

What Remains Uncaught?: Characterizing Sparsely Detected Malicious URLs on Twitter

Sayak Saha Roy

The University of Texas at Arlington
sayak.saharoy@mavs.uta.edu

Unique Karanjit

The University of Texas at Arlington
unique.karanjit@mavs.uta.edu

Shirin Nilizadeh

The University of Texas at Arlington
shirin.nilizadeh@uta.edu

Abstract—Twitter maintains a blackbox approach for detecting malicious URLs shared on its platform. In this study, we evaluate the efficiency of their detection mechanism against newer phishing and drive-by download threats posted on the website over three different time periods of the year. Our findings indicate that several threats remained undetected by Twitter, with the majority of them originating from nine different free website hosting services. These URLs targeted 19 popular organizations and also distributed malicious files from 9 different threat categories. Moreover, the malicious websites hosted under these services were also less likely to get detected by URL scanning tools than other similar threats hosted elsewhere, and were accessible on their respective domains for a much longer duration. We believe that the aforementioned features, combined with the ease of access (drag and drop website creating interface, up-to-date SSL certification, reputed domain, etc.) provides attackers a fast and convenient way to create malicious attacks using these services. On the other hand, we also observed that the majority of the URLs which were actually detected by Twitter remained active on the platform throughout our study, allowing them to be easily distributed across the platform. Also, several benign websites in our dataset were detected by Twitter as being malicious. We hypothesize that this is caused due to a blocklisting procedure used by Twitter, which detects all URLs originating from certain domains, irrespective of their content. Thus, our results identify a family of potent threats, which are distributed freely on Twitter, and are also not detected by the majority of URL scanning tools, or even the services which host them, thus making the need for a more thorough URL blocking approach from Twitter's end more apparent.

I. INTRODUCTION

Over the last decade, social networking websites, such as Twitter, Facebook and Instagram have seen a steady increase with respect to the presence of malicious URLs that are posted on these platforms, especially phishing websites [1]. Attackers find these platforms alluring because of the easy propagation of information through users' networks which also increases the exposure of these URLs to users as well [2]. The detection and prevention solutions against these types of attacks are provided by social media platforms themselves, as well as third-party URL scanning tools. Twitter, in particular, takes

a blackbox approach towards detecting malicious URLs [3]. There is a huge body of work studying the problem of malicious web attacks on Twitter. However, the majority of them attempt to measure the prevalence of malware and phishing attacks spreading through the user networks [4], [5], [6], [7], [8]. Some studies have also replicated phishing attacks [9], [10] to study their characteristics and how Twitter detects them. In contrast, our work focuses mostly on determining and characterizing malicious URLs which are not detected on Twitter for an extended period of time, as well as benign URLs which are incorrectly detected due to several factors.

In this work, we aim to find the effectiveness of Twitter's URL detection mechanism against several Phishing and Drive-by URL threats found on the platform through three different time periods of the year, and in turn, determine the features of the URLs which do not get detected by Twitter. In particular, this paper (i) evaluates the URLs that are posted on Twitter and identifies their response towards both Phishing and Drive-by URLs over the period of the study; (ii) categorizes the URLs that are not detected by Twitter based on their characteristics and also identifies certain features that can be used to recognize them in the wild; and (iii) examines the characteristics of *benign* URLs that are detected as malicious by Twitter.

We collected and analyzed more than 132k unique URLs, which had been collected from over 3 million posts, spread out over three different time periods of the year (January, June and November 2020). Using VirusTotal, an online tool to scan resources with multiple scanning engines at the same time and used by both regular users and the research community alike [11], [12], [13], [14], [15], [16], we initially labelled 4.5k URLs as 'malicious' based on the scores provided by VirusTotal, and then manually investigated each of them to check if they were indeed so. The URLs which were then confirmed to be malicious were analyzed to determine their potency, as well as patterns and obfuscation tactic used by the attackers to make these URLs remain undetected by both Twitter and popular anti-phishing engines over extended periods of time. Finally, we also determined the characteristics for the benign URLs in our dataset that were detected by Twitter as malicious.

Our findings can be summarized as below:

- 1) Twitter was unable to detect about 66% of the malicious

URLs in our dataset over the course of our study, most of which utilized one of nine different website creation services. These malicious URLs also targeted 19 popular organizations (phishing websites) and distributed malicious files from 9 different threat categories (drive-by downloads).

- 2) Twitter exhibited inconsistent protection behaviour against phishing websites in our dataset that it detected, only generating a warning for the majority of them, instead of removing them outright. Over the course of our study which spanned nearly 11 months, we found that a substantial amount of these URLs were still alive on the platform and could be freely redistributed.
- 3) We also found that more than 60% of these URLs were still accessible on their respective hosting domains even after a month of their first appearance in our dataset. They also got detected by very few URL scanning engines, with none of these tools being able to flag even half of the phishing URLs in our dataset as malicious (that had not been detected by Twitter throughout our study).
- 4) Several URLs in our dataset which exhibited no malicious behaviour were detected by Twitter. Our results indicate that the majority of these incorrect warnings were due to Twitter blocklisting websites which utilized several URL shortening and web hosting domains.

II. BACKGROUND AND RELATED WORK

Phishing Websites. Phishing websites are websites which attempt to imitate legitimate organizations in order to trick users into sharing their personal or financial information. The scale of damage done by these attacks is massive, with more than 1.4 million such websites being created every month [17] which leads to losses of half a million dollars to American businesses every year [18]. Established academic literature is plentiful in this regard, with researchers tracking several techniques used by these attackers [19], [20], [21], [22], [23], [24], determining how prevalent phishing detection measures fare against these attacks [25], [26], [27], [28], [29], as well as a myriad of obfuscation strategies used by the attackers to prevent their attacks from being detected by these tools. Significant progress has also been made towards detecting phishing websites, by bringing improvements to URL blocklists [30], [31], [32], [33], [34], [35], as well as developing novel machine learning based approaches [36], [37], [38], [39], [40], [41], [42], [43] which utilize several features derived from the characteristics of these websites. Considering the resulting improvements that have been made in detecting phishing attacks over the last few years, our work focuses mainly on a single, yet popular platform for spreading phishing and other malicious URLs alike - Twitter - and we aim to evaluate Twitter's performance against several URL threats posted on its platform.

VirusTotal. VirusTotal [44] is an online URL scanning tool which scans both files and URLs for malicious content. After a user submits a file (or URL), VirusTotal checks with 80 different URL scanning engines and returns the aggregated

total and name of the tools which detected the resource as a threat. In addition to an online web interface [44], VirusTotal also provides an API [45] for scanning larger volume of URLs. Researchers frequently use VirusTotal to label URLs in their dataset, with several published literature depending on it to create their ground truth for malicious URLs [46], [12], [47], [11], [48], [49]. We use VirusTotal in a similar way to determine what portion of scanners detect the URLs in our dataset in Section III-A, and later on in Sections IV-A and V-A to compare the detection statistics between phishing and drive-by URLs which were and were not detected on Twitter.

