# “HOW GOOD ARE THEY?”

## A STATE OF THE EFFECTIVENESS OF ANTI-PHISHING TOOLS ON TWITTER

by

**SAYAK SAHA ROY**

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

University of Texas at Arlington

August 2020

# Abstract

**“How Good Are They?:”**
A State of the Effectiveness of Anti-Phishing Tools on Twitter

**Supervisor:** Dr. Shirin Nilizadeh

**Committee Members:** Dr. Chengkai Li and Dr. Jiang Ming

Phishing websites are one of the most pervasive online attack vectors, with nearly 1.5 million such attacks created every month. Social media is the primary ground for phishing attacks, with 86% of these attacks originating from Twitter, Facebook, LinkedIn, etc. Prevalent approaches against these attacks includes URL scanners, anti-phishing blacklists and social media's own detection systems. In this work, we focus on Twitter, and through a combination of data-driven methods and emulations, we evaluate the verdicts provided by URL scanners, and Twitter's detection system.

We show that these sources provide a good amount of misinformation, which not only can lead users to visit malicious websites, but also can decrease the web traffic to legitimate websites. In particular, we analyzed 40k unique URLs obtained from 1 million tweets by grouping them into 5 different categories based on their characteristics, and found that Twitter is consistently unreliable at labelling URLs from 2 different categories - Phishing websites hosted under trusted domains, Benign URLs which are hosted under suspicious URL Shortening services or free Web Hosting domains. We also found that for accounts that continuously post malicious links, Twitter does not proactively suspend or remove them, instead relying heavily on users reporting these accounts before they are suspended. We also found about 71% of the URLs detected by URL Scanning engines as malicious were actually benign, which was caused due to three factors which frequently lead to false positives in these tools. These factors are URL domain bias, web hosting bias and reliance on PhishTank, a public URL blacklist. We also discovered a new form of phishing attack which leverages the use of popular web domains and further implements two obfuscation methods, to remain undetected for more than a month. Finally, we conducted an IRB approved survey study on Amazon Mechanical Turk, where we evaluated the impact of the misinformation, provided by both URL Scanning engines and Twitter, on the perception of users about the websites. We found that users have more confidence on certain URL features more than others, and in the later case, they heavily rely on the detection tools, irrespective of whether their verdict was right or not.

# Acknowledgements

I would like to thank Dr. Shirin Nilizadeh, my Thesis supervisor, who gave me the opportunity to work on such an interesting project, and guided me in an exemplary manner throughout. Her motivation has consistently piqued my interest in research, and I have learned a lot over the last two years working under her.

I would also like to extend my gratitude towards my Thesis Committee Members, Dr. Chengkai Li and Dr. Jiang Ming, for their time and immensely constructive feedback.

My heartfelt love and appreciation for my parents, Ashima Saha Roy and Sanjit Kumar Saha Roy and my sister, Tiasha Saha Roy, my pillars of strength and character, who have continuously encouraged me to cultivate my knowledge in this field, which eventually lead me down this memorable path.

Finally, a huge thanks to the teams at Twitter Developer and VirusTotal for giving me access to their respective Academic APIs, which helped me in my data collection and analysis.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# Introduction

Over the last decade, social networking websites, such as Twitter, Facebook and LinkedIn have seen a steady increase with respect to user exposure to malicious URLs, especially phishing websites. Attackers find these platforms alluring because the abundance and easy propagation of information through users' social networks (i.e., friends and followers) which increases the exposure of these malicious URLs [26]. The detection and prevention solutions against these forms of attacks is provided by social media platforms themselves as well as third-party URL Scanning Tools. For example, while Twitter previously depended on Google's Safe Browsing blacklist [74] to detect malicious URLs, they have recently moved away from relying on the latter [11], making Twitter's detection system a blackbox. URL scanners such as BitDefender [12], Fortinet [25], Kaspersky [36], ESET [24], and CRDF [23] maintain their own blacklist, with some also relying on their proprietary heuristics and machine learning based solutions as well.

There is a huge body of work studying the problem of malicious websites on the web. However, most work done in regarding Malicious URLs in the Twitter ecosystem [55, 20, 56, 15, 33] deal with trying to identify how common and identifiable malware and phishing attacks spreads through user networks. Some studies have also replicated many phishing attacks [13, 57] to study their characteristics. In contrast, our work is mostly focused on determining and characterizing phishing and other malicious threats which are not (or sparsely) detected by both Twitter and anti-phishing/URL scanning engines. We also focus on Benign URLs which are also misdetected by URL Scanners due to several factors that we discuss throughout this work.

