB lookup system, the library *gglsbl* [10] is used. This library allows our system to fetch the latest GSB hash prefixes and also perform lookups against the database. The library uses the SQLite [34] database for storing GSB data. The library contains a method *update_hash_prefix_cache()* which is used to update the URL hash prefix database. This method is called every 10 minutes.

An important modification was made to the *gglsbl* library to improve lookup times for large numbers of URLs. The method *lookup_url()* is used to lookup an individual URL in the local URL hash prefix database. It does this by performing an SQLite search for that URL's hash prefix. This lookup technique caused a significant bottleneck when testing the system on large volumes of URLs - due to an SQL search being performed for every URL to be checked. Therefore, we modified this process to extract all SHA256 URL hash prefixes directly from the local SQLite database into a Python *dictionary* (i.e. hash table). Our system then performs a lookup for each tweeted URL's SHA256 hash prefix against this *dictionary*. Since the Python *dictionary* implementation uses a hash map, the typical time complexity for this lookup is constant; $O(1)$. This means lookups are considerably faster using this modified lookup technique compared to the GSB library's lookup method.

During our experiment, we observe that the GSB blacklist typically contains approximately 4.8 million URL hash prefixes of which approximately 3.1 million are unique. Of these, there are approximately 1 million unique URL hash prefixes labelled malware and approximately 1.8 million unique URL hash prefixes labelled social engineering. The remaining URL hash prefixes labels include *unwanted software*, and *potentially harmful application*, which we label as "other" in our blacklist experiments.

Both the PT and OP blacklist datasets are download as JSON files from their websites. The URL entries from these files are then extracted and saved into our local MySQL database. Metadata stored along with URLs includes discovery timestamps from the blacklists and timestamps for when URLs were added to our database. Both datasets are downloaded every hour and new entries saved in the local database. URL lookups for these two blacklists are completed by importing all URLs from both local blacklist databases and storing them in a Python *dictionary* in order to perform faster lookups, as per our GSB lookup implementation.

4.4 Twitter URL Filter Checking System

Our *t.co* URL checking system determines if Twitter blocks tweeted URLs. The system uses Python's *Requests* library [30] to send an HTTP request to each *t.co* URL along with a Macintosh Safari user agent header setting so the request appears to come from a regular user via the Safari web browser. We discovered that *t.co* only displays warning pages to web browsers – a warning page was not displayed when this user agent setting was disabled. We use Python's *Threading* library to concurrently check 200 *t.co* URLs every approximately 3.5 seconds. This is achieved by setting the *Requests* library's *connect timeout* value to 3.05 seconds and *read timeout* value to 6.05 seconds. A value slightly larger than a multiple of 3 is used since this is the default TCP packet retransmission window[5]. These settings ensure our system runs efficiently and does not get stuck waiting indefinitely for servers to respond. Additional time is incurred when factoring in computations such as database inserts, GSB URL lookups etc – hence the overall average batch completion time of 3.5 seconds for 200 URLs.

Our *t.co* URL checking system takes approximately 7 hours to check all, approximately 1.5 million unique *t.co* URLs, collected during a 24-hour period (1.5 million / 200 [per batch] * 3.5 seconds = 7.3 hours). Of those approximately 1.5 million *t.co* URLs, our system typically drops approximately 400 (0.03%) connections (i.e. *read timeout* errors). It is important to note that many of these timeouts likely occur because the website being tested is offline therefore a larger timeout value would have minimal impact. We decided to process 200 *t.co* URLs per 3.5 second batch by testing various different timings. When more than 200 *t.co* URLs were processed in a 3.5 second period Twitter's servers returned HTTP 503 status code responses (*temporary overload*[6]). After fine tuning and experimenting with different timing values, we concluded that processing 200 *t.co* URLs per 3.5 second batch worked well – typically returning zero to one 503 errors per 1.5 million *t.co* URLs processed. This means that our system is rate limited and therefore not overloading Twitter's servers with requests.

5 RESULTS

5.1 Blacklisted URLs Posted to Twitter

To understand how widespread potentially harmful URLs are on Twitter, in this experiment we measure how many phishing and malware URLs are publicly tweeted during a 2-month period. We monitor URLs, tweeted between October and November 2017, for

[5] https://tools.ietf.org/html/rfc2988
[6] https://httpstatuses.com/503

	Oct '17		Nov '17	
Blacklist	**URLs**	**Domains**	**URLs**	**Domains**
GSB: phishing★	4,912	397	2,495	268
GSB: phishing†	3,273	212	930	182
GSB: malware★	1,563	250	1,054	144
GSB: malware†	718	82	543	65
Open Phish★	2	2	1	1
Open Phish†	1	1	1	1
Phish Tank★	2	2	4	3
Phish Tank†	1	1	4	3