III. DATA COLLECTION AND PROCESSING

We collected one million tweets from each of three time periods which consist of January (Dataset 1 or D1), June (Dataset 2 or D2) and November 2020 (Dataset 3 or D3) using the Twitter API [50]. Our unconventional approach towards spreading out the data collection over the entire year is to a) determine if our findings are consistent over a large period of time, and conversely, b) to check if Twitter improves on the issues raised in subsequent time periods. We only collected tweets which had URLs embedded in them. Since all URLs posted on Twitter are enclosed under the t.co URL shortening banner, we resolved each of them such that we were able to collect the next URL in the redirection chain. We then removed the URLs that were links to other tweets or images/videos posted on Twitter. This is to make sure we only preserve the tweets in our dataset, which contain URLs which link to third-party websites.

After this filtering procedure, we were able to retain 43,605 URLs for D1, 51,129 URLs for D2 and 37,951 URLs for D3, for a total of 138k URLs across the three datasets. Note that some of these URLs are duplicated in these three datasets, and later we only consider them in the dataset they first appeared in.

A. Identifying Malicious URLs

For the purposes of our study, we first had to reliably identify the URLs in our dataset which might be malicious. To do this, we at first use VirusTotal [44] to scan all the unique URLs in our datasets. We initially consider URLs which are detected by at least one engine on Virustotal as being 'malicious'.

Prior work in [51] has found that it is common for URL scanning engines to change their detection from malicious to benign within a short period of time. Also the label provided by VirusTotal for a particular engine also lags behind the engine's actual label in several cases. Since a majority of our URLs have only one detection, it is possible for these detections to be removed for a short time, (thus resulting in the URL being classified as benign in our dataset, as per our threshold) only for a detection to be added back to the URL in a few hours. Thus, to avoid this inconsistency, the URLs in our datasets that had at least one detection by VirusTotal on the first day were scanned regularly for a month. If a URL turned out to have zero detections in newer scans, it was separated

from the *malicious* URLs set and was again scanned after a week. In the event that it gained detections in the weekly scan, it was put back in the *malicious* URLs set again. On the other hand, URLs which did not get detected in the first scan were considered as benign initially, then scanned every week, and if they gained at least one detection during these weekly scans, were then moved to the *malicious* set, and scanned daily from thereon. After the conclusion of a month for the respective time frames of our dataset, we obtained 1,318 URLs from D1 (from 1,421 tweets), 1,827 URLs (from 2,082 tweets) from D2 and 1,401 URLs from D3 (from 1,650 tweets), which were still detected by one or more engines on VirusTotal, with all of them comprising about 3% of the total URLs collected from their respective datasets.

To identify whether the URLs are indeed malicious, and to further categorize them, we investigated them manually as described in the next section.

B. Categorization of 'Malicious' URLs

The 4,546 URLs across our datasets which had one or more detection on VirusTotal were evaluated manually by two security researchers, who labelled them independently. The researchers were advised about the risks associated with visiting the malicious URLs and took necessary precautions to prevent getting their systems infected. These precautions included using a virtual machine to visit the websites and using dummy account credentials for websites which required login information to be properly evaluated. Since malicious websites (especially those that are phishing) generally have a very short lifespan [52], all the URLs included in our study were evaluated on the day of their collection itself. We provided certain guidelines for the researchers, such as features to observe inside the website which helped them label each URL as one of the four pre-defined categories: Phishing, Drive-by Download, Benign and Unknown, which are discussed below.

1) Phishing: These websites imitate legitimate organizations and ask the victim for sensitive and private information, such as account passwords, social security numbers, payment information, etc. The majority of phishing websites are usually hosted on independent domains, lack SSL certification (though this phenomenon is changing slowly [53]) and/or are part of a larger phishing campaign.

2) Drive-by Download: Websites of this category usually claim to offer a legitimate software, which is not easily available otherwise, enticing the user to download files (usually executable programs), which come bundled with trojan-horses, ransomwares, spyware or potentially unwanted programs (such as adwares) [54]. However, a drive-by download attack might also occur without the user's knowledge, by often exploiting vulnerabilities in the victim's browser or operating system [55]. Unlike phishing URLs which are usually detected by frequently updated blocklists, file based malware detection is much more complex, and depends on several factors beyond simple signatures, such as file heuristics and other sophisticated behaviour based detection procedures, something which VirusTotal does not consistently emulate

in their scans [56][57][58][48]. Also several malicious files bundled in seemingly benign software do not create executables (such as malicious browser extensions, fileless malware, miscellaneous PuP, etc.) and thus cannot be easily uploaded and scanned on VirusTotal. Thus, for websites which asked to download a file, coders were prompted to scan the downloaded file using the desktop versions of four top rated antivirus engines (Bitdefender, Kaspersky, Avast and AVG) according to AV-Comparatives [59] and if at least two of these engines detect it as a threat, they labelled the URL as a *Drive-by Download* website. The threshold of detection for identifying the downloaded file as malicious is set to at least two engines to reduce the possibility of a false positive. For the files which did not pick up any detections or had only one detection, we keep re-scanning them throughout our study to check if detections went up. The real-time protection modules of these products were also able to detect malicious web-browser plugins and file-less malware, and other software exploits.

3) Benign websites: Websites in our datasets, which i) did not imitate any organization, ii) did not ask for sensitive data when it was unwarranted, and iii) did not ask to download a file, and if it did, it turned out clean in the four antivirus engine scans, were labelled as benign URLs.

4) Unknown websites: Websites which could not be reliably determined if they had any malicious intent. This included the URLs which could not be accessed, and several websites which appeared to be e-commerce portals, or even those which redirected the coders to online advertisements or third party websites. The e-commerce websites did not imitate any other legitimate organization but asked for sensitive data, such as credit card information, residential address, etc. with the promise of delivering some product/service at a later date. Thus, it is hard to evaluate whether these websites have any malicious intent. Findings in [60] shows that several popular and legitimate websites and domains become the source of unknowingly promoting malicious advertisements due to CDNs as well. Thus, the coders were asked to label websites which provided suspicious advertisements as *Unknown*. We do not include websites in this category for any further analysis in our study.

4) Label agreement: After labeling the URLs by two coders, we computed their inter-rater agreement, which is 96% with a Cohen's kappa of 0.72, indicating substantial agreement. For inconsistent results, coders discussed how to resolve disagreements and assigned a final label to the URLs. Table I provides a descriptive statistic of the URLs for each category.

IV. PHISHING URLs ON TWITTER

Based on our manual evaluation, we found 950 URLs across our datasets which exhibited phishing behaviour. Table III illustrates the statistics of the URLs which were detected by Twitter over the course of a month. By '*detected*', we refer to one of two situations. The first one is that the tweet was no longer available on Twitter, suggesting it might have removed the tweet(s) containing the URL, but, there remains

Dataset	Time Period	Unique URLs	detection rate ≥ 1	Phishing	Drive-by-download	Benign	Unknown
1	Jan 2020	43,605	1,318	186 (14%)	67 (5%)	879 (66%)	139 (11%)
2	June 2020	51,129	1,827	471 (25%)	109 (6%)	1041 (56%)	206 (11%)
3	Nov 2020	37,951	1,401	293 (21%)	91 (6%)	896 (63%)	121 (9%)

Table I: Distribution of the categories of the URLs across the three datasets.

the possibility that the author of the tweet removed it as well. To clarify this situation, for every URL that we could not access, we retried posting the same URL using one of our private accounts(which for ethical purposes, could not be accessed by any other user on Twitter), and only considered as 'detected' if Twitter prevented us from posting it or provided an explicit warning. The second scenario is when the URL was still accessible on Twitter, but it provided a warning indicating that the URL might be harmful to visit, as seen in Fig 11. About 62% of these URLs were still active, on average, across the three datasets after the month of their first appearance in our dataset, with around 57% of them still being accessible on Twitter even after the conclusion of the study. Thus, a total of 544 phishing URLs remaining undetected on Twitter which warrants a closer look into the characteristics of these threats.

