To Err.Is Human: Characterizing the Threat of Unintended URLs in Social Media

Beliz Kaleli
Boston University
bkaleli@bu.edu

Brian Kondracki
Stony Brook University
bkondracki@cs.stonybrook.edu

Manuel Egele
Boston University
megele@bu.edu

Nick Nikiforakis
Stony Brook University
nick@cs.stonybrook.edu

Gianluca Stringhini
Boston University
gian@bu.edu

***Abstract*—To make their services more user friendly, online social media platforms automatically identify text that corresponds to URLs and render it as clickable links. In this paper, we show that the techniques used by such services to recognize URLs are often too permissive and can result in unintended URLs being displayed in social network messages. Among others, we show that popular platforms (such as Twitter) will render text as a clickable URL if a user forgets a space after a full stop at the end of a sentence, and the first word of the next sentence happens to be a valid Top Level Domain. Attackers can take advantage of these unintended URLs by registering the corresponding domains and exposing millions of Twitter users to arbitrary malicious content. To characterize the threat that unintended URLs pose to social media users, we perform a large-scale study of unintended URLs in tweets over a period of 7 months. By designing a classifier capable of differentiating between intended and unintended URLs posted in tweets, we find more than 26K unintended URLs posted by accounts with tens of millions of followers. As part of our study, we also register 45 unintended domains and quantify the traffic that attackers can get by merely registering the right domains at the right time. Finally, due to the severity of our findings, we propose a lightweight browser extension which can, on the fly, analyze the tweets that users compose and alert them of potentially unintended URLs and raise a warning, allowing users to fix their mistake before the tweet is posted.**

I. Introduction

Social media platforms like Twitter, Facebook, and Linkedin are increasingly becoming the main way in which people obtain news and communicate with the rest of the world. Twitter, as one of the most popular of these platforms, was shown to be able to shape political campaigns [29] and even affect the number of citations that academic papers receive [23]. At the same time, the popularity of the platform has made it the ideal target for a variety of malicious activities, from spam [18], [25], [41], to online harassment [20], [32], [38], [46], to misinformation [16], [39], [53]. In the most recent high-profile example (July 2020), a sophisticated targeted attack showed the consequences that a compromise of the platform can have, resulting in multiple popular accounts posting links to a Bitcoin scam [8].

To improve the usability of their platforms, many online social networks automatically recognize links as users type them and render them as clickable. For example, if Twitter detects a URL in the text of a tweet, that part will be highlighted and users that have access to the tweet will be able to visit the link by just clicking on it. If the target Web page contains a so-called *Twitter Card*, a preview of the link will also be added to the tweet [2].

In this paper, we identify a potential attack vector in the way in which online social networks parse text and decide which parts of it should be rendered as clickable URLs. We show that it is not uncommon for social network users to supply text that is not supposed to be rendered as a clickable URL, yet the automated means by the social network platform mistakenly render it as such. Figure 1 depicts three typical examples of unintended URLs included in tweets. These examples showcase how unintended URLs enable third parties (well-meaning or malicious) to compromise the integrity of Twitter messages and expose the followers of popular Twitter accounts to arbitrary content. For example, when Rudy Giuliani (with 714.2K followers) tweeted the following on November 30, 2018: "`[...] as the President left for G-20.In July he indicted [...]` [11]," the missing space between "`G-20.`" and "`In`" caused Twitter to interpret that part of the sentence (`g-20.in`) as a URL (even though that domain was not registered at the time) and made that part clickable. The `g-20.in` domain was registered on the same day and served content that was critical of Mr. Giuliani and his policies.

We present the first analysis of the threat of unintended URLs on social media, with a particular focus on Twitter. We start by presenting a threat model detailing how unintended URLs can result from user posts which pose a threat to anyone who has access to these posts. As part of this process, we evaluate nine popular social media platforms and instant messaging applications (including Twitter, Facebook, LinkedIn, and Slack) to understand their behavior and identify the logic that they follow for expanding text into clickable URLs.

To characterize the threat of unintended URLs in the wild, we perform a large-scale measurement study on Twitter where we analyze public tweets from the 1% streaming API posted between January 2020 and July 2020, in search of unintended URLs. By manually analyzing tweets that include URLs, we build a ground truth dataset of intended and unintended URLs which we use to train a machine learning model that can differentiate between these two classes with 94% accuracy. We use this model to set up a pipeline that automatically identifies unintended URLs; over a period of 7 months, our pipeline identified 26,596 unintended URLs. In parallel to

Network and Distributed Systems Security (NDSS) Symposium 2021
21-24 February 2021, Virtual
ISBN 1-891562-66-5
https://dx.doi.org/10.14722/ndss.2021.24322
www.ndss-symposium.org

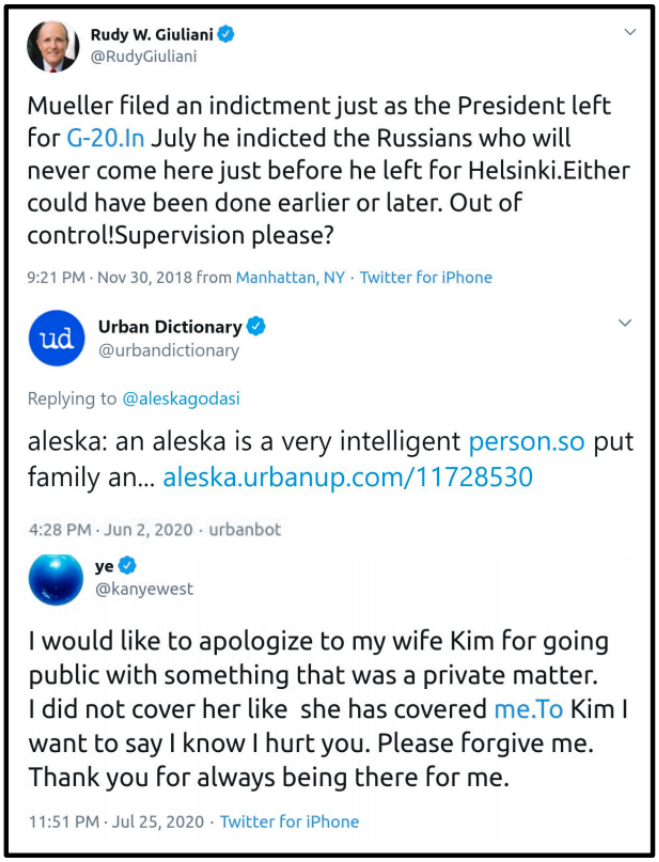

Fig. 1: *Examples of unintentional URLs posted by popular accounts on Twitter. (Rudy Giuliani: 714.2K followers, Urban Dictionary: 300.1K followers, Kanye West: 30.6M followers.)*

scanning the unintended URLs for evidence of abuse, we use our model to register a total of 45 domains, within hours after they were tweeted by popular accounts. Through this process, we find that unintended URLs receive an average of 103.65 visitors within one day of appearing in a tweet, with our most popular URL receiving 755 visits. Our results demonstrate the very real danger that unintended URLs pose to social media and how opportunistic attackers can hijack the content of social media posts, without compromising the user who posted them. Finally, to help users defend against unintended URLs, we implement our classifier in a browser extension that warns users whenever they type an unintended URL before posting a tweet, allowing them to correct their mistake. We will publicly release the source code of this browser extension upon publication of this paper.

In summary, the contributions of this paper are as follows:

- We show that automatically rendering unintended URLs can cause vulnerabilities in online services, and present a threat model in which an adversary abuses this phenomenon to launch various attacks on users who click on those unintentional links.
- We identify characteristics of unintended URLs by building a classifier that predicts those URLs in collected Twitter data and analyzing our results.
- We conduct a large-scale analysis of unintended URL websites using an automated data-collection infrastructure. By registering 45 domains that appear as unintended URLs on Twitter, we find that these domains receive spikes in traffic after a tweet is published. We also discover malicious content present on unintended URL websites.
- We build a Chrome extension for Twitter as a countermeasure to the problem at hand that can preemptively predict unintended URLs in tweets and warns the author.

II. Motivation and Background

In this section, we first present background information on the DNS system and then define unintended URLs. Next, we report on a number of experiments that we conducted to understand how different online services identify and render links. We then provide a rationale for why we focus on Twitter in this paper and provide background information about the platform. Finally, we present a threat model illustrating how an adversary could exploit unintended URLs for malicious gain.

A. Top Level Domains (TLD)

A top level domain (TLD) is the domain at the highest level in the hierarchical Domain Name System. The TLD is the last label in a domain name, e.g., in "`www.twitter.com`," the `com` label is the TLD. ICANN controls generic TLDs such as `.com` and `.net` whereas country-code TLDs (such as, `.it` and `.es`) are controlled by institutions within their respective countries. In this paper, we are mostly interested in the TLDs that are also valid dictionary words, either by chance (such as the article "`it`" also being the cc-TLD for Italy) or on purpose (such as "`online`" which is also a recent generic TLD available to the public).

B. Unintended URLs

In this paper, we consider an unintended URL as any string that is typed by a user on an online service and is rendered as a clickable link *without the user intending it to be a URL.* To be interpreted as a URL by an online service, a string has to contain at least two substrings, a dot and no spaces. In particular, the last substring needs to be a TLD.

A common reason that causes unintended URLs to occur is users making a typo [12] omitting a space after a dot at the end of a sentence, where the first word of the next sentence happens to be a TLD. For example, consider the following text extracted from a real tweet: "*I will always support you.You always support others.*" Here the unintended URL is `you.You` and it is converted to a clickable link by Twitter since "`You`" is an actual TLD name.