Table 1: Overview showing total number of phishing and malware URLs posted to Twitter (i.e. tweeted), in our dataset, during October and November 2017. Using blacklists GSB, OP, and PT.
★Blacklisted anytime during experiment.
†Blacklisted within 1 month from date first tweeted.

membership in blacklists: GSB, OP, and PT. During the measurement period we collected more than 182 million public tweets containing URLs from Twitter's Stream API. Table 1 shows the total number of blacklisted phishing and malware URLs we detected during this timeframe. We use 2 measurement views to count blacklisted URLs: those blacklisted anytime during the experiment and those blacklisted within 1 month from date of tweet. We also count the total number of domain names in each category to give a sense of how many blacklisted URLs originate from domains. These results show that significantly more phishing than malware URLs were tweeted during our study, a similar finding to 2010. This may be because there were nearly twice as many phishing than malware URLs in the GSB blacklist during this measurement timeframe. Also, Twitter may be a more suitable platform for carrying out phishing attacks due to features unique to Twitter. In our results we see a high number of blacklisted URLs come from a smaller set of domains. This shows how a single malicious user can leverage multiple attacks from a domain therefore highlighting the importance monitoring domain names for widespread attack URLs.

It is interesting to note that only 9 URLs from Open Phish and Phish Tank appeared in our dataset. Considerably fewer tweeted URLs appeared in the Open Phish and Phish Tank blacklists compared to GSB. One reason for this difference may be that the GSB blacklist contains approximately 3 million URLs whereas the Phish Tank and Open Phish blacklists contain 28,000 URLs combined; there are fewer URLs for Phish Tank and Open Phish to detect. Another possibility is that Twitter is using the Phish Tank and Open Phish blacklists and therefore preventing users from tweeting URLs contained within these blacklists. However, if that were the case, then we would still see URLs in the Twitter Stream before they appear in the Open Phish or Phish Tank blacklists.

A total of 10,029 unique tweeted URLs were blacklisted during our 2-month measurement period; more than 45% of these URLs took over 1 month to become blacklisted. This means these blacklisted URLs were publicly available for Twitter users to visit – therefore exposing people to these potentially harmful attacks.

5.1.1 Blacklisted URL Clicks from Twitter Users.

To explore the impact of phishing and malware URLs posted to Twitter, we analyse the number of clicks these potentially harmful URLs receive. To achieve this we use Bitly's API to collect data showing how many clicks – directly from Twitter (by using the referrer metadata) – blacklisted Bitly URLs received in our dataset. Bitly[7] is a URL shortening service that also provides public analytics for URL clicks, referrers, and location, via an API, for URLs shortened on its platform. We extract all tweeted Bitly URLs in our dataset, that appeared in the GSB blacklist, and retrieve their click data from the Bitly API for further analysis. We also include Bitly URLs within redirection chains that lead to blacklisted URLs in our dataset.

	Phishing (GSB)		Malware (GSB)	
Data	**Oct '17**	**Nov '17**	**Oct '17**	**Nov '17**
Tweets containing Bitly URLs	1,126	146	32	103
Unique Bitly URLs	376	141	30	66
Percentage of all blacklisted URLs in this category and timeframe	11%	15%	4%	12%
Bitly clicks	991,012	450,039	61,140	194,503

Table 2: Total number of tweets containing Bitly URLs, unique Bitly URLs, percentage of all URLs for each category and timeframe, and total Bitly clicks for tweets containing GSB blacklisted phishing and malware URLs, in our dataset, during October and November 2017.

Table 2 shows the total number of unique Bitly URLs, percentage of all URLs in this category and timeframe, total number of tweets containing Bitly URLs for this category and timeframe, and total number of Bitly URL clicks, from our dataset of blacklisted tweeted URLs, during October and November 2017. These results show that, during our 2-month measurement, approximately 10% of blacklisted phishing and malware URLs posted to Twitter received over 1.6 million clicks – from Twitter users. This gives a sense of the total clicks received by all blacklisted URLs posted to Twitter during this timeframe.

To investigate the impact of tweeting a blacklisted URL to a Twitter account with a high number of followers, we extracted a blacklisted URL, from our dataset, that was shortened with Bitly. The blacklisted Bitly URL was tweeted by an account with 3.7 million followers on October 24 and flagged as phishing in GSB on November 11. The URL received 276 clicks during the week of October 22, of which 270 came from Twitter. 176 of these clicks came from the USA, 19 from Canada, 12 from the UK and the remaining 34 from elsewhere. This URL did not receive any more clicks after the week of October 22 at which point it appears to have been blocked by Bitly. This example shows how a single tweet from a high follower account, posting a dangerous URL, can receive a high number of global clicks – therefore exposing a large amount of Twitter users to the attack. It also shows that GSB took 18 days to add the URL to its blacklist, while Bitly appears to have blocked the URL much sooner. In this scenario, Twitter appears to have outsourced its filter to Bitly – relying on Bitly to protect Twitter's users.