A. Comparing Detected and Undetected Phishing URLs

To check how other URL scanning tools fared against these undetected URLs ($n=544$), we compared their detection statistics with the 406 URLs which had been detected by Twitter during our study. We did this by scanning each of these URLs using VirusTotal to determine how many tools detected them. The descriptive statistics for this analysis is illustrated in Table II. We find that the undetected phishing URLs are detected on average by 2.95 engines ($\sigma=0.94$), whereas those which had been detected by Twitter at some point during the duration of the study maintained an average detection rate of 9.66 ($\sigma=4.21$). The undetected URLs were also caught by only 17 unique engines on VirusTotal, compared to 38 engines for the detected URLs. Figure 1(a) shows the cumulative distribution of number of times each engine detected these URLs throughout the duration of our study. We observe that nearly 80% anti-phishing tools were unable to detect even 10% of all the undetected URLs, with none of the engines being able to detect half of them. This is in stark contrast with these tools' performance against the phishing URLs which were detected by Twitter, for which 10 engines were able to detect half of the URLs, and three of them even managed to detect 80% of them. Thus, this difference in detection patterns indicates the undetected URLs (on Twitter) might have different characteristics than those which were detected (on Twitter) and so we dedicate Section IV towards closely investigating the them.

B. Analyzing the Undetected Phishing URLs

We found 544 phishing URLs which were not detected by Twitter throughout the course of our study. Also, after scanning these URLs using VirusTotal, we found that they had significantly lower detection rates by URL scanning tools compared to URLs which had been detected by Twitter.

URL Set	Total	AD	Median	Min/Max	Std. Dev	Engines
Undetected Set	544	2.95	3	0/12	0.94	17
Detected Set	406	9.66	10	1/21	4.21	38

Table II: URL Scanning Tool detection statistics for Detected and Undetected(on Twitter) Phishing URLs in our datasets.

AD=Average engine detections, Engines=Total number of unique engines.

Dataset	Total	Detected after a month	Detected throughout the study
D1	186	79 (42%)	85 (46%) ¹
D2	471	182 (38%)	209 (44%) ²
D3	293	107 (36%)	112 (38%) ³

Table III: Descriptive stats for Phishing URLs detected across the three datasets. **Time periods 1=Jan to Dec 2020, 2=June to Dec 2020, and 3=Nov to Dec 2020**

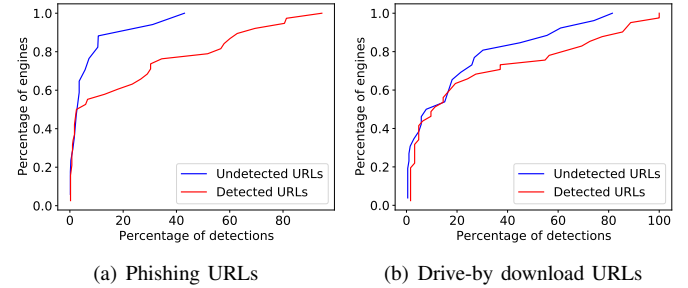


Figure 1: CDF of the percentage of detected and undetected (a) phishing and (b) drive-by download URLs that were flagged by the anti-phishing engines.

Through our manual investigation, we found that 435 (80%) of these URLs are hosted by one of six different free website creation services: Google Forms, Microsoft Forms, Wordpress, Microsoft Sharepoint, Wix, Weebly and one API interface: Google APIs. Websites hosted under these services always have SSL certification, the lack of which is often considered by regular users as a sign that a website might be malicious [53]. Table IV shows the distribution of the URLs in our dataset across these website hosting services. Regular phishing URLs are often removed by the domain hosting services or registrars themselves very quickly upon their appearance [61]. However, we found that only a small number of URLs hosted under these domains had been removed on the day after their appearance on Twitter, with 87% of the phishing attacks still active across all the different hosting services combined. This number drops to only about 57% even after a month of their appearance in our dataset. It is also observed that these attacks are predominantly present across all three of our datasets, signifying that phishing attacks having these features were

carried out throughout the entire year. Moreover, these attacks primarily targeted 19 popular organizations including AT&T, eBay, Amazon, Facebook, etc. as illustrated in Figure 2. We thus dedicate the next few paragraphs towards discussing the characteristics of the phishing websites hosted under each of these website creation services and also discuss possible features that can be observed to identify them in the wild.

Category	D1	D2	D3	A_{d1}	A_M
Google Forms	43	88	54	163 (88%)	139 (75%)
Microsoft Forms	16	41	23	73 (91%)	39 (49%)
Wordpress	12	9	20	31 (74%)	23 (55%)
Wix	6	15	21	35 (83%)	13 (31%)
Google APIs	10	29	18	52 (94%)	12 (21%)
Weebly	2	17	11	26 (86%)	21 (70%)
Total	89	199	147	380 (87%)	253 (57%)
	435 (100%)				

Table IV: Distribution of undetected phishing URLs across the 3 datasets. A_{d1} = phishing URLs still active, i.e., they can be visited on day one, and A_M = phishing URLs active after a month of their first appearance in our datasets.

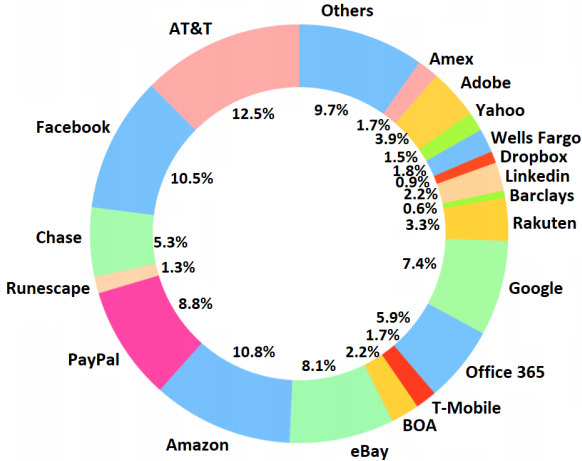


Figure 2: Organizations targeted by the Phishing URLs ($n=544$) which remained undetected on Twitter.

C. Google and Microsoft Forms URLs

From our datasets, we found 34% of the URLs were hosted on Google Forms and 15% of the URLs hosted on Microsoft Forms. These services allow for easy point to click interfaces for creating web-forms, where users can submit data by entering information in the provided form fields. However, we found that attackers frequently utilized these services to create websites which imitated legitimate organizations and asked for sensitive information, such as account passwords, credit card information, SSN information, etc. Form URLs created under both these services are hosted under Google's (<https://docs.google.com/forms/>) and Microsoft's (<https://forms.office.com/>) own domains. In fact, unlike domains which host regular phishing websites, we find that both these services are not very efficient at detecting

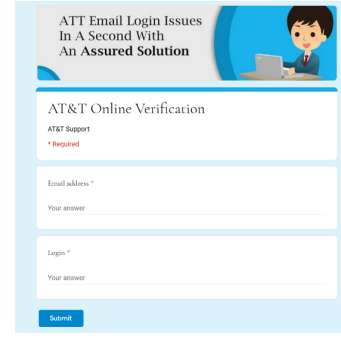



Figure 3: A simple phishing websites hosted on Google Forms which uses text for its form fields.

these phishing forms, with around 75% and 49% of them still active on Google Forms and Microsoft respectively after the first month of their appearance. However, websites created using these services are very limited with respect to design and only allow users to change the background colour and add images/videos in pre-defined field areas. This lack of customization separates these two website services from the others which hosted undetected phishing URLs in our dataset. However, we found that attackers were able to use creative obfuscation measures to construct phishing URLs which could partially imitate the organizations they were targeting and also remain undetected for extended periods. We discuss these strategies below.