We aim to determine the effectiveness of both Twitter and URL scanners/ blacklists against the newest URL threats posted on the platform. In particular, this work:

1. Evaluates the URLs that are posted on Twitter, and identifies the response of Twitter to malicious URLs, specially the phishing URLs, as well as Benign URLs;

2. Through several passive and active experiments, tries to identify the categories of malicious URLs that are not detected, by either Twitter or third-party URL scanner tools.

3. Examines the characteristics of *Benign* URLs that are detected as malicious URLs by either Twitter or third-party URL scanners.

4. Determines how Twitter keeps their platform free from URL threats by both detecting newer links and preventing malicious accounts from frequently posting these links.

5. Conducts a user study to examine the impact of misinformation provided by detection tools on users' perception of websites. This user study highlights the affirmative impact of verdicts by detection tools, especially when the URLs are falsely labeled.

6. Through the user study, identification of the factors that real-life Internet users check to identify if a website is legitimate. These information can help in developing new methods for informing the users.

Figure 1.1 provides a brief overview of the steps we have taken in this work. We analyzed more than 40k unique URLs collected from tweets. We used VirusTotal, one of the most popular URL Scanning tools, used by both regular users and the research community alike [66, 42, 81, 18, 50, 8] to ascertain the validity of new URLs. VirusTotal aggregates the score of 80 different URL Scanning engines, which gives a broad view of the scenario of how well they can detect these threats. We also manually investigated and categorized 1.3k URLs detected as malicious by the tool. We compared the detection of both Twitter and 80 different URL Scanning Engines against new Malicious URLs collected from 3 different public blacklists, to determine their efficacy, as well as consistently reoccurring patterns of misdetecting or mislabeling certain URLs. We then tracked how Twitter detected and blocked malicious URLs, especially phishing URLs, by emulating the behaviour of malicious accounts and continuously posting tweets which contained infected URLs, while also observing the action that Twitter takes against these accounts. We also studied the Benign URLs, which constituted a large part of our dataset, and were detected as malicious by both URL Scanning engines and VirusTotal, and found three factors which lead to such misinformation. These misinformation factors which caused both False Positives and False Negatives led us to an experimentation where we observed the impact of public blacklists on the detection of certain categories of URLs by popular URL Scanning tools. Finally, we conducted a survey study on Amazon M-Turk, where we queried 120 participants to rate both malicious and benign URLs and gauged their trust on these Anti-Phishing Tools and Blacklists. We found that respondents had confidence on certain characteristics of a website, whereas for those for which they had little confidence on, more often than not, they relied on the verdict of the Anti-Phishing tools.

Our findings can be summarized as below:

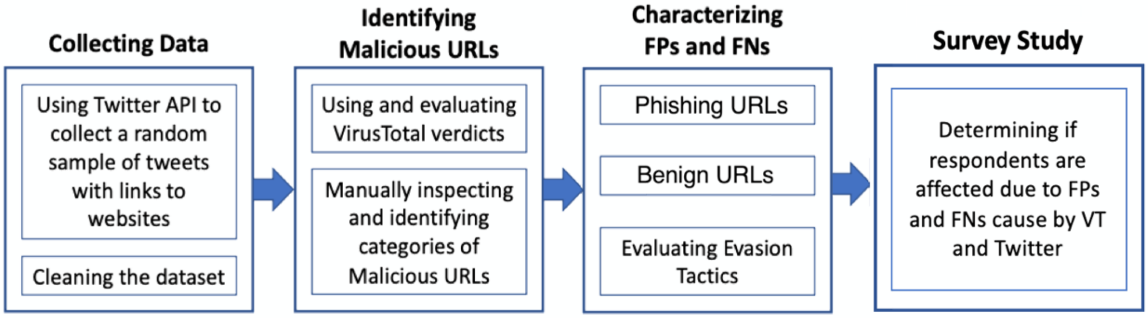

Collecting Data
Using Twitter API to collect a random sample of tweets with links to websites
Cleaning the dataset
Identifying Malicious URLs
Using and evaluating VirusTotal verdicts
Manually inspecting and identifying categories of Malicious URLs
Characterizing FPs and FNs
Phishing URLs
Benign URLs
Evaluating Evasion Tactics
Survey Study
Determining if respondents are affected due to FPs and FNs cause by VT and Twitter

FIGURE 1.1: An overview of our study about misinformation in malicious URLs on Twitter

1. Twitter is initially efficient at blocking URL threats, especially Phishing and Malicious Ad-Gateway URLs, only failing to detect 0.9% of malicious tweets in our dataset, but has a higher false positive rate of 6.5% towards legitimate links. However, when testing against newer URLs from public blacklists, Twitter has a detection rate of only 44%, suggesting that they are more effective at detecting Malicious URLs which are more than a day old.