[7] https://bitly.com/

5.1.2 Posting Blacklisted URLs to Twitter.

In a separate experiment we created a private account on Twitter whereby the account's tweets were not publicly visible. We then attempted to tweet a sample of 30 blacklisted URLs: 10 each from GSB, OP, and PT. In this experiment, 8 of the Open Phish URLs and 9 of the Phish Tank URLs could not be posted to Twitter. All of the GSB URLs were posted successfully to Twitter. For tweets containing blacklisted URLs that could not be posted to Twitter this error message was displayed: "This request looks like it might be automated. To protect our users from spam and other malicious activity, we can't complete this action right now. Please try again later". We were able to tweet messages that did not contain blacklisted URLs without receiving this error message. This suggests that Twitter may display this generic error message when users attempt to tweet URLs that are blacklisted. It is important to note that this was a small-scale study and that the Twitter account used for this experiment was set to private, therefore all tweets were hidden from the public. Public Twitter accounts may see different results in this experiment – for example: public tweets may go through a stricter filtering process. Due to ethical considerations, we did not post any public tweets containing blacklisted URLs.

Key Findings: The results in this subsection show that over 10,000 unique blacklisted phishing and malware URLs were posted to Twitter during a 2-month measurement timeframe in 2017. This lead to over 1.6 million clicks which came directly from Twitter users – therefore exposing people to potentially harmful cyber attacks. We also see that Twitter appears to be blocking a greater volume of URLs – at time of tweet – that reside in the PT and OP blacklists compared to GSB.

In this subsection we have seen evidence that Twitter blocks some blacklisted URLs at time of tweet – but that a large number of these URLs are not blocked. In the next subsection we assess if Twitter implements additional protection for its users against these attacks. We do this by investigating Twitter's URL shortener (*t.co*) to determine if a filtering system is implemented.

5.2 Investigation of Twitter Filtering

This subsection investigates if Twitter implements a "time of click" filtering system to protect its users against blacklisted URLs. This builds on our 2017 results, seen in the previous subsection, where we showed that phishing and malware URLs are being posted to Twitter and receive clicks from Twitter users. This subsection is split into two key measurement experiments: 24-hour and 7-day, carried out in 2018-19.

5.2.1 24-hour Measurement Period.

We collected tweeted URLs via Twitter's Stream API (using URL filter), to determine how many of these tweeted URLs are blocked by Twitter (via their URL shortener *t.co*); how many of these URLs are blacklisted by the 3 blacklists: Google Safe Browsing (GSB), Open Phish (OP), and Phish Tank (PT); and how many of these URLs are both blocked by Twitter **and** blacklisted. We collected these tweeted URLs over a 3-month period during November, December, and January 2018-19. All tweeted URLs from the past 24 hours were checked, approximately, every 10 minutes for GSB blacklist membership, every hour for OP and PT blacklist membership, and every 7 hours for Twitter filtering. Full details of the data collection and methodology are in Section 3, implementation details are in Section 4.

	Nov '18	Dec '18	Jan '19
Total tweeted URLs	**85M**	**88M**	**88M**
Unique t.co (shortened)	35M	35M	36M
Blocked by Twitter	10,297	4,038	5,636
Unique unshortened	31M	31M	32M
Blocked by Twitter	5,923	3,893	4,389
Blacklisted	**67**	**118**	**251**
GSB	62	117	249
GSB phishing	54	94	108
GSB malware	2	8	115
GSB other	6	15	27
PT	5	1	2
OP	0	0	0
Blocked by Twitter and blacklisted	**12**	**16**	**18**
GSB	7	15	16
PT	5	1	2
OP	0	0	0

Table 3: Overview showing total number of tweeted, blocked by Twitter (via t.co), and blacklisted URLs up to 24-hours after time of tweet, in our dataset. Experiments conducted during Nov, Dec, & Jan 2018-19 (M = million).