1) *Obfuscation approaches used by attackers:* A large portion of these phishing forms hosted on both the services used images for creating the form fields instead of regular text as shown in Figures 4(a) and 4(b). We found that all 17 of the Google Form URLs that were removed after one day consisted of text only form fields. In contrast, out of the 139 URLs that remained alive, 111 of them used images for form-fields, out of which 21 were removed after a month. Twenty-eight URLs used special characters in between the malicious form text (e.g., PA***WORD* instead of PASSWORD) as shown in Figure 4(c). Only 10 of these URLs were removed after a month. Similarly, all 73 of the Microsoft Form URLs, which were active after a day, consisted of images as form fields, though we found no URLs using special characters inside form fields. Microsoft fared a bit better than Google, removing 45% of these URLs with image based fields after a month of their appearance in our dataset. Thus, our findings indicate attackers often use images and special characters for obfuscating these phishing URLs, which ensures that these attacks remain online longer.

D. Wix and Weebly URLs




Eight percent of the URLs in our dataset were hosted on Wix, while 30 URLs were hosted on Weebly, two free hosting websites, and were not detected by Twitter throughout the course of our study. Unlike Google and Microsoft Forms, however, these services provide far more options for customization including layout themes and sophisticated web



YouTube Premium

With over 1000 hours of premium content, and no ADs, ever!

Enter your Credit Card number in the field below.




Note: You will not be charged before the end of your 30-day trial period.

Your answer

Submit

AT&T ACCOUNT UPGRADE



AT&T

1

Login-id

Enter your answer

2

Password


Enter your answer

YAHOO!

Figure 4: (a) and (b) Google and Microsoft Forms phishing with images as form fields (highlighted in red), (c) Google Forms phishing with special characters for form fields (highlighted in blue). These websites are removed at a much slower pace than similar websites with text-based form fields.

Welcome to Office

Microsoft Office 365 lets you work better, collaborate, and get going
and more.




Sign In







Email Address


Enter Password

LOG IN



Office 365





Microsoft


POWERED BY  bing

Figure 5: (a) Phishing website hosted on Wix, (b) Phishing website hosted on Weebly.

designing objects, all bundled in easy drag and drop interfaces. Attackers thus leveraged these features to create phishing websites which were much harder to recognize based on appearance than the form URLs we have discussed in IV-C, as is apparent in the illustrations shown in Figure 5(a) and (b). We found that 70% and 31% of the URLs hosted on Weebly and Wix respectively were alive even after a month of their appearance in our datasets.

E. Wordpress URLs

We found around 7.5% of the phishing URLs in our datasets, which were hosted on Wordpress’ own free web-hosting domain (wordpress.com). Users of wordpress.com have access to several customizability options and feature rich plugins, which resulted in attackers creating phishing websites which were practically indistinguishable in appearance from organizations that they were targeting, with little to no indication of them being a free website hosted on wordpress.com. Figure 6 shows one such website, which very accurately imitates the JP Morgan Chase login page. In addition, similar to the Wix and Weebly phishing URLs, these websites also included navigation bars containing links which lead to legitimate resources on the targeted organization’s website.

A screenshot of the Chase mobile app login screen. The background is a scenic image of a person walking on a wooden boardwalk over a body of water, with a forest in the background. The Chase logo is at the top center. Below it, there are input fields for 'Username' and 'Password'. There are two checkboxes: 'Remember me' and 'Use token'. A blue button labeled 'Sign In' is below the password field. Below the button, there is a link 'Forgot username/password?' and a link 'Not Enrolled? Sign Up Now.' At the bottom, there is a row of social media icons (Facebook, Instagram, Twitter, YouTube, LinkedIn) and the text 'Follow us:'. At the very bottom, there is a navigation bar with links: 'Security', 'Terms of Use', 'Accessibility', 'S&P Act, Chase Business Terms, Disclosures', 'Fee Limits', 'About Chase', 'U.S. Maps', 'Where Chase & Co.', 'Careers', and 'Feedback'.

F. Google APIs

Google’s Cloud Storage API is an online repository offered by Google to host web applications created using Firebase. We found 10% of the URLs in our dataset which utilized this service to host phishing web applications. Prior work in [62] has noted that scammers utilize this service to implement web applications which imitate popular organizations. We also found that 36% of the URLs target some form of Microsoft Outlook or 365 suite of login page. We found that Google was able to remove only 6% of these applications after a day, but went on to get rid of 79% of these URLs over the course of a month.

G. Identifiable characteristics

While we do not analyze how often these phishing attack URLs were distributed across Twitter, nor have any data of how often users fall victim to these attacks, their continuing prevalence across all of our datasets suggests that they might be effective, considering that attackers kept using them. We discuss some of their identifiable characteristics in this paragraph. For the phishing attacks hosted on Google and Microsoft Forms, along with the restrictive structure of designing these websites, both vendors also provide specific warnings to not enter private or sensitive information on websites hosted

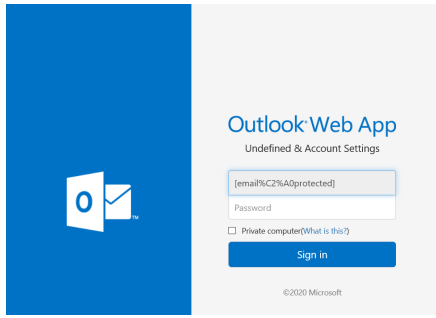


Figure 7: Phishing web application targeting Outlook Email login hosted on Google Cloud Storage API

by them. Microsoft also specifies that it is not affiliated with the owner of the website, and that all data entered on the forms will be sent directly to the author. These phishing attacks could be prevented if users pay attention to these disclaimers. For the Wix and Weebly URLs, based on their appearance alone, these websites appear to be much harder to recognize than Google and Microsoft’s form URLs, and unlike them, neither Wix nor Weebly provide any disclaimer to the user against entering sensitive information on these websites. Additionally, these websites often contained navigation bars location at either the header or footer of the websites, which contained links leading to the targeted organization’s website, thus further improving their false credibility. However both these vendors have a watermark which identifies that the respective service has been used to create these websites either at the top (for Wix URLs) or bottom (for Weebly URLs) of the website, which can be noticed accordingly in Figure 5. The phishing attacks on wordpress.com are much more sophisticated which makes them very difficult to detect based on appearance. Thus, the only way to identify them is to carefully study the URL string. Finally, the phishing attacks hosted on Google API are the most sophisticated and feature rich across all the approaches we have discussed so far, since the attacker has complete control over the front and back-end code of the phishing applications. There are also no indicators or disclaimers for the user to not enter their sensitive information on any applications hosted under this domain. Figure 7. It is worth mentioning that the websites hosted across all these services come with up-to-date SSL certifications. Thus, the only way to recognize all these threats is to pay close attention to the URL strings and be able to recognize they are different from their legitimate target organizations.