2. We created several dummy Twitter accounts, which continuously posted malicious URLs for over a month, and observed that Twitter did not suspend them, only doing so when these accounts were manually reported by other Twitter accounts. Also, we found that Twitter relies on manual reporting for a category of URLs that is not easily found on URL blacklists. For the URLs that Twitter did not initially prevent us from posting, if it later detected them, they were not removed, but hidden, with the tweets being visible to the authors.

3. We found a new category of Phishing URLs, which we call *Web Form URLs*, who leverage the use of the Google Forms and Google Sites platforms to build web-pages which can steal user data. Twitter fails to detect tweets containing this category of Phishing URLs, even after they have been active for more than a month. Also, URL Scanners have a very low detection rate for this category of Phishing URLs as well, with a few of them detecting these URLs due to their reliance on Public blacklists, and not due to their own detection capabilities.

4. We could correlate that links posted on PhishTank, a public URL blacklist, and the detection of certain URL scanning engines, suggesting their partial reliance on them. This not only affects malicious URLs, but also benign URLs which are posted on the blacklist, resulting in false positives in several prominent URL scanning engines.

5. We found a huge majority of our dataset(71%) who had been detected as malicious by Twitter, were actually benign. For these URLs, we found three characteristics which led to them being undetected:i) They were using one of six URL Shorteners which are detected by both URL Scanners and Twitter ii) Similarly, URLs which were using one of five free web hosting domains which are detected by the Anti-Phishing tools. We found for i) and ii) that Anti-Phishing Tools often use very conservative approaches, often checking only the top level domains of these URLs and then flagging them as malicious, irrespective of the content of the website. iii) A large number of the Benign URLs were found on PhishTank, a public blacklist, leading to them being detected. We run our own simulation on VM systems to find that Anti-Phishing Tools partially rely on PhishTank, resulting in often detecting benign URLs as Malicious. This leads to a possible new attack vector which we explore in this work also.

6. Finally, through our survey study hosted on Amazon M-Turk, we found our respondents were impacted by the misinformation spread by these Anti-Phishing Tools, but, we found that this reliance is dependent on certain characteristics of the website. To measure these characteristics quantitatively, we create a new metric, *Impact Factor* which measures the extent to which respondents trust a certain website features(such as SSL Certification), and if they are confused by certain features as well, which leads to them solely relying on the verdict of the Anti-Phishing Tools.

# Chapter 2

# Background and Related Work

### Phishing Attacks

Phishing websites are designed in such a way that they imitate other popular websites, in order to trick users into sharing their personal or financial information. The scale of damage by these attacks is massive, with more than 1.4 million such websites being created every month [1]. Established academic literature is plentiful in this regard, with researchers tracking several techniques used by these attackers. [21] and [63] have done in-depth analysis on the characteristics of Phishing approaches and how prevalent they are in the online sphere, concluding that indeed these attack techniques are not only successful, but are also very widespread. On the other hand, [7] provides one of the first notable works where the focus on Phishing websites shifts from the traditional email based attacked vectors to mobile and social networks. They propose a taxonomy for phishing detection techniques with the intention of laying the groundwork for improving phishing detection tools. [22] has extended this work by focusing just on the mobile platform, noting that due to the ease in accessibility and exponential growth in usage over the last few years, Phishing attackers target mobile devices more. The authors further found that prevalent Anti-phishing measure on mobile devices are not sufficient for combating the huge onslaught of these phishing attacks. They thus concluded that improvement is required both in the Ant-phishing tools as well as user awareness. Similarly, the work in [30] surveys phishing websites to determine various strategies that are used by attackers to implement social engineering phishing attacks that leverages several unique methods such as Tab-napping, Spoofing emails etc. to increase their fatality. The authors followed this up in [29] by focusing on the prevalent Anti-phishing detection tools and data-sets used by researchers and how efficient they are.Focusing on social engineering attacks in emails, [17] empirically observed how regular users react to several spoofed emails, and found that users were more susceptible to these attacks, when they imitated authority. Also, they found that users who were less impulsive tended to judge the email links more carefully. This suggests that education in the field of online security is a necessity for self-awareness and protection against these attacks.Corroborating this statement, [59] stated that while Anti-phishing tools are accurate while detecting phishing websites, from a survey study that they conducted, it was found that 61% of respondents were not aware of Anti-Phishing tools, and thus did not avail of their protection. [45] takes the claim of the accuracy of Anti-Phishing tools, and performed a unique study where they found that Attackers are wary of the detection techniques of these tools, and thus, continuously try to change their attack approaches to hoodwink the latter.