Table 3 shows an overview of the total number of tweeted, blocked by Twitter (via *t.co*), and blacklisted URLs we collected during November, December, and January 2018-19. The table also shows the total number of tweeted URLs that were both blocked by Twitter **and** blacklisted (in one of the 3 blacklists). The total number of tweeted URLs we collected is broken down into total unique *t.co* (shortened) and total unique *unshortened* URLs. All URLs posted to Twitter (i.e. tweeted) are shortened via Twitter's URL shortener, *t.co*. We define an "*unshortened* URL" to be the original tweeted URL that has an associated (i.e. shortened) *t.co* URL. A single tweeted URL can be shortened to one or more different *t.co* URLs – for example: a number of different Twitter users may tweet the same URL which gets shortened to separate *t.co* URLs. Therefore we see a higher number of unique *t.co* URLs in the results table compared to unique *unshortened* URLs. The table also shows how many unique *t.co* URLs were blocked by Twitter and how many *unshortened* URLs were blocked by Twitter; how many tweeted URLs appeared in the 3 main blacklists (GSB, OP, and PT), along with categorisations for GSB; and finally how many tweeted URLs were blocked by Twitter **and** blacklisted, along with which blacklists they appeared in.

The results in Table 3 show that Twitter's defence strategy does indeed involve filtering URLs at time of click via their URL shortener, *t.co*. However, we also see that, of the tweeted URLs which were blacklisted, only 7-18% were also blocked by Twitter: 18% (12 of 67) in November, 14% (16 of 118) in December, and 7% (18 of 251) in January. This shows that, in our dataset, over 80% of blacklisted phishing and malware URLs posted to Twitter are not blocked by the social network – therefore allowing Twitter users to click on these potentially dangerous links. These results show that Twitter blocked over 14,000 unique, unshortened URLs during our

3-month measurement period – but only 2% of these were blacklisted phishing and malware URLs in GSB, OP, or PT. It is important to note that Twitter's Rules[8] mainly focus on preventing graphic violence, abusive behaviour, violence and physical harm, hateful conduct, impersonation, and spam. Twitter does explicitly state that malware/phishing is not allowed on its platform: "You may not publish or link to malicious content intended to damage or disrupt another person's browser or computer or to compromise a person's privacy". However, this is a small part of its extensive list of rules. Therefore, it is likely that the majority of these URLs blocked by Twitter are for violations of its rules other than phishing/malware. Analysis of the categorisation of URLs blocked by Twitter, that are not phishing/malware, is outside the scope of our study.

Our results also show that a larger volume of phishing URLs than malware URLs were posted to the social network, which we also saw in our 2017 experiments. This suggests that Twitter continues to be an effective platform for malicious users to carry out social engineering attacks on – or perhaps that Twitter is more aggressive at removing malicious URLs therefore eliminating them from the platform in the first place. It is interesting to observe that all tweeted URLs residing in the PT blacklist were also filtered by Twitter. This may suggest that Twitter blocks any URLs residing in the PT blacklist at time of tweet – however it is not possible to extract this information from our experiments directly. Upon further analysis of the top domains in this dataset, we discovered that the majority of domains blocked by Twitter are shortened URLs (via services Bitly, Zaper, goo.gl, Ow.ly, flic.kr, page.link, etc.) and a large number of pornographic websites. This suggests the shortened and pornographic URLs direct to websites that are in violation of Twitter's Rules. The results also show a small number of GSB blacklisted URLs labelled as "other", these include unwanted software and *potentially harmful applications*.

It is important to note that, during these 24-hour measurement experiments, tweeted URLs were checked for GSB blacklist membership once within 10 minutes of tweet and checked for Twitter filtering up to 4 times within 24 hours of tweet. Tweeted URLs were not measured beyond 24 hours. We improve on this methodology in the next subsection by checking all tweeted URLs, collected over a 24-hour period, for GSB membership **and** Twitter blocking – at the same time – every 7 hours over a 7-day period.

5.2.2 7-day Measurement Period.

The results in this subsection analyse tweeted URLs we collected via Twitter's Stream API, to determine how many of these tweeted URLs are blocked by Twitter (via their URL shortener *t.co*); how many of these URLs are blacklisted by the 3 blacklists: Google Safe Browsing (GSB), Open Phish (OP), and Phish Tank (PT); and how many of these URLs are both blocked by Twitter **and** blacklisted. We carried out 4 key experiments, each over a duration of 7 days. Each experiment collects 3 million tweeted URLs from the past 24 hours then analyses that batch of URLs over a 7-day period. This differs from the previous section whereby tweeted URLs were only analysed for up to 24 hours after tweet. These experiments were conducted in April and May 2019. All tweeted URLs from the most recent 24-hour period were checked every 7 hours for Twitter filtering and membership in the 3 blacklists (GSB, OP, and PT). In total, all tweeted URLs from the same 24-hour period are checked 25 times for Twitter filtering and blacklist membership. Full details of the data collection and methodology are in Section 3, implementation details are in Section 4.