V. DRIVE BY DOWNLOADS ON TWITTER

Based on our manual evaluation in Section III-B, we found 267 URLs in our dataset, which hosted malicious drive-by downloads. We found that Twitter was able to detect an average of 17% of them at the end of the month across all the datasets and only 23% overall till the end of the study. Thus, 205 URLs were not detected by Twitter throughout the entire duration of our work. As illustrated in Table VI, 169 (82%) of the 205 undetected URLs originated from

one of three services: Microsoft Sharepoint, Wordpress and Blogspot. Similar to the phishing URLs that we observed in Section IV, all websites hosted under these services had up-to-date SSL certification. We used four popular antivirus vendors (as mentioned in Section III-B), whose detection label was used to categorize each file into one of nine different threat categories as illustrated in Fig 8. We found more than 71% of the threats targeted Microsoft Windows systems (including Trojans, Ransomware and Spyware), and 11% of these URLs included malicious Android APK files. Also nearly 15% of the downloads included malicious plugins (extensions) that were compatible with Google’s Blink engine (used in the Chromium family of web-browsers) and Gecko/Quantum (used in Mozilla’s Firefox web-browser and other similar open source alternatives).

A. Comparing Detected and Undetected Drive-by Download URLs

To check how other anti-phishing engines fare against these undetected drive-by URLs, we selected the 62 URLs which had been detected by Twitter and compared them with the 205 URLs which remained undetected throughout our study, then scanned them using VirusTotal. The descriptive statistics for this analysis is illustrated in Table VII. We found the undetected drive-by URLs were caught by an average of 5.92 engines ($\sigma=2.20$), whereas those which had been detected by Twitter at some point during the duration of the study maintained an average detection rate of 11.01 ($\sigma=2.92$). The undetected set of URLs were detected by 26 unique anti-phishing engines on VirusTotal, compared to 41 engines for the detected URLs. Fig 1(b) shows the cumulative distribution of engine detections towards these URLs. We find that nearly 40% of the engines were unable to catch 90% of the undetected drive-by download URLs. Only 4 engines were able to detect at least half of them, with 11 engines detecting less than 5% of these URLs, compared to 12 engines which were able to achieve the same feat for the drive-by URLs which were detected on Twitter. We also observed that nearly 83% of the undetected drive-by URLs remained active on Twitter after a month (and 76% at the end of our study) and are also detected by very few URL scanning engines. Additionally, from Table V, we found that 63% of these URLs remained alive on their respective hosting services even after a month. Again, similar to phishing URLs, we also observed these attacks are consistently present across all three of our datasets, signifying they have remained prevalent throughout the entire year that we conducted our experiments. We thus dedicate the following paragraphs to discuss the characteristics of these websites.

B. Microsoft Sharepoint

Microsoft Sharepoint is often used as a collaborative platform for sharing and working on documents or presentations online. We found that 18% of the URLs in our datasets, which distributed malicious downloads used this service, out of which more than 83% remained active even after a month.

Category	D1	D2	D3	A_{d1}	A_M
Sharepoint	18	38	41	93 (96%)	81 (83%)
Wordpress	5	9	11	25 (100%)	15 (60%)
Blogspot	13	15	17	37 (82%)	12 (27%)
Total	36	62	69	155 (91%)	108 (63%)

Table V: Distribution of undetected drive-by downloads URLs. A_{d1} = Active on day one, and A_M = Active after a month.

Dataset	Total	Detected after a month	Detected throughout the study
1	67	21(31%)	23(34%) ¹
2	109	17(16%)	31(28%) ²
3	91	8(9%)	8(9%) ³

Table VI: Descriptive stats for Drive-by download URLs detected across the three datasets. **Time periods 1=Jan to Dec 2020, 2=June to Dec 2020, and 3=Nov to Dec 2020**

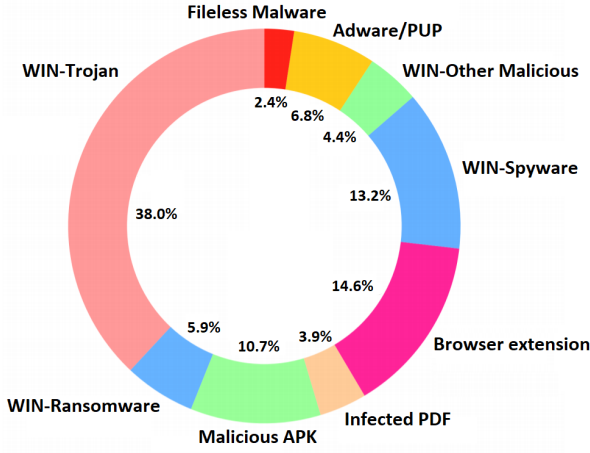


Figure 8: Categories of threats distributed by the Drive-by download URLs ($n=205$), which remained undetected on Twitter.

Nearly 76% of the Sharepoint URLs led directly to documents hosted on Microsoft OneNote containing links to the malicious downloads, which were usually executable files (Windows) disguised as a PDF or Microsoft Word documents. We also observed a unique two step attack vector in the remaining 24% of the URLs, which asks users to enter sensitive login information to get access to download a malicious file. The attack was usually carried out in the following manner: first, the user was asked to enter their account login information on a login form page created used Microsoft Office Online, which imitated one of several different legitimate organizations, as shown in Fig. 9(a), so that the user can gain access to the file. If the user complies and enters their information, they are provided with a download link for the document distributed using Microsoft OneNote as shown in Fig 9(b). Thus these URLs might also be classified as a hybrid phishing/drive-by download attack. Note that forms created using Microsoft Excel Online have very low customizability, even when compared with the URLs we

URL Set	Total	AD	Median	Min/Max	Std. Dev	Engines
Detected	205	5.92	6	1/17	2.20	26
Undetected	62	11.01	12	3/24	2.92	41

Table VII: URL Scanning Tool detection statistics for Detected and Undetected(on Twitter) Drive-by download URLs in our datasets. AD = average engine detections. $Engines$ =Total number of unique engines.

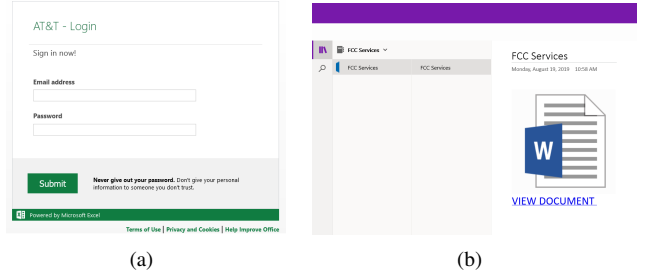


Figure 9: Two step attack hosted on Sharepoint, (a) Webform hosted using Microsoft Excel, which leads to (b) a malicious download on OneNote

discussed in IV-C. Users are only allowed to edit the title of the form page, and the text of the field prompts, and there is no functionality for inserting additional images/video content, or even changing the design theme. The default green Microsoft Excel theme is the only option. Also, Microsoft provides a detailed disclaimer similar to the one found in IV-C, urging users to not enter any sensitive information in these web forms. Interestingly, the majority of these two step attacks appear in D3, which suggests that it is a newer threat.