A similar error made by users is forgetting a space after a number when they are listing items. For example, in the following tweet: "*1.Team Leader - Outlet 2.Assistant Manager - Outlet 3.Executive Housekeeper*" the user forgot to put a space for all the list items. However, only "`1.Team`" is converted to a URL since "`Team`" is a real TLD whereas "`Assistant`" and "`Executive`" are not.

Another common way to cause an unintended URL is users putting a dot between two words in the same sentence instead of a space. Users sometimes use this method to emphasize these words. For instance, we excerpted the following from a real tweet: "*Dont you know who.I.am?*" The proper punctuation would be to put spaces between the words "*who*," "*I*" and "*am*". The author put dots instead of spaces and caused the unintended URL, "`who.I.am`" to be rendered as clickable by Twitter.

C. Understanding the Link Rendering Behavior of Online Services

In an effort to make their platforms more user friendly, online services automatically render the text that they identify as a URL, as clickable links. The precise algorithm for recognizing URLs in the user-provided text is something internal to each online service and, to the best of our knowledge, not known to the public.

In this section, we aim to understand the mechanisms used by different online services to identify and render clickable links. To this end, we test nine popular social media and instant messaging platforms for their automatic URL rendering functionality, applying 23 different test cases to each one. Our test cases are essentially strings without spaces containing at least one dot so that they can be interpreted as URLs. We try posting these strings in different platforms and observe whether the posted string is clickable (i.e., rendered as a URL by each platform). These strings are different combinations of the following conditions:

- Contains an existing domain name (XDOMAIN)
- Contains a non-existing domain name (NXDOMAIN)
- Contains a capital letter after a dot
- Contains a traditional TLD (e.g., `.net`)
- Contains a new TLD (e.g., `.dev`)
- Contains an invalid TLD (i.e., `.ttt`)
- Contains a number as the domain name
- Starts with "`www`"
- Contains a subdomain

Our test results are shown in Table I. The platforms that resulted in the exact same results for all 23 test cases are grouped together. For all tested platforms, test cases that contain NXDOMAIN and XDOMAIN (conjugates of each other) returned the same results. Thus, we conclude that none of the platforms perform a name server lookup before automatic rendering. We observe that Linkedin does not have any sort of mechanism to distinguish between an unintentional and an intentional URL. Twitter, Snapchat and Whatsapp Web do not render text as a link if the text contains an invalid TLD. Slack, Facebook and Whatsapp Mobile do not render the links with the recent TLD `.dev`. The reason could be that those platforms have not yet updated their TLD list in their servers since they also did not render the link with an invalid TLD. Telegram was the only platform that has an algorithm that checks the occurrence of a capital letter after a dot and the validity of a TLD. Skype and Google Hangouts only rendered a link if it starts with "`www`."

Our experiments show that the most permissive platforms are Linkedin, Twitter, Snapchat, and Whatsapp Web. To further understand the threats posed by unintended URLs we decide to focus on Twitter for the rest of this paper. Our reasons for this decision have to do with the nature of social media vs. instant messaging applications as well as the overall popularity of the Twitter platform. Specifically, the potential threat posed by unintended URLs on instant messaging apps is lower than on social media platforms, because only the users inside the chat can access the unintended URLs. In terms of popularity, Twitter is not only more popular than LinkedIn (Alexa Rank 50 vs. 63) but Twitter posts are public by default, thereby exposing more users to the threat of unintended URLs.

D. Twitter Features

In this section we briefly describe the Twitter features relevant to the issue of unintended URLs, to allow unfamiliar readers to follow the remaining of the paper.

a) Link Preview and Rendering: Twitter detects links in plain text and automatically renders them as URLs. As a user types a tweet, Twitter shows it as a link by turning that portion of the text blue. Twitter also provides a link preview functionality. If a linked website uses Twitter Cards [2], the posted tweet will include a link preview consisting of an image, title and description of the posted link. These types of previews are generated by Twitter after the user posts a tweet, whereas the link highlighting happens as users type their tweets.

b) Retweet Types: Retweeting is a mechanism that allows Twitter users to share another user's tweet and making it appear on their timeline, which is also accessible by the user's followers. As a result, retweeting increases the impact of the tweet because it makes it visible to more users. Twitter allows two different types of retweets. The first one is simple retweeting, in which the other user's tweet appears exactly as it is in the retweeter's timeline. This method increases the retweet count of a tweet by one each time a different user retweets. The second method is retweeting with a comment. Here, the original tweet is compressed in a box and the comment is shown as a regular tweet on top of this box. The original tweet loses its features, such as the link preview, highlighted links and images. This method does not increase the retweet count of a tweet.

E. Threat Model

Twitter does not have a mechanism to check spelling or punctuation and anyone can create an account and post tweets in seconds. According to our data gathering, approximately 400 million tweets are posted by users on a daily basis. Given the lack of native spell checking, the onus (or choice) of proper spelling falls unto the users. As a result, tweets including typos are not uncommon. Due to the aforementioned retweeting mechanism, a typo in a tweet of an unpopular user can still be shown to millions of users, if a popular user somehow discovers and retweets that typo-including tweet. The same goes for other types of unintended URLs.

Figure 2 provides an example attack scenario for the type of vulnerability studied in this paper. First, a Twitter

NX Test URLs	X Test URLs	Linkedin	Twitter Snapchat Whatsapp Web	Slack Facebook Whatsapp Mobile	Telegram	Skype Google Hangouts
notdomain.net	php.net	C	C	C	C	U
notdomain.Net	php.Net	C	C	C	U	U
notdomain.dev	web.dev	C	C	U	C	U
notdomain.Dev	web.Dev	C	C	U	U	U
sub.notdomain.net	windows.php.net	C	C	C	C	U
sub.Notdomain.Net	windows.Php.Net	C	C	C	U	U
sub.notdomain.dev	auth.web.dev	C	C	U	C	U
sub.Notdomain.Dev	auth.Web.Dev	C	C	U	U	U
5.net	123.net	C	C	C	C	U
5.dev	1.dev	C	C	U	C	U
www.notdomain.dev	www.web.dev	C	C	C	C	C
notdomain.notatld	-	C	U	U	U	U

TABLE I: *Unintended URL test results on different social media and messaging platforms. "U" depicts test URL is not clickable after posting whereas "C" depicts test URL is clickable after posting on the particular platform.*

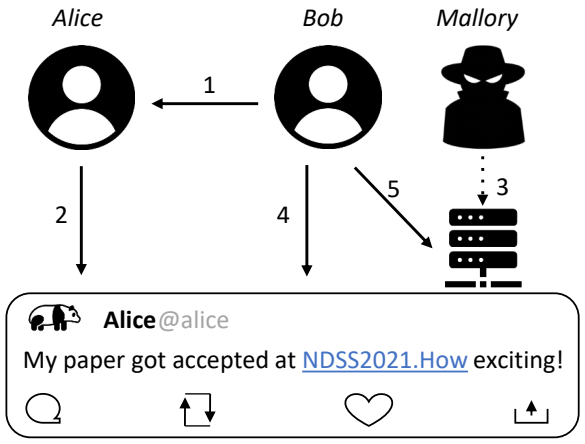

Fig. 2: *Attack scenario on how an attacker (Mallory) can weaponize the unintended URLs posted by a user (Alice) to expose Alice's followers (such as Bob) to malicious content.*

user (Bob) follows Alice on Twitter (Step 1) who, at some point, posts a tweet containing an unintended URL (Step 2). Mallory observes this, registers the unintended domain (`NDSS2021.how` in our example) controlling that URL, and points it to a malicious server (Step 3). Eventually, Alice's tweet appears on Bob's timeline (Step 4) who clicks on the link and is now exposed to Mallory's malicious content (Step 5). If Alice's followers retweet Alice's tweet (either because they did not pay attention to the URL or because they retweeted it before Mallory registered it) the reach of that unintended URL can increase exponentially.

III. Methodology

As we show in the previous sections, the threat of unintended URLs in social media is a real one, not just for the average Twitter user, but for accounts with millions of followers. To be able to characterize this phenomenon in the wild, we must first compile a ground truth dataset containing Tweets with intended and unintended URLs. We will then use this ground truth to train a classifier able to automatically identify unintended URLs on Twitter.

In this section, we describe our process for curating this ground truth dataset and the features that we use for our classifier. We then present the accuracy of the resulting classifier and how we use it to measure the phenomenon and abuse of unintended URLs in the wild. Our analysis pipeline is illustrated in Figure 3.

A. Automatically Detecting Unintended URLs on Twitter

To study the threat of unintended URLs on Twitter at scale, we first need to be able to automatically distinguish between unintended and intended URLs. To this end, we develop a machine learning model.

We manually label and analyze Twitter data to identify promising features for our model and develop a ground truth set. To build our model, we follow three steps: i) data gathering, ii) prefiltering, and iii) feature engineering. These steps are described in detail in the rest of this section.

a) Data Gathering: We collect the 1% sample of all public tweets posted worldwide using the Twitter streaming API [4] between January 2020 and July 2020. To build our ground-truth dataset, we extract five days of data from the collected tweets. This set consists of approximately 20M tweets of which around 3M (15%) containing URLs.

We use this set to gain insights about the differences between unintended URLs and intentional URLs. Our observations in this step helped us to identify prefiltering conditions and design model features.

b) Prefiltering: We first apply a prefiltering procedure to our set of tweets to remove links that are unlikely to be unintentional, based on our threat model and our preliminary observations. This step of our model-developing stage allows us to have a simpler and more accurate model without unnecessary features.