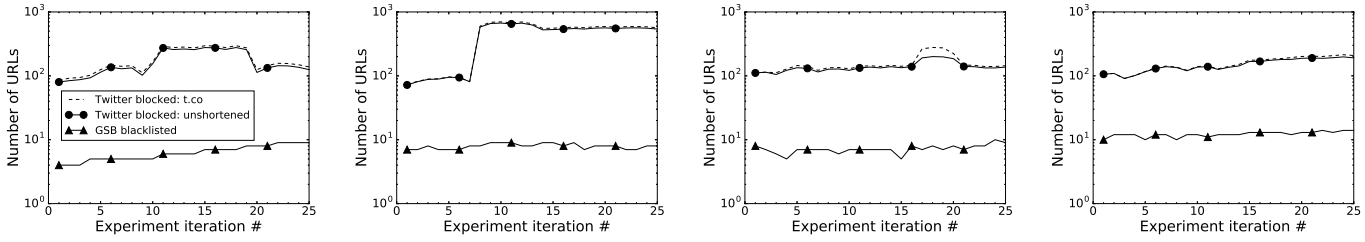
[8] https://help.twitter.com/en/rules-and-policies/twitter-rules

	Exp. #1	**#2**	**#3**	**#4**
Total tweeted URLs	**3M**	**3M**	**3M**	**3M**
Unique t.co (shortened)	1.4M	1.4M	1.3M	1.4M
Blocked by Twitter	373	765	313	260
Unique unshortened	1.2M	1.2M	1.2M	1.2M
Blocked by Twitter	350	734	230	241
Unique Blacklisted	**9**	**13**	**13**	**18**
GSB	9	13	13	16
GSB Phishing	5	11	12	14
GSB Malware	1	0	0	1
GSB Other	3	2	1	1
PT	0	0	0	2
OP	0	0	0	0
Blocked by Twitter and blacklisted	**1**	**1**	**3**	**4**
GSB	1	1	3	2
PT	0	0	0	2
OP	0	0	0	0

Table 4: Overview of experiments #1-4 showing total number of tweeted, blocked by Twitter (via t.co), and blacklisted URLs, in our dataset. Each experiment measures a 24-hour batch of tweeted URLs over a 7-day period during April & May 2019 (M = million).

Table 4 shows the overview of results for these experiments and are categorised into the 4 experiments that each ran for 7 days. When comparing the results in Tables 3 and 4 (i.e. comparing the 24-hour to 7-day experiment results), Table 3 shows that 0.01-0.02% unique unshortened URLs were blocked by Twitter when measured for 24-hours . In comparison, Table 4 shows 0.02-0.06% unique unshortened URLs were blocked by Twitter when measured for 7 days. Table 3 shows that 6-13% of tweeted URLs residing in GSB were also blocked by Twitter when measured for 24 hours. In comparison, Table 4 shows 8-23% of tweeted URLs residing in GSB were also blocked by Twitter when measured for 7 days. This increase suggests that the number of URLs filtered by Twitter and residing in GSB might increase as the duration of time measured since time of tweet increases. This may be due to a delay between tweeted URLs appearing in GSB and Twitter updating its list of URLs to block via *t.co*. This can be further investigated by analysing the number of tweeted URLs blocked by Twitter and residing in GSB during each of the 25 iterations of the experiments.

Figures 2a to 2c show the total number of unique *t.co* and unique unshortened URLs blocked by Twitter alongside how many tweeted URLs appeared in the GSB blacklist. These totals are measured during 25 iterations for each of the 4 experiments shown in Table 4. A single iteration takes approximately 7 hours to complete and involves analysing all 3 million tweeted URLs from the most recent 24-hour period prior to that experiment starting. These URLs are checked to determine if they are blocked by Twitter and also to determine if they appear in one of the 3 blacklists (GSB, OP, and PT). Each experiment (#1-4) measures a batch of 3 million tweeted URLs for 25 iterations - just over 7 days. Figures 2a to 2c show that Twitter's system to block URLs posted on its network is dynamic –



(a) Exp. #1 (b) Exp. #2 (c) Exp. #3 (d) Exp. #4

Figure 2: Experiments #1-4 showing total number of unique t.co and unique unshortened URLs blocked by Twitter alongside total number of tweeted URLs residing in GSB blacklist for each experiments' 25 iterations. Each experiment was measured over a 7-day period during April & May 2019. Shown on a logarithmic scale Y-axis.

it is actively blocking and unblocking URLs. At some point during the 7 day measurements, all 4 experiments see a temporary spike in the number of URLs blocked by Twitter. This shows that Twitter is actively monitoring its network for websites that breach its rules[9] and blocking such websites once discovered. We can also deduce that Twitter unblocks websites presumably once it is determined they no longer breach the rules.