C. Wordpress

We found about 4.5% of the URLs hosted on wordpress.com distributed malicious downloads. Despite the appearance of these websites being far more sophisticated (similar to URLs discussed in 6) than the Sharepoint URLs, all of them were one step attacks, which distributed malicious executable files directly via a link embedded on the webpage and did not ask for any sensitive information. Wordpress was able to remove only 40% of these URLs over the course of a month. One such attack is illustrated in Figure 10. It is possible that, similar to the phishing URLs hosted on wordpress.com, these websites are not detected on Twitter due to the reputation of the domains.

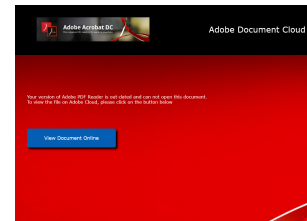


Figure 10: Example of a malicious Drive-by download URL imitating an Adobe Document reader page and hosted on Wordpress)

D. Blogspot

Eight percent of the URLs which distributed malicious domains and remained undetected by Twitter for over a month were hosted under Blogspot. Blogspot is Google’s free hosting service for creating online blogs. Attackers exploited this service by consistently posting malicious executable files which masqueraded as software or other downloadable resources. Based on our observation, the majority of the malicious downloads tried to tempt visitors by offering a ‘free’ or ‘patched’ version of a commercial software which could be downloaded at no additional cost. Our data suggests that Google is efficient at detecting these URLs, as they removed 79% of them over the period of the period of a month.

VI. TWITTER’S REACTION TOWARDS DETECTED MALICIOUS URLS

Among the 406 Phishing URLs that were detected by Twitter, 48% of the tweets were no longer accessible on the website at the end of the study. However, 52% of the URLs still remained active on the platform, despite generating a warning similar to the one in Figure 11. These tweets could be retweeted and sent as direct messages to other twitter users, despite them containing phishing URLs. These visible tweets were posted by 185 unique accounts who had 274.2 followers on average ($\sigma=103.9$). A similar trend was seen for the 62 URLs in our dataset which distributed drive-by downloads and were detected by Twitter. Twenty-nine of them were still active on Twitter at the end of our study, which could again be distributed freely across the platform and were posted by the same number of unique user accounts. These accounts had 563.8 followers on average ($\sigma=321.7$). We believe that this approach can encourage the easy distribution of malicious URLs across the platform, and since we have already found several malicious URLs in our dataset which were able to remain alive for a significant period of time, users can be exposed to these threats months after their initial appearance in the wild. Our experiment in this regard, however, is not conclusive, and should be pursued with more rigor to generalize our limited findings. Also considering that the accounts which posted these malicious URLs had several followers, a future study can be dedicated to determine how Twitter reacts to accounts which frequently post malicious URLs and re-distributes them as well among their network.

VII. BENIGN URLS ON TWITTER

During our manual investigation in Section III-B, coders found 2,816 (62%) of the all URLs across our three datasets (that had been flagged by at least one engine on VirusTotal) to be benign in nature. Twitter detected 293 (11%) of these benign URLs over the course of our study. The distribution of the detected benign URLs on Twitter is illustrated in Table VIII. We analyzed the characteristics of these URLs and our subsequent findings suggest that Twitter blocked websites from 6 URL shortening services and 4 free web hosting domains irrespective of whether the content of these websites

Warning: this link may be unsafe

The link you are trying to access has been identified by Twitter or our partners as being potentially spammy or unsafe, in accordance with Twitter’s [URL Policy](#). This link could fall into any of the below categories:

- malicious links that could steal personal information or harm electronic devices
- spammy links that mislead people or disrupt their experience
- violent or misleading content that could lead to real-world harm
- certain categories of content that, if posted directly on Twitter, are a violation of the [Twitter Rules](#)

[Back to previous page](#)

Ignore this warning and [continue](#)

Figure 11: Warning message generated by Twitter when it detected the URL as malicious.

were malicious or not. We elaborate more on this in the proceeding sections.

Dataset	Total	Detected after a month	Detected throughout the study
1	879	63(7%)	76(8%) ¹
2	1041	153(15%)	189(18%) ²
3	896	24(3%)	32(4%) ³

Table VIII: The distribution of benign URLs detected by Twitter as malicious (False positives). Time periods 1= Jan to Dec 2020, 2= June to Dec 2020, and 3= Nov to Dec 2020

A. URL Shortening services

One hundred seventy-eight URLs in datasets D1 and D2 were shortened by 7 different URL shortening services, which included *bitly.com*, *forms.gle*, *rebrand.ly*, *ow.ly*, *us.to*, *chng.it*. Anti-phishing tools often block the entire top/sub level domains of websites from which several malicious URLs originate from [63], [64]. We assume Twitter takes a similar approach, since we found that it provides a warning for any URL which uses one of several URL shortening services, irrespective of whether they are malicious or not. The distribution of the benign URLs by Twitter across the 7 different URL shortening services in D1 and D2 are shown in Table IX. In fact, to further confirm our suspicion that Twitter was detecting the URL based on the name of the domain only, we performed a small experiment where we created some URL strings which do not exist (for example, *forms.gle/this-is-not-a-real website*). We found that Twitter immediately detected these non-existent URLs as malicious (‘potentially harmful’) by providing a warning similar to Fig. 11.

During the collection of our final dataset in November 2020 (D3), we found that Twitter had stopped detecting benign URLs hosted by 4 of these services but were still detecting URLs from the remaining three services consisting of *forms.gle* (17), *bitly.com* (8) and *chng.it* (3).

Web Hosting Domains. We found 117 URLs being hosted across 4 free web hosting domains, including *000webhost*,

URL Shortener	URLs	URL Shortener	URLs
bitly.com	54	000webhost	63
rebrand.ly	26	hpage	28
ow.ly	23	freehostia	20
chng.it	35	Atspace	6
forms.gle	31		
us.to	9		

Table IX: **(On left)** URL Shortening services detected as malicious by Twitter throughout D1 and D2. **(On right)** Web Hosting services detected by Twitter

hpage, *freehostia*, and *atspace* which had been detected by Twitter as malicious. We assume these free web-hosting services are detected for the same reason that URL shortening services were detected, i.e. reputation of the domain. Indeed, similar to the URL shortening services, Twitter warns against even nonexistent URLs if they are using one of these three services. The distribution of the detected URLs across the four hosting domains are illustrated in Table IX. Similar to the URL shortening services, this area too saw an improvement on Twitter’s end towards the final stages of our study, with Twitter no longer detecting URLs generated from 000webhost and Atspace; however, it still detected websites hosted under *hpage* (3) and *freehostia* (1).

To summarize, our findings in this section indicate that Twitter detected URLs hosted by 6 URL shortening services, and 4 free web hosting domains over at least the first half of the year (up until the analysis of D2), irrespective of the content of these websites. We assume that Twitter did so based on the reputation of these services, though it is hard to ascertain it because of the former’s blackbox nature of detection. We do find that they are improving their allowlist towards not detecting several URLs hosted by these services towards the end of our study (in dataset D3). This further justifies that a similar improvement is required in updating their blocklist to better protect against the threats that we have discussed in Sections IV and V.