In this step, we first filter out the tweets that do not contain any URL since those tweets are not of interest for this work. This leaves us with approximately 3 million tweets per day. Then we apply the following filters:

- **URLs starting with "www"**: We discard the tweets that have URLs starting with "`www`" since "`www`" is

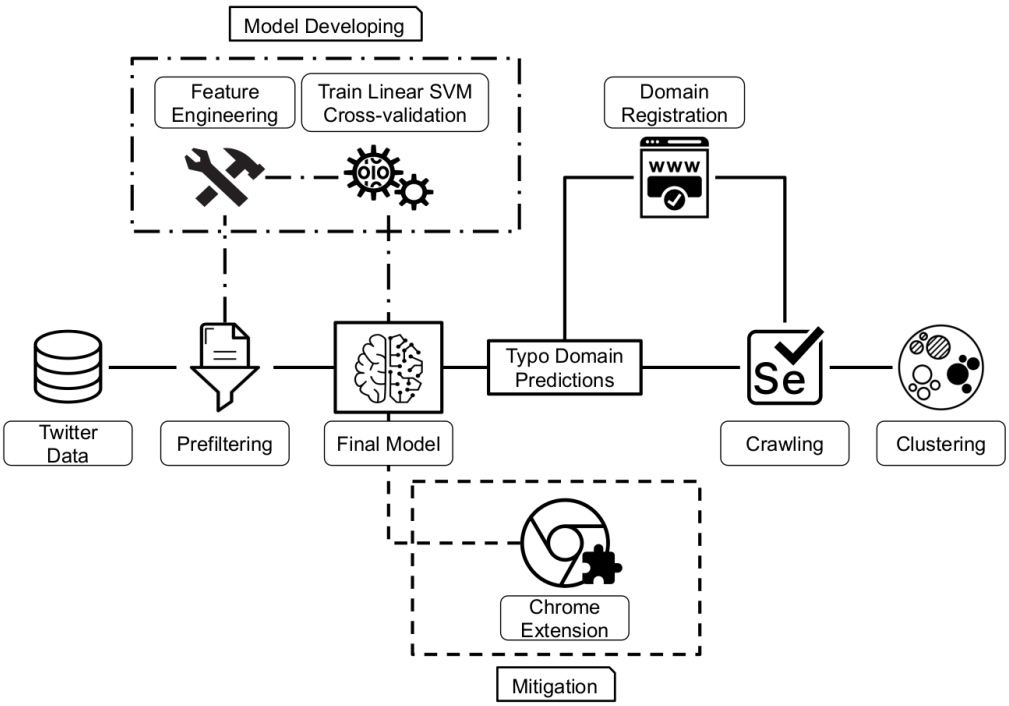

Fig. 3: *System Diagram: Twitter data is collected daily throughout the experiment. We extract five days of tweets from the Twitter 1% streaming API. We then apply prefiltering to this set. The resulting tweets are labeled manually to get the ground truth set and features are obtained. We then train a linear SVM classifier to obtain the final model. We use this model to get unintended domain predictions for the rest of the experiment. Then, we register unintended domains if suitable, and monitor the traffic to these domains. For those unintended domains that have a third party website pointing to them, we crawl their home pages and record screenshots. From the obtained information, we cluster unintended domains. Finally, we develop a mitigation by building a Chrome extension using our final model.*

not a dictionary word and therefore someone who explicitly types "`www`" is clearly intending to post a URL.

- **Non-English tweets**: For this paper, we focus on Twitter accounts that tweet in English. This allows us to reason about the intended/unintended nature of URLs using common grammar rules as well as later utilize NLP tools that work best on English corpus. Understanding whether unintended URLs happen in other languages is an interesting path for future work but outside the scope of this paper. After filtering out non-English texts, we keep approximately 1.1 million tweets.

- **URLs with paths and/or subdomains higher than third-level**: We eliminate a tweet if it only contains URLs with paths and/or subdomains higher than third-level. During our manual investigation of URL-including tweets we observe that in nearly all cases, URLs with paths and subdomains (higher than third-level) were clearly intentionally posted by users.

- **URLs having "`com`" or "`org`" as TLDs**: Even though "`com`" and "`org`" are part of the English dictionary ("`com`" as a prefix and "`org`" as an abbreviation) they are not particularly popular outside intentional URLs. As such, we filter any URLs that ended in these TLDs.

- **URLs having TLDs that are not dictionary words**: Given that our intuition is that unintended URLs are constructed by concatenating real words, we also filter URLs with TLDs that are not English dictionary words.

After prefiltering, we identify a set of 1,068 tweets that potentially contain unintended URLs. To determine which of these URLs are indeed unintended, we follow an inductive approach, with three authors of this paper discussing them until a good agreement is reached. After this process, our final ground truth set is labeled as 644 tweets containing intended URLs and 424 containing unintended URLs.

c) Feature Engineering: After determining our ground truth set, we aim to develop features that can characterize tweets containing intended and unintended URLs, with the goal of performing automated detection of the latter class. To determine the first batch of features, we go through our initial set of pre-filtered tweets and gather statistics for tweets containing unintended URLs such as common TLDs, DNS responses, string properties, location of the URL inside tweet text, etc. Then, to refine our features, we manually investigate our ground truth dataset to identify new features and have more accurate classification. We test our model on the labeled ground truth set to fine-tune the features. We repeat the process of investigating daily collected tweets and adding new features, and removing features with low effect until we are satisfied with the overall accuracy and the simplicity of our features. After this process is completed, we have the following features:

- **DNS Response.** We observe that many of the unintended URLs correspond to non-existing domains. To figure out whether a domain is registered, we check whether that domain had name server (NS) records. This allows us to conclude that a domain was either registered/un-registered, irrespective of whether that domain resolved when regular users were visiting (i.e., had A/AAAA records in place).

- **Sentence Segmentation.** If the first part of a link logically belongs to one sentence whereas the text of the remainder of that link belongs to the next sentence, this link has a high likelihood to be an unintended one. We use a tool called Deepsegment [5] to extract the sentences out of a tweet, treating URLs as regular text. Our preliminary results with the tool indicate that Deepsegment has a 97% segmenting accuracy if the punctuation is correct, 71% accuracy if the punctuation is partially correct and 53% accuracy if there is no punctuation in the tweet. For this binary feature, we mark it as positive if parts of the URL is at the end of one sentence and the remaining part forms the beginning of the next sentence.

- **String Properties.** We identify five characteristics of strings that are either indicative of intended or unintender URLs, and codify them as binary features. iThe following characteristics are indicative of intended URLs:
    - Any of the subdomains or the domain contains a dash.
    - Subdomains and domains are in camelcase (contains a mix of capital and non-capital letters).

  Whereas the following characteristics are indicative of unintended URLs:
    - The length of both the subdomains and domain are at most two (i.e., contains two or fewer characters).
    - Subdomains and domains are English words or numbers.
    - The link contains a non-capital letter followed by a dot followed by a capital letter (i.e., [a-z].[A-Z])

- **Repetition of URL.** Another binary feature is set to True/False depending on whether a URL appears more than once in the tweet. We consider it unlikely that a user will accidentally introduce the same URL twice in a single tweet.

- **Location of URL.** We use three different binary features identifying the location of the URL inside tweet text. The URL can be at the beginning, in the middle, or at the end of the text. URLs that are at the end of the text tend to be intended ones (such as when someone explains what they are posting and ending the tweet with a link to that post) whereas the ones that are in the middle tend to be unintended. Here, the full stops that users are using to mark the end of a sentence are more likely to be recognized as part of a URL, when whitespace is missing after them.

- **TLD Type.** Other than the prefiltering for "`com`" and "`org`" TLDs, we use ten binary features for the following ten different TLDs: "`net`," "`co`," "`gov`," "`it`," "`my`," "`no`," "`so`," "`you`," "`to`" and "`zip`." The reason for explicitly converting these TLDs into features is two-fold. First, the TLDs such as "it," "`my`," "`no`," "`so`," "`you`" and "`to`" are English words that are also likely to be used at the beginning of sentences. As such, URLs including these TLDs have a high likelihood of being unintended. Second, TLDs such as "`net`," "`co`," "`gov`" and "`zip`" are not usually used at the beginning of sentences and hence are likely to be found in intended URLs.

We train and test our model on our ground truth dataset. We experiment with a random forest classifier [31], a decision tree classifier [44], a k-nearest neighbors classifier [10] and support vector machines (SVMs) with different kernels [49]. We use 10-fold cross-validation to test the performance of each classifier and obtained the highest accuracy with an SVM model using a linear kernel. Our classifier uses binary features and outputs a binary number corresponding to two classes namely unintended (binary 1, positive) and intended (binary 0, negative).

The resulting classifier achieves 94% accuracy, 94.3% precision, 90.6% recall and 92.2% f1 score on our ground truth dataset. Using this classifier, we can now process the 1% tweet stream and record the tweets that the classifier predicts to be including unintended URLs. To make sure that our features do not overfit on our ground truth set, we perform a validation on independent data in Section IV-A, showing that our model achieves similar performance on an unseen dataset. After running our model, we use the identified URLs both to understand whether attackers are already abusing them but also, when possible, to register them so that we can quantify the number of Twitter users that attackers can victimize. Note that the overall goal for this classifier is to achieve reasonably high accuracy while keeping the features interpretable. We, therefore, do not experiment with neural-network-based classifiers that require significantly more ground truth and are difficult to interpret.

B. Unintended URL Crawling

Since unintended URLs in tweets can drive unsuspecting visitors to potentially malicious websites, we seek to determine the kind of content that they serve, as well as the extent of malicious activity that leverages this traffic source. Thus, we implement an automated URL crawling infrastructure that collects data on all unintended URLs uncovered by our classifier.