In experiment #1 there are 4 tweeted URLs residing in GSB during iteration 1 compared to 9 tweeted URLs residing in GSB during iteration 25 (Figure 2a). This shows that the number of tweeted URLs residing in GSB has slightly more than doubled during the 7 day period of this experiment. Whereas experiment #2 started with 7 tweeted URLs residing in GSB and finished, at the end of day 7, with 8 tweeted URLs in GSB. The number of tweeted URLs in GSB throughout all 25 iterations of this experiment stayed relatively consistent with a range from 7 at its lowest to 9 at its highest (Figure 2b). A similar pattern can also be seen in experiment #3 which started with 8 tweeted URLs residing in GSB on the first iteration and finished on 10 with a range from 5 to 10. Interestingly, the first 4 iterations of this experiment measured the number of tweeted URLs in GSB as 8, 7, 6, and 5, respectively (Figure 2c).

Analysing the individual iterations shows how both Twitter filtering and GSB blacklisting are dynamic; the number of blocked or blacklisted URLs can change depending on a websites threat level. This means that Twitter can unblock a website once it no longer poses a threat to its users. We see a trend that the number of URLs blocked by Twitter does typically increase over time, although not necessarily by that much – and in some cases the volume of URLs blocked by Twitter may temporarily increase due to a large attack campaign. We see that the volume of tweeted URLs residing in GSB does not necessarily increase as time passes in all cases. This suggests that running experiments for longer than 7 days may not yield a significant increase in the number of URLs blacklisted by GSB.

We do see a slight increase in the percentage of URLs blocked by Twitter that are also blacklisted by GSB between the 24-hour measurement and 7-day measurement. The 24-hour measurement experiment saw, on average, 10% of GSB blacklisted URLs also blocked by Twitter. Whereas the 7-day measurement experiment saw, on average, 13.5% GSB blacklisted URLs also blocked by Twitter. However, the number of blacklisted URLs in the 7-day experiments is significantly lower than the 24-hour experiments therefore a comparison between the two percentages is not entirely equal.

Key Findings: Our results in this subsection show that fewer phishing and malware URLs were posted to Twitter during these 2018-19 measurement experiments, compared to our 2017 results. This suggests that Twitter may have become more effective at blocking blacklisted URLs at time of tweet. Additionally, we see that Twitter does indeed implement a "time of click" defence strategy to protect users from websites which break its rules. This may be because phishing/malware attacks are only part of a large set of rules defined by Twitter[10] – therefore Twitter may prioritise other violations of their rules. Despite this reduction in phishing and malware URLs posted to Twitter, our measurement experiments in 2018-19 see that Twitter's "time of click" filtering (via *t.co*) only blocks approximately 12% of blacklisted phishing and malware URLs – therefore still leaving its users exposed to potential cyber security attacks.

6 DISCUSSION

In this paper we carried out 2 key experiments:

(1) Measure number of phishing and malware URLs posted to Twitter. Also, to assess impact, measure clicks from Twitter users to these blacklisted URLs
(2) Measure Twitter Filtering: does Twitter filter URLs (via its *t.co* URL shortener)? If so, does Twitter block blacklisted URLs?
    - 24-hour measurement period
    - 7-day measurement period

Our results from (1) show that, during our 2-month measurement timeframe in 2017, over 10,000 unique phishing and malware URLs were posted to Twitter. The impact of these blacklisted URLs appearing on Twitter was that Twitter users generated over 1.6 million clicks to these potentially harmful websites. Our results from (2) show that, over a 3-month period in 2019, Twitter filtered over 14,000 tweeted URLs via its URL shortener (*t.co*). However, only 2% of these filtered URLs were blacklisted phishing and malware URLs.

[9] https://help.twitter.com/en/rules-and-policies/twitter-rules

[10] https://help.twitter.com/en/rules-and-policies/twitter-rules

We see that, during a 24-period, Twitter filtered 6-10% of blacklisted URLs. This increased to 8-30% when measured over a 7-day period. This suggests that Twitter's "time of click" URL filter does improve slightly as time increases, but that a large number of blacklisted phishing and malware websites are not filtered. It is important to note that our 2017 and 2019 datasets were collected using the same sampling strategy.

Our results from these 2 experiments show that, overall, there are fewer phishing and malware URLs being posted to Twitter in 2019 when compared to 2017 and 2010 results. We also see that Twitter adopts both a "time of tweet" and "time of click" defence strategy to protect its users not only from phishing and malware attacks, but also from any user or URL in violation of its set of rules[11]. Twitter is filtering large numbers of tweeted URLs, at time of click, but it appears the majority of these blocked URLs are for violations of its rules other than phishing/malware attacks. Twitter appears to have improved its "time of tweet" defence strategy, since 2010 and 2017, and is perhaps relying on this defence strategy to protect its users from phishing and malware attacks.