Significant progress has also been made towards detecting phishing websites, by bringing improvements beyond the traditional URL blacklists that most Anti-Phishing tools use. [65] and [60] take 2 unique approaches, with the former doing an N-gram analysis between the names of the legitimate websites and their imitated phishing counterparts to determine their differences and then trigger a search query which tries to find the domain that the phishing page is hosted on. The later depends on a feature set for the domain that the phishing website is hosted on. This approach of cross-checking URLs to determine their legitimacy proves to be much more efficient at detecting Phishing websites than traditional blacklist approaches.

[44] have presented a new rule-based approach which uses several features to determine characteristics in both the content and URL of the webpage. They train an SVM model using these features, as well amalagamating it with a String matching algorithm to create an Anti-phishing browser extension called PhishDetector, which achieves a 99% accuracy while detecting phishing websites. [41] has a similar approach where they evaluate page component similarity in websites to detect suspicious patterns.[61] further expands these works by using a five feature layer detection mechanism, taking a more comprehensive look into all the website features.Despite having a much smaller training set, they were able to achieve an accuracy of 92%. Finally, to account for the ever-changing terrain of phishing attacks, and their short lifetime, [6] takes a very unique approach of using a Cased based reasoning system to detect very new phishing threats and achieves an accuracy of 95%.

Beyond the more traditional usage of categorizing URL and website features, recent work has also relied on novel machine learning approaches to detect Phishing websites. [31] utilizes the word2vec model to generate a vector representation of Phishing email content(instead of relying on keywords only), and then fed this data to a neural network, which was able to detect 96% of the phishing emails succesfully. [54] and [48] use a model based on Natural Language Processing(NLP), training it on an enormous set of URLs to acquire language based data. The former approach led to an accuracy of 98% which indicates the efficacy of their model. [52] uses a Random Forest Classifier by utilizing heuristic features based on URL Source code and information from third-party services to build their model which scores an accuracy of 99%. Taking a different direction, [82] builds a Deep Belief Network and utilizes two improvised website features to train their model. Instead of relying on simple website features, they instead look at IP flows to and from the website to detect if it is sending some suspicious data packets. Finally, [4] compares these approaches and suggests optimium ways to integrate them into prevalent Anti-phishing solutions.

Thus, considering the resulting improvements that have been made in detecting phishing attacks over the last few years, our work is mainly focused on a single, yet popular platform for spreading phishing and other malicious URLs alike - Twitter, and we aim to evaluate how both Twitter and several anti-phishing tools perform when they are exposed to these threats on this platform.

**URL Scanning Tools.**

In recent times, anti-phishing tools and URL scanning tools are synonymous, with both of them going beyond detecting just phishing URLs, instead also focusing on malicious Ad Gateways, social engineering scams, spam, drive-by downloads, etc. These tools maintain databases known as URL blacklists containing list of URLs which are harmful, and frequently update said list, adding newer URLs in the process. Usually, URLs on these blacklists are verified by skilled personnel. However, in some cases such as PhishTank [49] anyone can post and verify links. Beyond Phishing websites, research is also abundant regarding these other forms of Malicious URLs and online web attacks. [34] explored the inclination of regular users towards providing sensitive shopping information to an unknown surveying entity. The authors suggested that providing sensitive information to a medium without knowing the trustworthiness of it can lead to social engineering scams. This susceptibility to provide sensitive information was further studied in [16] where the authors found that persuasion plays a huge role in convincing the users. In our work we observe that Malicious Ad-Gateways are seldom detected by Anti-phishing tools due to them being being indirect sources of spreading malicious content. [42] combines the detection capabilities of three Anti-phishing services-VirusTotal, URLVoid and TrendMicro, to build an Anti-phishing model which can detect Malicious Ad Gateways. However, the accuracy for this model was only 73% and thus to further increase this accuracy, [62] suggests a novel system which extracts a SWF file from a webpage and then use their analysis engine to generate a risk rating of the URL based on several factors. For Drive-by Downloads, [51] found that several of these forms of attacks stay undetected even after 2 years. This flaw is caused due to Scanning engines depending on labelling each file separately. Thus, the authors develop a new rule based system which learns from the already available ground-truth, and use that to differentiate between both Malicious and Benign downloads.

The input from these works has led to the Anti-phishing tools evolving from being solely dependent on blacklists, moving forward to use heuristic methods to better detect these URLs. Our work mostly deals with phishing URLs, but we also investigate other categories, including Ad Gateways and drive-by downloads, finding co-relations between their detection and other factors on both Twitter and URL scanning tools.