Our crawling infrastructure visits each website with an instrumented Chrome browser using Selenium [1], and records the following information: (1) webpage HTML, (2) screenshot, (3) TLS certificate, (4) redirect URL and IP address, (5) IP address information, (6) URL blacklist information, and (7) Alexa rank.

To reduce the ability of websites to fingerprint our crawler and cloak malicious content, we take measures to mask our use of browser instrumentation. This involves changing the HTTP User-Agent to appear as though the browser is Google Chrome on a Windows 10 desktop, as well as injecting JavaScript into each rendered page that modifies global JavaScript variables

(such as `navigator.webdriver`) to hide signs of browser automation.

Using the data gathered by our crawling infrastructure, we cluster webpages based on the perceptual hashes of their screenshots [54]. We then manually label these clusters based on the content that was recorded by our crawlers.

C. Unintended URL Registration

To determine the traffic directed towards unintended URLs present in tweets, we register a subset of available domains (i.e., unintended domain names that are part of tweets and are also available for registration) and forward traffic to web servers under our control. There, we record information about each request including the client's IP address and request headers.

To decide which unintended domains to register, we manually analyze all the unintended URLs from each previous day and focus on the ones that we reason will reach the most users. Specifically, we determine the reach of a tweet by observing the follower count of the tweet's author as well as the number of the tweet's likes and retweets.

In parallel to the tweets that our classifier discovers as part of the 1% of global tweets that Twitter offers via its API, we also deploy an infrastructure that specifically monitors the tweets of the most popular Twitter users, searching for tweets that include domains which are available for registration. In total, we monitor the tweets of 20,000 users with follower counts ranging from 11 thousand to 118 million. Whenever we encounter an unregistered domain tweeted by these top accounts, the infrastructure — in real-time — automatically registers that domain and forwards traffic to our webservers. This allows us to observe traffic from tweets as soon as they are posted, even in the cases where users later discover their error and delete the unintended-URL-including tweets.

D. Ethical Considerations

Analyzing social media activity has important ethical implications. In this work, we only analyze data that is publicly posted on Twitter. For our registered unintended domains, we do not interact in any way with the users who click on those links and visit the web pages that we set up. We merely count the number of visits that a page receives. Additionally, we argue that by registering these domains we are reducing potential harm, since we prevent attackers from leveraging them for malicious purposes. Since all the data that we use is public and we do not interact with users in any way, this research is not considered as human subjects research by our institution's IRB.

IV. Evaluation

In the following, we first present an experiment to validate our classifier's accuracy on unseen data. Then, we run our model on the entire dataset and analyze the detected unintended URLs in detail, together with the characteristics of the accounts that posted them. We also report on our experiment involving the registration of 45 domains that appear as unintended URLs in our dataset and were available for registration. This experiment allowed us to characterize the type and amount of traffic that attackers can get when they weaponize unintended URLs.

A. Validation of the Classifier

In Section III-A0c we performed a 10-fold cross validation on our ground truth set and obtained the following average performance scores: 94% accuracy, 94.3% precision, 90.6% recall and 92.2% f1 score. To be that our classifier can generalize to unseen data and rule out overfitting concerns, however, we want to test our classifier on a different dataset than the one the features were developed on. To this end, we collect a week of unseen tweets, extract the URLs and apply our prefiltering mechanism on this new set. After prefiltering, we manually label the resulting URLs as intentional/unintentional following the same process used to determine our ground truth. This set consists of 1302 unintended and 1829 intended URLs. We then train our classifier on the ground truth set, and test it on this new dataset to get predictions. By comparing the model's predictions and the manual labels of this set we obtain 93.3% accuracy, 94.5% precision, 89.1% recall and 91.7% f1 scores. Since we obtained similar performance scores on our ground truth set and a completely unseen dataset, we conclude that our feature set does not overfit on the training set, and can therefore be run in the wild. We also observe that around 40% of false positive URLs originate from spam tweets belonging to betting, cryptocurrency, and gaming websites such as "`IQ.Cash`," "`GG.bet`," and "`iBlocks.Games`." Particularly, we observe the same spam URL appearing among the false positives more than once because the spam Twitter accounts post almost exact tweet containing the URL many times throughout the day. Our analysis shows that 25% of false negatives contain either "`.so`" or "`.in`" as TLD. These two words are commonly used to start sentences in spoken English and having "`.so`" as TLD is among our features, however, that was not enough by itself to classify these URLs correctly.

B. Common Properties of Unintended URLs

We ran our daily pipeline for 7 months, between January 2020 and July 2020, recording a total of 26,596 unintended URLS. Figure 4 shows the daily number of unintended URLs on Twitter identified by our analysis pipeline. Overall, 19,195 (72% of the total) domains resulting from unintended URLs are non existent (NXDOMAINs), while 7,401 (28% of the total) domains are existing (XDOMAINs). On average, 75% of the unintended URLs posted every day are non-existing. This result is expected since an unintentionally posted link would only be an existing domain name due to pure coincidence. Our results highlight the threat of adversaries opportunistically registering these unintended domains and populating them with arbitrary malicious content.

a) Unintended URL placement: We next look at the placement of unintended URLs in tweets. We find that 23,813 unintended URLs are in the middle of the tweet (89.5%), 2,388 (9%) are at the beginning and 395 (1.5%) reside at the end of the text. As we said previously, one of the reasons why unintended URLs tend to appear in the middle of tweets is that users forget to put a space between two sentences, creating an unintended URL that combines the last word of the previous sentence and first word of the next sentence. This mistake may also cause unintended URLs at the beginning or at the end of

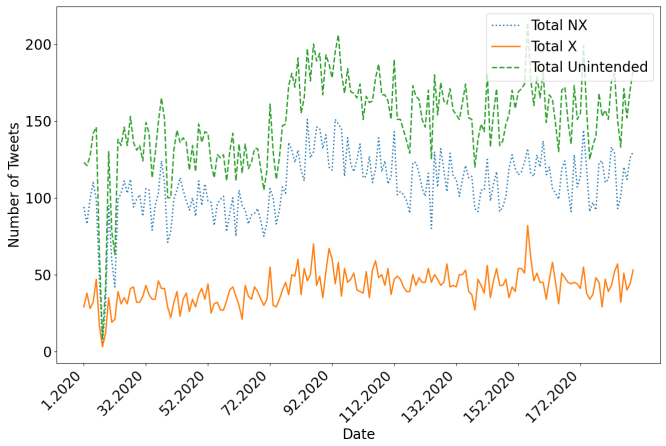

Fig. 4: *Number of unintended URLs per day as detected by of our model over the whole duration of the experiment. Unregistered domains (NXDOMAINs) dominate the set of unintended URLs posted on Twitter.*

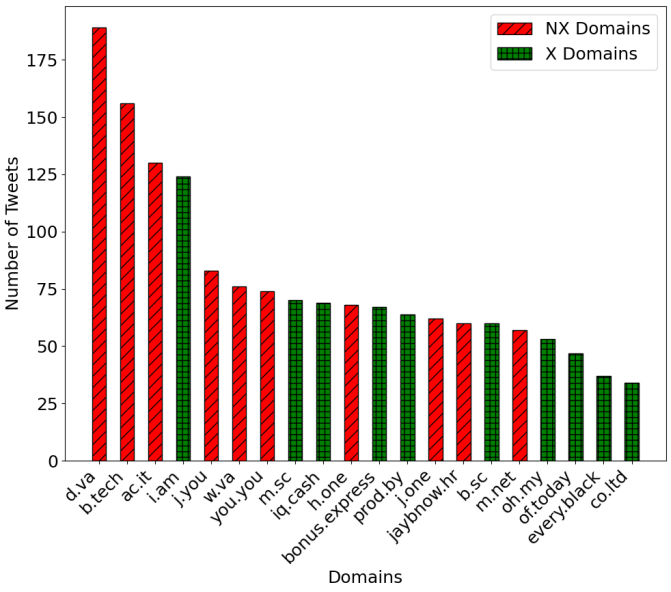

Fig. 5: *Number of tweets vs. Domain Names: The total number of tweets posted containing the unintended domain name throughout our experiments. The graph shows the top 10 NX (non-existing) and X (existing) domains with the highest count.*

a sentence if the first or last sentence only contains one word. However, the likelihood of this happening is low since single-word sentences are uncommon. We observe that if a user wants to intentionally post a link, one of the common ways to do that is to first write about the website and then place the link at the end of the sentence usually following a punctuation character (e.g., ":"). Hence, our results are in line with this observation.

b) Common Domain Names: We next focus on the domain names that are frequently rendered by Twitter when expanding unintended URLs. Our goal is to observe the most common mistakes and discover any active malicious campaigns abusing them. Figure 5 shows the top 10 non-existing and existing domain names with the highest occurrence throughout our experiments.

Four of the existing domains shown in Figure 5 are false positives: "`Iq.cash`," "`Bonus.express`," "`Of.today`" and "`Every.black`." The domain "`Bonus.express`" is promoted through spam and all 67 tweets are posted by the same account. Similarly, all 47 tweets containing "`Of.today`" are posted by the same account promoting a website with many subdomains. Moreover, tweets with "`Every.black`" domain are posted by 11 different accounts that all appear to belong to the owners of this domain in an attempt to advertise the website. The tweets containing the domain "`Iq.cash`" can also be considered spam since they all aim to promote a specific crypto currency. These domains, especially "`Of.today`" and "`Every.black`" carry many of the unintended URL properties that we identified in our preliminary analysis, and could be true positives in other scenarios.

Among the domains in the top 20, 6 of them are abbreviations, which are commonly typed by users and result in unintended URLs: "`B.tech`," "`W.va`," "`B.sc`," "`Prod.by`," "`M.sc`" and "`Co.ltd`."