6.1 Limitations

There are some limitation to the data sources used in our measurement experiments. These limitations include: Twitter's data feed is a "small sample" of all global tweets, the GSB API blacklist is limited to 10,000 daily lookup per day, and our Twitter URL filter checking system cannot send too many HTTP requests per second because this would flood Twitter's servers. To compensate for these limitations we planned our methodology and built an infrastructure in a way that, to the best of our ability, allows us to carry out accurate measurements to answer our research questions.

We do not detect tweets containing phishing or malicious URLs that never make it into GSB. Therefore GSB is our "ground truth". We attempt to mitigate this by using the Open Phish and Phish Tank blacklists.

There are limitations to carrying our measurement experiments in an environment which evolves quickly and constantly. This makes it challenging to directly compare results obtained in 2010 and 2017 to results carried out in 2018-19. We mitigate this by setting up experiments in similar ways to previous studies, comparing figures from previous studies in percentages, and exploring other aspects that may have altered the environment which we are measuring.

7 CONCLUSION

This paper measured the effectiveness of Twitter's URL shortener (t.co) at protecting users from phishing and malware attacks. In 2017 we analysed over 182 million URL containing tweets collected from Twitter's Stream API over a 2-month period, and compared these URLs against 3 popular social engineering, phishing, and malware blacklists. Our main findings were that over 10,000 unique blacklisted phishing and malware URLs were posted to Twitter during this timeframe. This lead to over 1.6 million clicks which came directly from Twitter users – therefore exposing people to potentially harmful cyber attacks.

In 2018-19 we investigated if Twitter filters URLs posted to its platform. We discovered a reduction in the number of phishing and malware URLs posted to Twitter in 2018-19 compared to 2017 – suggesting an improvement in Twitter's effectiveness at filtering blacklisted URLs at time of tweet. However, we also discovered that only about 12% of blacklisted phishing and malware URLs – which were not blocked at time of tweet and therefore posted to the platform – were blocked by Twitter in 2018-19. Our results indicate that, despite a reduction in the number of blacklisted URLs posted to the social network, Twitter's URL shortener is not particularly effective at filtering phishing and malware URLs – therefore people are still exposed to these cyber attacks on Twitter.

ACKNOWLEDGMENTS

Research supported by the UK EPSRC grant EP/K035584/1 (Centre for Doctoral Training in Cyber Security). We would like to thank Kenny Paterson and Lorenzo Cavallaro for their contributions, and the anonymous reviewers for their insightful comments and observations.

REFERENCES

[1] Tim Armstrong. 2011. Twitter – Malware through time. https://securelist.com/twitter-malware-through-time/29775/. (2011).
[2] Salman Aslam. 2018. Twitter by the Numbers: Stats, Demographics & Fun Facts. https://www.omnicoreagency.com/twitter-statistics/. (2018).
[3] Simon Bell, Kenny Paterson, and Lorenzo Cavallaro. 2019. Catch Me (On Time) If You Can: Understanding the Effectiveness of Twitter URL Blacklists. *arXiv preprint arXiv:1912.02520* (2019).
[4] Christina Bonnington. 2018. Twitter is promoting a 'get verified' phishing scam. https://www.dailydot.com/debug/twitter-promoted-phishing-site/. (2018).
[5] Sidharth Chhabra, Anupama Aggarwal, Fabricio Benevenuto, and Ponnurangam Kumaraguru. 2011. Phi. sh/$ ocial: The Phishing Landscape Through Short URLs. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*. ACM, 92–101.
[6] Rachna Dhamija and J Doug Tygar. 2005. The battle against phishing: Dynamic security skins. In *Proceedings of the 2005 symposium on Usable privacy and security*. ACM, 77–88.
[7] Rachna Dhamija, J Doug Tygar, and Marti Hearst. 2006. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 581–590.
[8] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. 2008. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1065–1074.
[9] ESET. 2016. First Twitter-controlled Android botnet discovered. https://www.welivesecurity.com/2016/08/24/first-twitter-controlled-android-botnet-discovered/. (2016).
[10] Aleh Filipovich. 2014. gglsbl. https://github.com/afilipovich/gglsbl/. (2014).
[11] FTC. 2010. Twitter Settles Charges that it Failed to Protect Consumers' Personal Information; Company Will Establish Independently Audited Information Security Program. https://www.ftc.gov/news-events/press-releases/2010/06/twitter-settles-charges-it-failed-protect-consumers-personal. (2010).
[12] Google. 2015. Safe Browsing protection from even more deceptive attacks. https://security.googleblog.com/2015/11/safe-browsing-protection-from-even-more.html. (2015).
[13] Google. 2018. Safe Browsing. https://safebrowsing.google.com/. (2018).
[14] Google. 2018. Transparency Report - Safe Browsing: malware and phishing. https://transparencyreport.google.com/safe-browsing/overview. (2018).
[15] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. 2010. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 27–37.
[16] Tom N Jagatic, Nathaniel A Johnson, Markus Jakobsson, and Filippo Menczer. 2007. Social phishing. *Commun. ACM* 50, 10 (2007), 94–100.
[17] Marc Kührer and Thorsten Holz. 2012. An empirical analysis of malware blacklists. *PIK-Praxis der Informationsverarbeitung und Kommunikation* 35, 1 (2012), 11–16.
[18] Marc Kührer, Christian Rossow, and Thorsten Holz. 2014. Paint it black: Evaluating the effectiveness of malware blacklists. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 1–21.