VIII. CONCLUSION AND DISCUSSIONS

In this work, we evaluated and characterized more than 1.2k unique malicious URLs that remained undetected on Twitter for a significant period of time from the date of their first appearance in our dataset. We have been able to establish the majority of these URLs are hosted over nine different website/web application creation services. Our findings indicate these services are favorable to attackers for the following reasons: 1) Twitter’s apparent inability to detect and block several of the URL threats in our dataset is also shared by many other anti-phishing tools and blocklists, with nearly 80% of them unable to block above 90% of phishing threats the same URLs that Twitter was unable to detect. Additionally, the website creation services themselves are not very efficient at removing these URLs, with more than 57% of phishing URLs and 63% of drive-by download URLs remaining active even after a month of these URLs appearing in our dataset. Both of these factors can allow attackers to organize successful

phishing campaigns over extended periods of time.3) During our study, we found Twitter did not remove several malicious URLs which it had detected, allowing for them to be easily redistributed over the platform at a later time. 4) All the website creation services in consideration have simple drag and drop interfaces to create webpages, are free to use and come with up-to date SSL certification.

Though Twitter and several URL scanning tools use black-box approaches which are unknown to us, our findings strongly hint towards the fact the sparse detection of the majority of the malicious URLs in our dataset can be contributed to the reputation of the domains that the website creation services hosting them provide. Thus, from Twitter’s or the URL scanning tools’ end, allowlisting websites using these services, without considering their content is far from an ideal practice. On the other hand Twitter also detected websites using one of 6 URL shortening services and 4 web hosting domains, irrespective of their content. But, their improvement in this regard towards the end of our study indicates that Twitter has improved their blocklisting approach when compared to their allowlisting implementations towards real malicious websites.

IX. LIMITATIONS AND FUTURE WORK

Age of malicious URLs. For the URLs in our dataset that turned out to be malicious, we considered the day the URL first appeared online as the day the tweet containing it was posted. This may not always be the case, as the URL might have appeared in the wild at an earlier time and also possibly in an older tweet that was not part of our dataset.

Effectiveness of undetected malicious URLs. The majority of the malicious websites (61%) in our study did not invoke any warnings from Twitter, and it would be interesting to observe how users interact with them and if they are more likely to be affected by the attacks we have illustrated in IV and V. In Section IV-G, we mentioned several features that help users in identifying the URLs as phishing, but there is a need to determine if users are actually able to recognize these indicators.

Impact of warnings against Benign URLs. More than 11% of the benign URLs were detected by Twitter. Adherence to these warnings might deter the users from visiting genuine and useful resources which might be inconvenient for them or for the owners of the website (for example lower traffic for an e-commerce websites). In the future, we plan to extend these nascent findings into a concrete analysis to determine how users react to these warnings and if there is significant decrease of traffic to the affected websites.

REFERENCES

- [1] “Why Social Media is Increasingly Abused for Phishing Attacks,” <https://info.phishlabs.com/blog/how-social-media-is-abused-for-phishing-attacks>, 2020.
- [2] E. D. Fraenstein and S. V. Flowerday, “Social network phishing: Becoming habituated to clicks and ignorant to threats?” in *2016 Information Security for South Africa (ISSA)*. IEEE, 2016, pp. 98–105.
- [3] S. Bell, K. Paterson, and L. Cavallaro, “Catch me (on time) if you can: Understanding the effectiveness of twitter url blacklists,” *arXiv preprint arXiv:1912.02520*, 2019.