Unexpectedly, our analysis also revealed an additional cause for unintended URLs which we had not considered when we started this work. Instagram allows users to have a dot inside their profile names, while Twitter does not. This semantic difference makes it so that any time a Twitter user posts an Instagram profile name in their tweets, this may result in an unintentional URL if the final part of the profile name happens to be a valid TLD. "`D.va`," "`J.you`," "`H.one`," "`Jaybnow.hr`" and "`J.one`" are examples of these incorrectly expanded Instagram handles. Again, since users do not intend to include a link when typing these profile names, we consider them as unintended URLs which can be weaponized by attackers by merely registering them.

Among the top 20 domain in Figure 5, four are commonly appearing in tweets as a consequence of users making typos (e.g., using a dot instead of space to separate between two words): "`Ac.it`," "`You.you`," "`I.am`" and "`Oh.my`." In total, we record 375 tweets for these URLs. To determine if they actually are unintended URLs or false positives, we check a random sample of 10 tweets including each domain (40 in total). We find that all the sampled tweets that include "`Ac.it`," "`You.you`" and "`Oh.my`" contain links due to typographical errors. Hence, we conclude that these three domain names are unintended URLs. For "`I.am`" we observe three different causes. This domain name is expanded when users mention "`will.i.am`" to refer to the name or the website of a famous American rapper. This domain can also be used inside an Instagram handle or may occur due to a typo. Among the recorded 124 tweets containing the domain "`I.am`," we observe that 73 of them are referring to the American rapper "`will.i.am`" (and are therefore false positives), 20 of them are typos, 3 are Instagram handles and 28 of those tweets are unavailable due to the author making their account private or deleting the tweet, possibly after realizing their mistake.

c) Common TLDs: During our initial assessment of the issue of unintended URLs in social media, we identified several TLDs (e.g., "`it`," "`my`," "`no`," "`so`," "`you`" and "`to`")

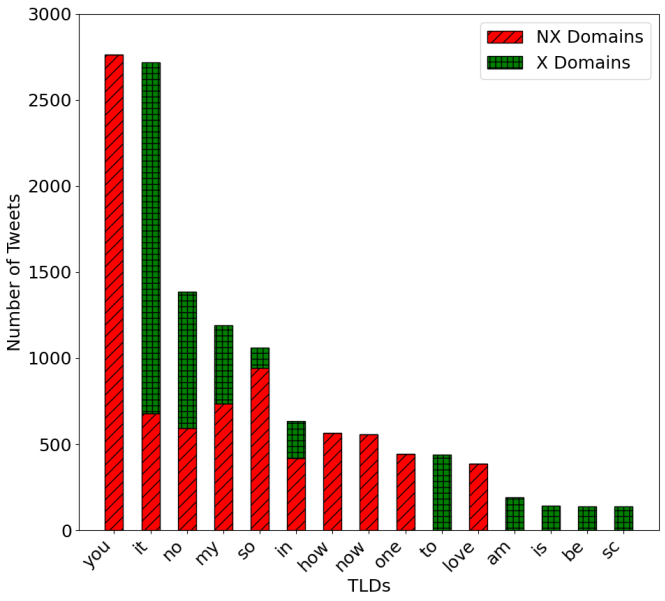

Fig. 6: *Number of tweets vs. TLDs: The total number of tweets posted containing the TLD name throughout our experiments. Red patch of a bar shows the contribution from NX (non-existing) domains and the green patch shows the contribution from X (existing) domains.*

that commonly occur in unintended URLs and we used them as features in our machine learning model (Section III-A). Figure 6 shows the top 15 TLDs that are used in unintended URLs in our dataset.

Our findings are aligned with our initial assessment since the resulting top 15 TLDs contain the 6 TLDs that we used as features. Most of the TLDs plotted in Figure 6 are English dictionary words that are commonly used at the beginning of a sentence. For example, "`you`," "`it`," "`no`," "`my`," "`so`," "`in`," "`how`," "`now`," "`one`," "`to`," "`love`" and "`be`" are usually used at the beginning of a regular sentence whereas "`am`" and "`is`" are used at the beginning of questions. Only "`sc`" is not used as such but, as mentioned earlier, we can straightforwardly conclude that the high occurrence count is due to the following abbreviations: "`B.sc`" and "`M.sc`." From Figure 6 we notice that unintended URLs with TLDs that happen to be dictionary words tend to point to non-existing domains (e.g., "`.you`," "`.how`"). TLDs that are dictionary words but also correspond to a country code (e.g., "`.it`," "`.no`") point to a mixture of existing and non-existing domains.

The "`.you`" TLD is an interesting case because, while the TLD is valid, registrations of domains ending in "`.you`" are not yet available to the public. As such, even though Twitter renders `.you`-ending domains as clickable links, none of these domains actually serve any content. While this limits the immediate exploitation of these unintended URLs, it also means that, when registrations open to the public, all of these domains will be exploitable, essentially overnight.

C. Impact of an Unintended URL

Intuitively, one would expect that not all unintended URLs have the same impact on Twitter users. Instead, their ability to reach users is influenced by how popular was the account

	Unintended URL	Original Author	Follower Count
NX	SEE.YOU	Harry_Styles	34M
	Kobe.Osaka	Harry_Styles	34M
	c.bank	Reuters	22M
	im.mo	9GAG	16M
	kuba.black	9GAG	16M
	raminta.art	9GAG	16M
	PERMANENTLY.MORTGAGE	iamcardib	13M
	sign.Hair	iamcardib	13M
	shake.You	iamcardib	13M
	unexpected.Love	iamcardib	13M
X	thing.It	Oprah	42M
	moment.In	deepikapadukone	27M
	people.It	iamcardib	13M
	tongue.Today	iamcardib	13M
	pregnant.My	iamcardib	13M
	street.It	JKCorden	11M
	Rt.live	TechCrunch	10M
	movie.Best	dhanushkraja	8.9M
	airbnb.To	anandmahindra	7.8M
	violence.In	marcorubio	4.2M

TABLE II: *unintended URLs with highest original author follower count.*

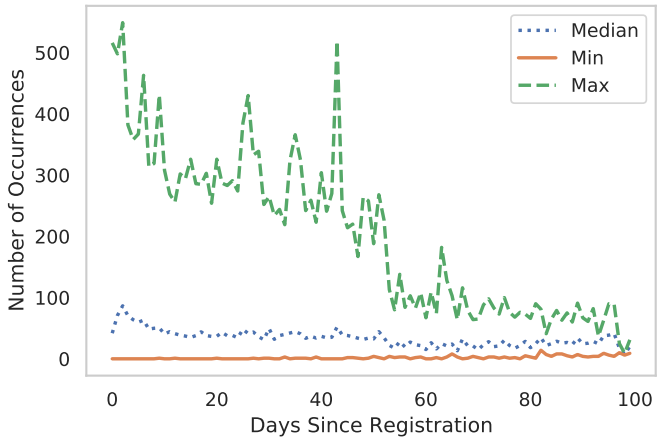

Fig. 7: *Average daily requests for all unintended domains registered.*

that posted it as well as how many times the tweet was shared (retweeted). In this section, we aim to investigate the relationship between the popularity of a Twitter account and the impact of the unintended URLs that they post.

In Table II we record the top 20 unintended URLs that we recorded, ranked by account popularity in terms of their number of followers. The follower count of these authors ranges between 34M and 4.2M. For example, when Harry Styles tweeted "`SEE.YOU`" from his Twitter account, there are potentially 34M users who can view the tweet on their timelines and click on the unintended URL. If an attacker registered this domain name shortly after the tweet is posted, up to 34M victims (ignoring re-tweets that could further enlarge the users who are exposed to the tweet) could have visited their website and be directed to a malicious page. Therefore we say that the magnitude of the audience, in this case, is at least 34M.

As described in Section III-C, part of our study involves the registration of the domains corresponding to unintended URLs so that we can more precisely quantify the number of users that attackers could victimize.

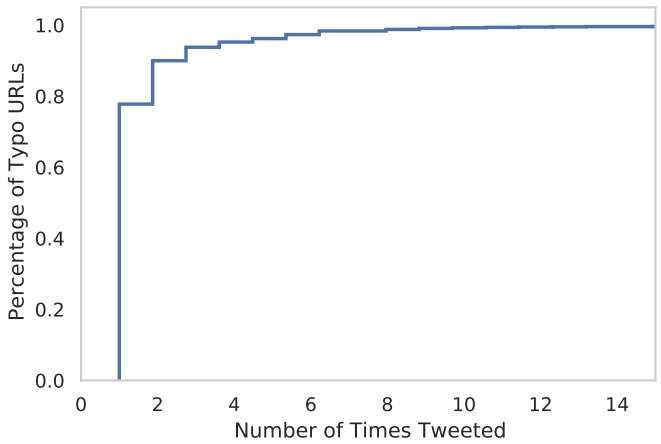

Fig. 8: *Frequency unintended URLs are included in tweets throughout our study.*

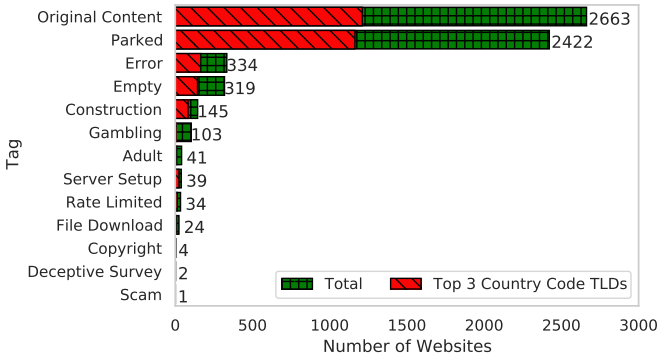

Fig. 9: *Content of unintended URL websites.*

Figure 7 shows the number of daily requests to the 45 unintended domains that we register during our data collection period. Our data shows that requests for unintended URLs are at their peak immediately after appearing in a tweet, followed by a downward trend each day thereafter. The magnitude of the spike in traffic after a tweet's publishing is dependent on the following of the tweet's author. Hence, we observe a number of tweets from users with large followings directing hundreds of visitors to our web server promptly after registration and continually providing residual traffic for the days following. The decrease in traffic over time to the domains we register is indicative of the effect unintended URLs located in tweets have on driving views to websites. Attackers who register domains before (e.g., by predicting common unintended URLs) or shortly after they appear in a tweet will receive significantly more requests compared to registering the same domains a few days later.