[11] https://help.twitter.com/en/rules-and-policies/twitter-rules

[19] Ponnurangam Kumaraguru. 2009. *Phishguru: a system for educating users about semantic attacks.* Carnegie Mellon University.

[20] Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. 2010. Teaching Johnny not to fall for phish. *ACM Transactions on Internet Technology (TOIT)* 10, 2 (2010), 7.

[21] Sangho Lee and Jong Kim. 2012. WarningBird: Detecting Suspicious URLs in Twitter Stream.. In *NDSS*, Vol. 12. 1–13.

[22] Christian Ludl, Sean McAllister, Engin Kirda, and Christopher Kruegel. 2007. On the effectiveness of techniques to detect phishing sites. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 20–39.

[23] Federico Maggi, Alessandro Frossi, Stefano Zanero, Gianluca Stringhini, Brett Stone-Gross, Christopher Kruegel, and Giovanni Vigna. 2013. Two years of short urls internet measurement: security threats and countermeasures. In *proceedings of the 22nd international conference on World Wide Web*. ACM, 861–872.

[24] Ryan Naraine. 2018. Twitter turns to Google for help with malware attacks. http://www.zdnet.com/article/twitter-turns-to-google-for-help-with-malware-attacks/. (2018).

[25] OpenPhish. 2018. OpenPhish - Phishing Intelligence. https://openphish.com/. (2018).

[26] Oracle. 2018. MySQL. https://www.mysql.com/. (2018).

[27] Bryan Parno, Cynthia Kuo, and Adrian Perrig. 2006. Phoolproof phishing prevention. In *Financial Cryptography*, Vol. 4107. Springer, 1–19.

[28] PhishTank. 2018. Friends of PhishTank. https://www.phishtank.com/friends.php. (2018).

[29] PhishTank. 2018. PhishTank | Join the fight against phishing. https://www.phishtank.com/. (2018).

[30] Python. 2018. Requests: HTTP for Humans. http://docs.python-requests.org/en/master/. (2018).

[31] Steve Sheng, Mandy Holbrook, Ponnurangam Kumaraguru, Lorrie Faith Cranor, and Julie Downs. 2010. Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 373–382.

[32] Steve Sheng, Bryant Magnien, Ponnurangam Kumaraguru, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. 2007. Anti-phishing Phil: the design and evaluation of a game that teaches people not to fall for phish. In *Proceedings of the 3rd symposium on Usable privacy and security*. ACM, 88–99.

[33] Steve Sheng, Brad Wardman, Gary Warner, Lorrie Faith Cranor, Jason Hong, and Chengshan Zhang. 2009. An empirical analysis of phishing blacklists. *Proceedings of Sixth Conference on Email and Anti-Spam (CEAS)* (2009).

[34] SQLite. 2018. SQLite Home Page. https://www.sqlite.org/. (2018).

[35] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. 2011. Suspended accounts in retrospect: an analysis of Twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 243–258.

[36] Tweepy. 2018. Tweepy: An easy-to-use Python library for accessing the Twitter API. http://www.tweepy.org/. (2018).

[37] Twitter. 2018. The Twitter Rules. https://twitter.com/rules. (2018).

[38] TwitterCounter. 2018. Twitter Top 100 Most Followers. https://twittercounter.com/pages/100. (2018).

[39] Alex Hai Wang. 2010. Don't follow me: Spam detection in Twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*. IEEE, 1–10.

[40] WebProNews. 2012. Google Discusses Its Safe Browsing Record. https://www.webpronews.com/google-discusses-its-safe-browsing-record-2012-06/. (2012).

[41] Min Wu, Robert C Miller, and Simson L Garfinkel. 2006. Do security toolbars actually prevent phishing attacks?. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 601–610.

[42] Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong. 2006. Phinding phish: Evaluating anti-phishing tools. In *Tech Report: CMU-CyLab-06-018*. ISOC.

[43] Yue Zhang, Jason I Hong, and Lorrie F Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 639–648.