- [4] A. Sanzgiri, A. Hughes, and S. Upadhyaya, "Analysis of malware propagation in twitter," in *2013 IEEE 32nd International Symposium on Reliable Distributed Systems*, 2013, pp. 195–204.
- [5] C. Chen, S. Wen, J. Zhang, Y. Xiang, J. Oliver, A. Alelaiwi, and M. M. Hassan, "Investigating the deceptive information in twitter spam," *Future Generation Computer Systems*, vol. 72, pp. 319–326, 2017.
- [6] A. Sanzgiri, J. Joyce, and S. Upadhyaya, "The early (tweet-ing) bird spreads the worm: An assessment of twitter for malware propagation," *Procedia Computer Science*, vol. 10, pp. 705–712, 2012.
- [7] D. A. Broniatowski, A. M. Jamison, S. Qi, L. AlKulaib, T. Chen, A. Benton, S. C. Quinn, and M. Dredze, "Weaponized health communication: Twitter bots and russian trolls amplify the vaccine debate," *American journal of public health*, vol. 108, no. 10, pp. 1378–1384, 2018.
- [8] A. Javed, P. Burnap, and O. Rana, "Prediction of drive-by download attacks on twitter," *Information Processing & Management*, vol. 56, no. 3, pp. 1133–1145, 2019.
- [9] M. Bossetta, "A simulated cyberattack on twitter: Assessing partisan vulnerability to spear phishing and disinformation ahead of the 2018 us midterm elections," *arXiv preprint arXiv:1811.05900*, 2018.
- [10] J. Seymour and P. Tully, "Weaponizing data science for social engineering: Automated e2e spear phishing on twitter," *Black Hat USA*, vol. 37, pp. 1–39, 2016.
- [11] Y. Tanaka, M. Akiyama, and A. Goto, "Analysis of malware download sites by focusing on time series variation of malware," *Journal of computational science*, vol. 22, pp. 301–313, 2017.
- [12] R. Masri and M. Aldwairi, "Automated malicious advertisement detection using virustotal, urlvoid, and trendmicro," in *2017 8th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2017, pp. 336–341.
- [13] F. Wei, Y. Li, S. Roy, X. Ou, and W. Zhou, "Deep ground truth analysis of current android malware," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2017, pp. 252–276.
- [14] F. O. Catak and A. F. Yazı, "A benchmark api call dataset for windows pe malware classification," *arXiv preprint arXiv:1905.01999*, 2019.
- [15] P. Prasse, L. Machlica, T. Pevný, J. Havelka, and T. Scheffer, "Malware detection by analysing encrypted network traffic with neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 73–88.
- [16] N. Alshahwan, E. T. Barr, D. Clark, G. Danezis, and H. D. Menéndez, "Detecting malware with information complexity," *Entropy*, vol. 22, no. 5, p. 575, 2020.
- [17] "1.4 million phishing websites are created every month," <https://www.zdnet.com/article/1-4-million-phishing-websites-are-created-every-month-heres-who-the-scammers-are-pretending-to-be/>, 2020.
- [18] "Phishing Scams Cost American Businesses Half A Billion Dollars A Year]," <https://www.forbes.com/sites/leemathews/2017/05/05/phishing-scams-cost-american-businesses-half-a-billion-dollars-a-year/#7a1b70773fa1>, 2020.
- [19] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: their types, vectors and technical approaches," *Expert Systems with Applications*, vol. 106, pp. 1–20, 2018.
- [20] A. Aleroud and L. Zhou, "Phishing environments, techniques, and countermeasures: A survey," *Computers & Security*, vol. 68, pp. 160–196, 2017.
- [21] S. Gupta, A. Singhal, and A. Kapoor, "A literature survey on social engineering attacks: Phishing attack," in *2016 international conference on computing, communication and automation (ICCCA)*. IEEE, 2016, pp. 537–540.
- [22] M. Butavicius, K. Parsons, M. Pattinson, and A. McCormac, "Breaching the human firewall: Social engineering in phishing and spear-phishing emails," *arXiv preprint arXiv:1606.00887*, 2016.
- [23] B. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, "Fighting against phishing attacks: state of the art and future challenges," *Neural Computing and Applications*, vol. 28, no. 12, pp. 3629–3654, 2017.
- [24] A. Sumner and X. Yuan, "Mitigating phishing attacks: An overview," in *Proceedings of the 2019 ACM Southeast Conference*, 2019, pp. 72–77.
- [25] A. Tewari, A. Jain, and B. Gupta, "Recent survey of various defense mechanisms against phishing attacks," *Journal of Information Privacy and Security*, vol. 12, no. 1, pp. 3–13, 2016.
- [26] B. B. Gupta, N. A. Arachchilage, and K. E. Psannis, "Defending against phishing attacks: taxonomy of methods, current issues and future directions," *Telecommunication Systems*, vol. 67, no. 2, pp. 247–267, 2018.
- [27] S. P. Chorghe and N. Shekhar, "A survey on anti-phishing techniques in mobile phones," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 2. IEEE, 2016, pp. 1–5.
- [28] H. Sharma, E. Meenakshi, and S. K. Bhatia, "A comparative analysis and awareness survey of phishing detection tools," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, 2017, pp. 1437–1442.
- [29] A. Oest, Y. Safei, A. Doupe, G.-J. Ahn, B. Wardman, and G. Warner, "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in *2018 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2018, pp. 1–12.
- [30] M. Moghimi and A. Y. Varjani, "New rule-based phishing detection method," *Expert systems with applications*, vol. 53, pp. 231–242, 2016.
- [31] J. Mao, W. Tian, P. Li, T. Wei, and Z. Liang, "Phishing-alarm: robust and efficient phishing detection via page component similarity," *IEEE Access*, vol. 5, pp. 17020–17030, 2017.
- [32] G. Sonowal and K. Kuppusamy, "Phidma—a phishing detection model with multi-filter approach," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 99–112, 2020.
- [33] C. L. Tan, K. L. Chiew, K. Wong, *et al.*, "Phishwho: Phishing webpage detection via identity keywords extraction and target domain name finder," *Decision Support Systems*, vol. 88, pp. 18–27, 2016.
- [34] H. Shirazi, B. Bezawada, and I. Ray, "kn0w thy domain name" unbiased phishing detection using domain name based features," in *Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies*, 2018, pp. 69–75.
- [35] H. Y. Abutair and A. Belghith, "Using case-based reasoning for phishing detection," *Procedia Computer Science*, vol. 109, pp. 281–288, 2017.
- [36] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.
- [37] T. Peng, I. Harris, and Y. Sawa, "Detecting phishing attacks using natural language processing and machine learning," in *2018 IEEE 12th international conference on semantic computing (icsc)*. IEEE, 2018, pp. 300–301.
- [38] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Computing and Applications*, vol. 31, no. 8, pp. 3851–3873, 2019.
- [39] P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang, and T. Zhu, "Web phishing detection using a deep learning framework," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
- [40] R. Hassanpour, E. Dogdu, R. Choupani, O. Goker, and N. Nazli, "Phishing e-mail detection by using deep learning algorithms," in *Proceedings of the ACMSE 2018 Conference*, 2018, pp. 1–1.
- [41] N. Abdelhamid, F. Thabtah, and H. Abdel-jaber, "Phishing detection: A recent intelligent machine learning comparison based on models content and features," in *2017 IEEE international conference on intelligence and security informatics (ISI)*. IEEE, 2017, pp. 72–77.
- [42] E. Buber, Ö. Demir, and O. K. Sahingoz, "Feature selections for the machine learning based detection of phishing websites," in *2017 international artificial intelligence and data processing symposium (IDAP)*. IEEE, 2017, pp. 1–5.
- [43] A. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," in *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*. IEEE, 2017, pp. 1–5.
- [44] "VirusTotal," <https://www.virustotal.com/gui/home/>, 2020.
- [45] "VirusTotal API," <https://support.virustotal.com/hc/en-us/articles/115002100149-API>, 2020.
- [46] J. Charlton, P. Du, J. Cho, and S. Xu, "Measuring relative accuracy of malware detectors in the absence of ground truth," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, 2018, pp. 450–455.
- [47] B. Sun, M. Akiyama, T. Yagi, M. Hatada, and T. Mori, "Automating url blacklist generation with similarity search approach," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 4, pp. 873–882, 2016.
- [48] S. Zhu, J. Shi, L. Yang, B. Qin, Z. Zhang, L. Song, and G. Wang, "Measuring and modeling the label dynamics of online anti-malware engines," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.
- [49] H. Wang, J. Si, H. Li, and Y. Guo, "Rmvdroid: towards a reliable android malware dataset with app metadata," in *2019 IEEE/ACM 16th Interna-*

- tional Conference on Mining Software Repositories (MSR). IEEE, 2019, pp. 404–408.
- [50] “Twitter Developer[.]” <https://developer.twitter.com/en/docs>, 2020.
 - [51] P. Peng, L. Yang, L. Song, and G. Wang, “Opening the blackbox of virustotal: Analyzing online phishing scan engines,” in *Proceedings of the Internet Measurement Conference*, 2019, pp. 478–485.
 - [52] “84% of Phishing Sites Last for Less Than 24 Hours[.]” <https://www.infosecurity-magazine.com/news/84-of-phishing-sites-last-for-less/>, 2020.
 - [53] “More Than Half of Phishing Sites Now Use HTTPS[.]” <https://info.phishlabs.com/blog/more-than-half-of-phishing-sites-use-https>, 2020.
 - [54] “What is a drive by download[.]”
 - [55] “How malware works: Anatomy of a drive-by download web attack[.]”
 - [56] “Why do not you include statistics comparing antivirus performance?[,]” <https://support.virustotal.com/hc/en-us/articles/115002094589-Why-do-not-you-include-statistics-comparing-antivirus-performance->, 2020.
 - [57] “VirusTotal is not a Comparative Analysis Tool![,]” <https://www.welivesecurity.com/2008/09/12/virustotal-is-not-a-comparative-analysis-tool/>, 2008.
 - [58] “VirusTotal += Behavioural Information[.]” <https://blog.virustotal.com/2012/07/virustotal-behavioural-information.html/>, 2012.
 - [59] “AV-Comparatives Summary Report 2019[.]” <https://www.av-comparatives.org/tests/summary-report-2019/>, 2019.
 - [60] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang, “Knowing your enemy: understanding and detecting malicious web advertising,” in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 674–686.
 - [61] “Hosting malicious sites on legitimate servers: How do threat actors get away with it?[,]” <https://blog.malwarebytes.com/cybercrime/malware/2019/01/hosting-malicious-sites-legitimate-servers-threat-actors-get-away/>, 2020.
 - [62] “How scammers are hiding their phishing trips in public clouds [.]” <https://blog.checkpoint.com/2020/07/21/how-scammers-are-hiding-their-phishing-trips-in-public-clouds/>, 2020.
 - [63] “Block that top level-domain[.]” <https://bluecatnetworks.com/blog/block-that-top-level-domain>, 2020.
 - [64] “Blacklisted URL Shorteners - Stop Using Them in Emails![,]” <https://support.rebrandly.com/hc/en-us/articles/228632488-Blacklisted-URL-Shorteners-Stop-Using-Them-in-Emails->, 2020.