We also discover isolated spikes in requests to unintended domains in the days following registration. These spikes can be attributed to the same unintended URLs appearing in additional tweets. Figure 8 shows the number of times that unintended URLs were included in tweets throughout our entire dataset. We find that 20% of these URLs are tweeted more than once, hence driving new traffic to the websites of potential attackers.

D. Unintended Domain Properties

As we discussed earlier, most unintended URLs (72%) belong to non-existing (i.e., available for registration) domain names. In this section, we focus on the minority of URLs (28%) pointing to domains that were registered, in an effort to understand what kind of content is hosted on the websites and whether these domain registrations are coincidental as opposed to motivated by tweets including unintended URLs. During our period of experiments, our data collection infrastructure visited and recorded information for 15,301 unintended URLs. Analyzing the content and reputation of these webpages allows us to uncover the dangers users face when visiting unintended URLs.

a) Unintended URL Website Content: Figure 9 shows the content of unintended-URL websites as determined through the perceptual hash clustering of screenshots and manual labeling. We find that 43.4% of websites hosted on unintended URLs in our dataset belong to benign entities hosting original content such as small business webpages. A large percentage of these domains are registered under one of the country TLDs including "`.it`," "`.so`" and "`.is`." Given that these are the prominent TLDs of each respective country as well as commonly used words, we attribute this to the over-representation of such content in our dataset.

An additional 39.5% of our dataset is made up of domain parking webpages. Previous work has shown that domain parking services commonly serve malicious content to visitors including malware and technical support scams [50], [15]. Additionally, 171 webpages contain sensitive or malicious content such as deceptive surveys and downloads. Thus, we find 42.3% of the webpages in our dataset could expose users to potentially dangerous and unwanted content.

b) Unintended Domain Maliciousness: In total, 118 target domains and an additional 40 landing domains appear on at least one blacklist reported by VirusTotal [13]. Furthermore, we discovered a large-scale malvertising campaign consisting of 71 domains registered to the same IP address. When visiting any one of these domains, the user is met with an attacker-controlled Traffic Distribution System [30] which, through a series of redirections, lead users to one of many malicious webpages. These webpages contain content including phishing, deceptive surveys, and file downloads. An example of the malicious content served by this campaign can be found in Figure 10.

Our results show that users who follow URLs present in tweets may be subjected to malicious content. The trust users have in the authors of tweets increases the potential damage a malicious webpage can cause, as more users will be confident in the validity of the content.

c) Unintended Domain Registration Date: For the unintended URLs that correspond to registered domain names, a critical question is the following: were these domains already registered at the time of the tweet, or were they registered some time after the URL-including tweet? In the former case, one could straightforwardly argue that these domains are most likely benign and their matching with tweets is coincidental. In the latter case, however, a domain that was registered *after*

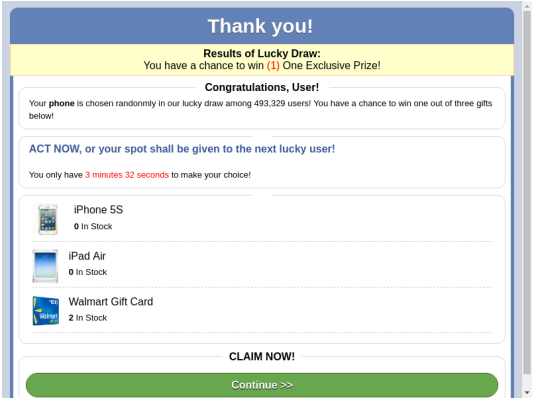

Fig. 10: *Deceptive survey from malicious campaign found among unintended URLs.*

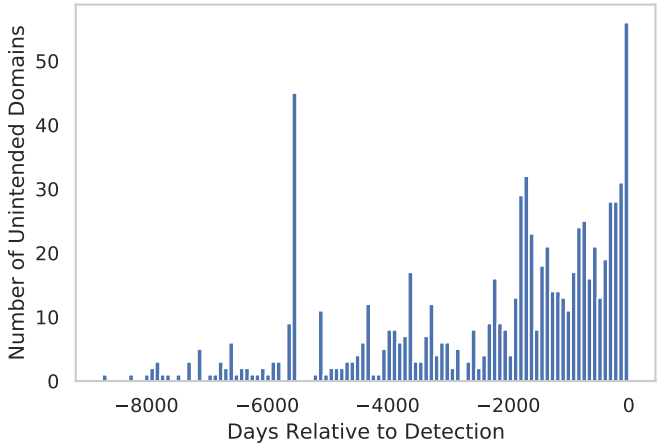

Fig. 11: *Days between the registration of domains and detection by our infrastructure.*

a tweet including a corresponding unintended URL is a clear indication of malice.

Figure 11 shows the registration date of domains in our dataset relative to detection by our infrastructure. We observe a long tail distribution with the majority of registrations occurring within five years of detection. We note a spike in registrations occurring at around 6,000 days before detection. These domains all belong to the .in TLD and were registered after a relaxing of regulations for that TLD in 2005 [9]. As shown in Figure 9, most of the domains in our dataset belong to benign entities. These registrations are therefore most likely independent of tweets recorded by our infrastructure. However, we also observe a spike in domain registrations shortly before detection. Given the detection lag of our infrastructure as well as the fact that some unintended domains are tweeted repeatedly by different users making the same mistake, the spikes at the right-hand side of the graph can be attributed to attackers who observe these tweets and attempt to benefit from the traffic that they generate.

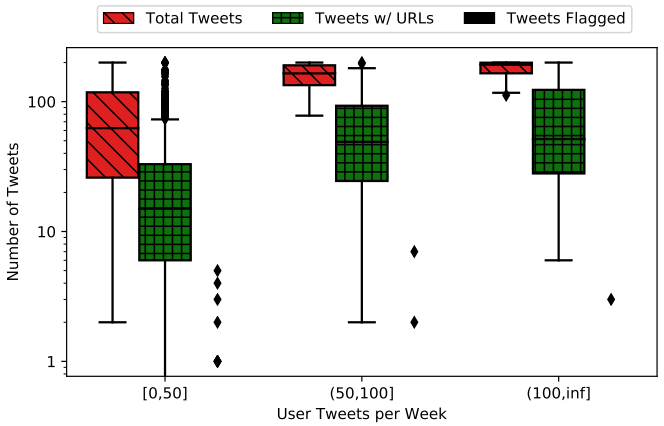

Fig. 12: *Statistics on tweets from users observed publishing at least one tweet containing an unintended URL. Data is grouped by the number of tweets each user publishes weekly.*

V. Developing a Countermeasure Against Unintended URLs on Twitter

In Section III-A we showed that it is possible to train a machine learning classifier to be able to detect unintended URLs on Twitter with high accuracy (94% on our ground truth dataset). In the long run, we expect that Twitter as well as other social media platforms will adopt a classifier similar to the one we proposed and alert users about unintended URLs before posting them on their timelines. We argue that it is important that users are involved in this process, being warned about the potentially unintended URLs that they introduce and being given a chance to correct their tweets. If Twitter (or another social media platform) makes this decision automatically, users whose flagged tweets were false positives, may consider the forced change as a form of censorship.

In the mean time, however, given the potential threat posed by unintended URLs, we want to equip users with tools that they can use to protect themselves and their audience. Since we assume a non-cooperating social network, any solution must be limited to the client side. Given Twitter's software ecosystem, a client-side solution can either be in the form of a special Twitter client (that a user must install and adopt) or in the form of a browser extension (for the users who operate Twitter through its web interface). Due to the non-intrusive nature of browser extensions, their cross-platform operation, and the fact that they can be installed and uninstalled without requiring any other changes to the user's workflow, we opted to deploy our countermeasure through a browser extension.

Specifically, we developed a Chrome browser extension that parses the text of tweets as users type them and runs our unintended URL classifier on that text. In the case of a positive label (i.e., a discovered unintended URL) the extension will alert users so that they can correct their typos (or override our warning) before posting their tweets. In this section, we first describe the functionality of our browser extension and then analyze its performance impact on a user's browser.

A. Browser Extension

Our Chrome browser extension analyzes the text of new tweets as the user is typing them, identifying unintended URLs. When enabled, our extension registers event handles on the "Post Tweet" button in the Twitter Web interface. After the user types the tweet text and clicks the post button, the extension analyzes the tweet text and applies the prefiltering step described in Section III-A. Tweets containing URLs that pass the prefiltering steps are then evaluated using a pre-trained machine learning model that uses most of the features of our aforementioned classifier. Specifically, for our proof-of-concept extension, we excluded the Sentence Segmentation feature that our regular classifier uses. Our analysis of features indicated that this feature had a low importance in our model, in a way that did not justify the time investment that was necessary to port it to the JavaScript language (a step that is necessary for the extension to be self contained).

If the model reports that the tweet contains one or more unintended URLs, the browser extension shows a warning dialog containing all the unintended URLs (positive predictions). This dialog enables the users to review the posted URLs and optionally edit their text before posting. Alternatively, in the case of a negative prediction, the tweet is posted as usual. A video demonstrating our extension along with the source code of our extension can be found in our public GitHub repository: https://github.com/belizkaleli/TypoNoMo. Note that our browser extension only operates locally, and no data about the Twitter user or the text they typed is sent to a third party. As such, our extension is not privacy invasive.

B. Performance Impact

A browser extension should not disrupt the overall user experience if it is to be adopted by users. To understand the number of alerts that Twitter users would experience, we measure the frequency in which the average user (of those who are prone to unintended URLs) would see an alert from our browser extension. We accomplish this by analyzing tweets from each user we observed publishing a tweet containing an unintended URL in SectionIV-C. For each user, we download all authored tweets using the Twitter API (maximum of 200 per user) and record the following: the total number of tweets published, the number of tweets containing at least one URL, and the number of tweets flagged as containing an unintended URL by our extension. We divide our dataset into three groups, based on the number of tweets users in each group publish each week. Figure 12 demonstrates our findings. Overall, out of 93,187 total tweets in our dataset, only 51 would be flagged by our browser extension, with users on average seeing no more than one alert each week. Moreover, we find that medium and high-activity users post a similar number of tweets containing URLs, demonstrating an upper-bound on tweets which could potentially trigger an alert from our extension.

Additionally, we measure the time overhead induced by our extension in the following tweet-posting scenarios:

- The posted tweet does not contain any links.
- The posted tweet contains only one link which does not get filtered in the Prefiltering step (tested for both positive predictions and negative predictions).

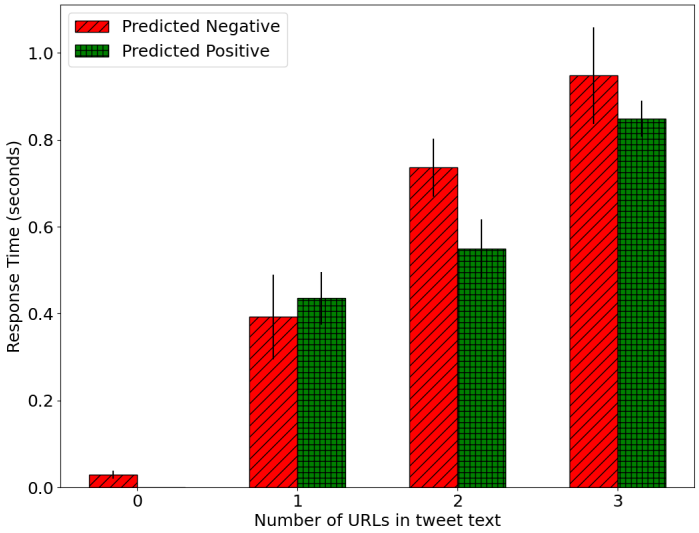

Fig. 13: *Chrome extension response time for different test cases.*

- The posted tweet contains two links which do not get filtered in Prefiltering step (tested for both positive predictions and negative predictions)
- The posted tweet contains three links which do not get filtered in Prefiltering step (tested for both positive predictions and negative predictions)

For the tweets including negative predictions (i.e., the URLs are classified to be intended ones), we record the time from the moment the user clicks the "Tweet" button to the tweet getting posted on the user's timeline. For the tweets including positive predictions (i.e., the URLs are classified to be intended ones), we record the time from the moment user clicks the "Tweet" button to the time they are shown the extension's warning dialog. The posting of tweets is automated using Selenium. Figure 13 shows the results of our performance tests, averaged over ten runs. The results show that while our extension does add a delay to the posting of tweets, this delay is typically under a second, even in the extreme cases of a user posting three URLs (thereby causing three classification tasks) in the same tweet. We can therefore conclude that our extension could protect users and their followers from unintended URLs, for a minimal performance overhead.

VI. Discussion

In this section, we first discuss the implications of our results for online services. We then highlight the limitations of our study and sketch some future work directions.

A. Implications for Online Services

In this paper, we show that the efforts expended by many online services (including Twitter) to identify URLs in the text of their users and render them as clickable links, can produce unintended URLs with negative security consequences. There is an inherent tension between the usability of an online service and its security, leading to problems that platform designers

need to face. Our advice to online services is to consider the threats highlighted in this paper when designing and updating their URL rendering systems. An option would be to develop an unintended URL detection system similar to the one proposed in this paper on their side. This deployment, however, should follow a rigorous risk-benefit analysis weighing the security of users, the usability of the platform, and the user friction introduced by false positive alerts. Given that, according to our results in Section IV, 72% of unintended URLs point to unregistered domain names, we argue that showing a warning whenever users post tweets including an unregistered domain name, would cover the majority of unintended URLs posted by users with virtually no negative side effects.

B. Limitations

Our dataset comes from the 1% streaming API that Twitter provides to vetted researchers. As such, we expect that all the numbers that we presented in this paper are lower bounds of the problem of unintended URLs. Another limitation is that we focus on tweets authored in English for both our model as well as our mitigation. Since most of our features depend on language, building a language-agnostic model is not a straightforward task and therefore we chose not to pursue it in the scope of this work.

C. Future Work

In this paper, we presented a series of promising results towards automatically detecting unintended URLs on Twitter. However, the accuracy of our machine learning algorithm could be further improved. Adding more complicated features and potentially analyzing each tweet in the context of other tweets from the same user, could lead to higher accuracy. At the same time, heavyweight features will also considerably increase the time needed for analysis and therefore increase the performance overhead, particularly if it is to be applied at the client side.

A possible direction for future work is designing a system that preemptively identifies future unintended domains, based on commonly used words and the evolution of TLDs. These domains could then be essentially "cached" by the classifier, leading to classification speedups. This approach would be conceptually similar to the work of Marchal et al. who propose Markov chains built from past phishing websites, to proactively predict future phishing URLs [33]. Orthogonally to predictive classification, a separate direction is to experiment with convolutional neural networks that have exhibited immense accuracy improvements in other fields, compared to traditional machine-learning classifiers. These classifiers tend to require significantly larger ground-truth datasets compared to traditional machine-learning algorithms which made them inapplicable for us. We expect that our proposed classifier (or our simplified heuristic of alerting on tweets including unregistered domain names) could produce such a dataset and therefore enable a transition to more advanced classifiers in the long run.

VII. Related Work

To the best of our knowledge, this paper is the first one to draw attention to the phenomenon of unintended URLs in social media and to characterize the threat that they pose to users.

Due to its popularity, Twitter has attracted large amounts of abuse and commensurate amounts of past research. Spam has always been an issue which has inspired work that quantifies the spam activity on the platform [25], [48], [21], [42] as well as methods to detect fake accounts [47], [43], [19] and differentiate them from compromised accounts [22], [51], both on Twitter as well as other popular social networks [24], [7]. An important differentiator of our work is that the unintended URLs are legitimately posted by benign users, not by spammers controlling fake and compromised accounts. As we showed in this paper, attackers can, after the fact, register the accidentally-introduced URLs and therefore expose the followers of the original Twitter users to arbitrary malicious content.

Orthogonally to spam and account hijacking, researchers have also investigated the security side-effects of allowing users to change their usernames on popular social network platforms [34] as well as whether attackers can confuse users about the nature of posted URLs via web cloaking [40].

The negative consequences of typos to the security and privacy of users have been extensively studied in the area of domain squatting. Typosquatting specifically, refers to attackers registering mistyped domain names (such as `twitte.com`) in an attempt to capture all the traffic from users who mistype a website's URL in their browsers. Past research has characterized the typosquatting abuse in the wild [14], [17], [28], [35], [52], [45] as well as the effects of typos in related areas, such as, website development [36], package managers [3], and mobile app stores [27]. In this paper, we discovered that while typos are one of the reasons for unintended URLs (where a Twitter user intends to tweet one URL but tweets another), the main culprit of unintended URLs is the semantic gap between what a user types (such as a sentence with a missing space or an Instagram ID) and what Twitter infers that that user typed (i.e., a URL).

One of the reasons why Twitter and other social network platforms are so eager to find URLs in user tweets, is the constant expansion of valid domain TLDs. Next to traditional generic TLDs (such as "`.com`" and "`.org`") and country-code TLDs (such as "`.it`" and "`.es`"), ICANN has, since 2013, approved more than 1,200 new gTLDs, such as, "`.life`," "`.love`" and "`.beer`" [6]. These new gTLDs combined with user typos are making it more likely that a social network platform will identify URLs when users never intended them. For at least some of these TLDs, researchers have questioned whether they fulfill a real user need or are just creating more opportunities for domain squatting and trademark abuse [26], [37].

VIII. Conclusion

In this paper, we showed that the automatic link-rendering feature of popular social media, combined with incorrect spelling and punctuation, can result in unintentional URLs. We presented a threat model on social media platforms in which an adversary abuses this phenomenon to launch attacks on users who click on those unintentional links. We evaluated the link-rendering behavior of several online platforms to show

the extent of the problem and focused on the manifestation of unintended URLs on Twitter.

Given the volume of Twitter data, we proposed features that can be used in the context of supervised machine learning to identify unintended URLs in user tweets and used our classifier over a period of 7 months, processing millions of tweets and discovering a total of 26,596 unintended URLs. We analyzed the properties of these unintended URLs and characterized the abuse that attackers could inflict by registering 45 domains found in unintended URLs. There, we discovered that, as long as attackers register unintended domains shortly after they are posted on Twitter, they will receive visits from hundreds of unsuspecting Twitter users who are merely following the links posted by trusted user accounts. Lastly, we presented a lightweight browser extension which will warn users when they are about to tweet text that includes an unintended URL.

Our study sheds light on the previously unexplored issue of unintended URLs which we hope will be used by online platforms to re-evaluate their link-rendering algorithms and consider warning users when unintended URLs are about to be posted. At the same time, our work highlights the importance of being careful when authoring messages on social media, where the absence of a space can now be weaponized to expose millions of users to malicious content.

IX. Availability

The code for our proposed browser extension can be found at the following URL: `https://github.com/belizkaleli/TypoNoMo`.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback. For Boston University, this work was supported by the National Science Foundation under grant CNS-1942610 and by a seed grant from the Center for Information & Systems Engineering and the College of Engineering at BU. For Stony Brook University, this work was supported by the National Science Foundation under grants CNS-1941617, CNS-1813974, and CMMI-1842020 as well as by the Office of Naval Research under grant N00014-20-1-2720.

References

[1] https://selenium.dev.

[2] "About twitter cards - twitter developers," https://developer.twitter.com/en/docs/tweets/optimize-with-cards/overview/abouts-cards.

[3] "Attackers use typo-squatting to steal npm credentials," https://threatpost.com/attackers-use-typo-squatting-to-steal-npm-credentials/127235/.

[4] "Consuming streaming data," https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.

[5] "deepsegment," https://pypi.org/project/deepsegment/.

[6] "Delegated Strings — ICANN New gTLDs," https://newgtlds.icann.org/en/program-status/delegated-strings.

[7] "Detecting malicious content on facebook," https://arxiv.org/abs/1501.00802.

[8] "Hackers tell the story of the twitter attack from the inside," https://www.nytimes.com/2020/07/17/technology/twitter-hackers-interview.html.

[9] ".in tld regulations," https://web.archive.org/web/20080119113830/http://www.registry.in/policies/.

[10] "K-nearest neighbor," http://scholarpedia.org/article/K-nearest_neighbor.

[11] "Rudy Giuliani accuses Twitter of bias for hyperlinking text," https://www.theverge.com/2018/12/5/18127063/rudy-giuliani-twitter-bias-accusation-hyperlinked-text-president-trump.

[12] "typo - dictionary definition," https://www.vocabulary.com/dictionary/typo.

[13] "Virustotal," https://virustotal.com.

[14] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis, "Seven months' worth of mistakes: A longitudinal study of typosquatting abuse," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2015.

[15] S. Alrwais, K. Yuan, E. Alowaisheq, Z. Li, and X. Wang, "Understanding the dark side of domain parking," in *Proceedings of USENIX Security Symposium*, San Diego, CA, August 2014.

[16] A. Badawy, E. Ferrara, and K. Lerman, "Analyzing the digital traces of political manipulation: The 2016 russian interference twitter campaign," in *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Barcelona, Spain, August 2018.

[17] A. Banerjee, D. Barman, M. Faloutsos, and L. N. Bhuyan, "Cyber-fraud is one typo away," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Phoenix, AZ, April 2008.

[18] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *Proceedings of the International Conference on Email and Anti-Spam (CEAS)*, Redmond, WA, July 2010.

[19] Y. Boshmaf, D. Logothetis, G. Siganos, J. Lería, J. Lorenzo, M. Ripeanu, and K. Beznosov, "Integro: Leveraging victim prediction for robust fake account detection in osns." in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2015.

[20] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali, "Mean birds: Detecting aggression and bullying on twitter," in *Proceedings of the Conference on Web Science (WebSci)*, New York, NY, June 2017.

[21] C. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou, "6 million spam tweets: A large ground truth for timely twitter spam detection," in *Proceedings of the International Conference on Communications (ICC)*, London, UK, June 2015.

[22] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "Compa: Detecting compromised accounts on social networks." in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2013.

[23] G. Eysenbach, "Can tweets predict citations? metrics of social impact based on twitter and correlation with traditional metrics of scientific impact," *J Med Internet Res*, 2011.

[24] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," in *Proceedings of SIGCOMM conference on Internet Measurement Conference (IMC)*, Melbourne, Australia, November 2010.

[25] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: The underground on 140 characters or less," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Chicago, IL, October 2010.

[26] T. Halvorson, K. Levchenko, S. Savage, and G. M. Voelker, "Xxxtortion? inferring registration intent in the. xxx tld," in *Proceedings of the International Conference on World Wide Web (WWW)*, Seoul, Korea, April 2014.

[27] Y. Hu, H. Wang, R. He, L. Li, G. Tyson, I. Castro, Y. Guo, L. Wu, and G. Xu, "Mobile app squatting," in *Proceedings of the Conference on Web Science (WebSci)*, Taipei, Taiwan, April 2020.

[28] M. T. Khan, X. Huo, Z. Li, and C. Kanich, "Every second counts: Quantifying the negative externalities of cybercrime via typosquatting," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, San Jose, CA, May 2015.

[29] S. Kruikemeier, "How political candidates use twitter and the impact on votes," *Computers in Human Behavior*, 2014.

[30] Z. Li, S. Alrwais, Y. Xie, F. Yu, and X. Wang, "Finding the linchpins of the dark web: A study on topologically dedicated hosts on malicious web infrastructures," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, May 2013.

[31] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News*, 2002.

[32] C. Ling, U. Balcı, J. Blackburn, and G. Stringhini, "A first look at zoombombing," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Virtual, May 2021.

[33] S. Marchal, J. François, and T. Engel, "Proactive discovery of phishing related domain names," in *Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, Amsterdam, Netherlands, September 2012.

[34] E. Mariconti, J. Onaolapo, S. S. Ahmad, N. Nikiforou, M. Egele, N. Nikiforakis, and G. Stringhini, "What's in a Name? Understanding Profile Name Reuse on Twitter," in *Proceedings of the International Conference on World Wide Web (WWW)*, Perth, Australia, April 2017.

[35] T. Moore and B. Edelman, "Measuring the perpetrators and funders of typosquatting," in *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*, Tenerife, Canary Islands, January 2010.

[36] N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. Van Acker, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "You are what you include: Large-scale evaluation of remote javascript inclusions," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Raleigh, NC, October 2012.

[37] S. Pouryousef, M. D. Dar, S. Ahmad, P. Gill, and R. Nithyanand, "Extortion or Expansion? An Investigation into the Costs and Consequences of ICANN's gTLD Experiments," in *Proceedings of the Passive and Active Measurement Conference (PAMC)*, Virtual, March 2020.

[38] H. Sanchez and S. Kumar, "Twitter bullying detection," *USENIX Symposium on Network Systems Design and Implementation*, 2011.

[39] K. Starbird, A. Arif, and T. Wilson, "Disinformation as collaborative work: Surfacing the participatory nature of strategic information operations," in *Proceedings of ACM on Human-Computer Interaction (PACM HCI)*, Taipei, Taiwan, October 2019.

[40] G. Stivala and G. Pellegrino, "Deceptive previews: A study of the link preview trustworthiness in social platforms," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2020.

[41] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Austin, TX, December 2010.

[42] G. Stringhini, P. Mourlanne, G. Jacob, M. Egele, C. Kruegel, and G. Vigna, "EVILCOHORT: Detecting communities of malicious accounts on online services," in *Proceedings of USENIX Security Symposium*, Washington, DC, August 2015.

[43] G. Stringhini, G. Wang, M. Egele, C. Kruegel, G. Vigna, H. Zheng, and B. Y. Zhao, "Follow the green: Growth and dynamics in twitter follower markets," in *Proceedings of SIGCOMM conference on Internet Measurement Conference (IMC)*, Barcelona, Spain, October 2013.

[44] P. H. Swain and H. Hauska, "The decision tree classifier: Design and potential," *IEEE Transactions on Geoscience Electronics*, 1977.

[45] R. Tahir, A. Raza, F. Ahmad, J. Kazi, F. Zaffar, C. Kanich, and M. Caesar, "It's all in the name: Why some urls are more vulnerable to typosquatting," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Honolulu, HI, April 2018.

[46] K. Thomas, D. Akhawe, M. Bailey, D. Boneh, E. Bursztein, S. Consolvo, N. Dell, Z. Durumeric, P. G. Kelley, D. Kumar *et al.*, "Sok: Hate, harassment, and the changing landscape of online abuse," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Virtual, May 2021.

[47] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Berkeley, CA, May 2011.

[48] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect," in *Proceedings of SIGCOMM conference on Internet Measurement Conference (IMC)*, Berlin, Germany, November 2011.

[49] S. V. M. Vishwanathan and M. Narasimha Murty, "Ssvm: A simple svm algorithm," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Honolulu, HI, May 2002.

[50] T. Vissers, W. Joosen, and N. Nikiforakis, "Parking sensors: Analyzing and detecting parked domains," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2015.

[51] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Towards detecting anomalous user behavior in online social networks," in *Proceedings of USENIX Security Symposium*, San Diego, CA, August 2014.

[52] Y.-M. Wang, D. Beck, J. Wang, C. Verbowski, and B. Daniels, "Strider typo-patrol: Discovery and analysis of systematic typo-squatting," in *Proceedings of the Conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, San Jose, CA, July 2006.

[53] S. Zannettou, T. Caulfield, W. Setzer, M. Sirivianos, G. Stringhini, and J. Blackburn, "Who let the trolls out? towards understanding state-sponsored trolls," in *Proceedings of the Conference on Web Science (WebSci)*, Amsterdam, Netherlands, May 2019.

[54] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," 2010.