### VirusTotal

VirusTotal [80] is an online URL scanning tool which scans both files (for malicious code) and URLs (for malicious content). After a user submits a file (or URL), VirusTotal checks with 80 different URL scanning engines to see which ones label it as a threat and then reports the aggregated score as well as the engines which detected the URL, back to the user. VirusTotal has an online web interface [80], in addition, it also provides an API available [79] for scanning a large volume of URLs.

Researchers have frequently used this tool to label URLs in their dataset, with several works depending on it to create their ground truth for Malicious URLs.[19] estimates the relative detection accuracy of several Anti-Malware engines by comparing them using the detection metrics of VirusTotal for those samples as their ground truth. [64] and [66] also use VirusTotal as one of the sources to classify URL which distributes Drive-by downloads. As mentioned earlier, [42] combines the detection data from VirusTotal along with several other scanning sources to build an Anti-Phishing tool. We use VirusTotal in a similar way, using it to determine what portion of scanners detect the newer URLs posted on Twitter, and also drawing a comparative analysis between several scanning engines. However, recent development in studying this tool [47] has found that it exhibits some inconsistencies when compared to the detection label of respective scanners, with often VirusTotal's report for a particular engine lagging behind the engine's detection label found on it's own website or tool. The authors found that this occurs due to VirusTotal using stripped down versions of the URL Scanning engines that participate in it's program. Considering that these inconsistencies can lead to errors in academic research, [83] performs an analysis where they survey 115 academic papers(which depend on VirusTotal to label their ground truth). They get a broader picture of the thresholds and features which researchers rely on for aggregating VirusTotal reports, and illustrates the inaccuracies of these practices. Finally, they suggest certain precautionary measures which can aid in mitigating the inconsistency errors in future VirusTotal data analysis. Taking these findings into account, we designed our methods in such a way, so as to make sure that the data we acquire from VirusTotal is checked for accuracy.

### Amazon Mechanical Turk

Amazon Mechanical Turk [9] is a crowd-sourcing platform where real-life users(called Workers) can be assigned to manual tasks which cannot easily or efficiently be automated with the help of machines. These tasks mostly include data labelling and surveys. Individuals or organizations(called Requesters) who want to hire Workers to perform their tasks can do so on this platform in exchange of monetary compensation. Amazon M-Turk is popular due to its ease of finding an enormous roster of individuals from various demographic background for working on data. It is used very frequently in Academic research, and is popular in the field of Data Security and Privacy. [35] uses data obtained from respondents on this platform to find the effect of maliciously compromising user systems. On the other hand, [43] evaluates user data from this platform to determine how users are susceptible to authoritative URL messages from spammers as well as they how they react to Browser URL Warning messages.[71] extrapolates Protection Motivation theory on M-Turk respondents to find how classic and newer PMT factors influence user decision to malicious attacks. From these works, we can understand that researchers are often dependent on respondents on Amazon M-Turk to understand security and privacy related behaviours, as well as the efficiency of security tools. Following on this, [53] elaborated on the accuracy of this approach, by finding that M-Turk results often do not generalize as perfectly as previously expected, with the possibility of some bias always existing in these evaluations. In our work , we take these biases into account, and incorporate countermeasures against them while conducting our survey. The works stated above try to find user behaviour towards online threats. Our study is slightly different, in the way that we attempt to determine how misinformation propagated by URL Scanning Tools affect the respondents, and if it can possibly lead to them being exposed to malicious websites.

# Chapter 3

# Data Collection and Categorization

## 3.1 Data Collection

We collected one million tweets from Jan 2, 2020 to January 16, 2020 using the Twitter API [73]. Since we are investigating URLs posted on Twitter, we only collected tweets which had URLs embedded in them. To make sure we have a random and unbiased sample for our dataset, we used the Python Twitter API *Tweepy*[72], which provides a randomly sampled collection of tweets everyday. To further make sure we were only collecting URLs that were posted on the day of the collection itself, we modified our crawler script to discard tweets from any other days. To collect URLs that were posted on the day of the collection itself, we discarded tweets from any other days.

### 3.1.1 Removing URLs Linking to Tweets.

After collecting the tweets, we found that the majority of the them were links to embedded images or to other tweets. These links are signified with Twitter's own URL Shortening service, *t.co* [32] used to shorten links to tweets posted on their own platform. But, t.co also shortens URLs to third party links as well. Hence, to make sure we only remove t.co URLs which are links to tweets/images posted on Twitter only, we resolved all the URLs from the tweets using Python's *Request* module. The URLs were resolved such that we would collect the next URL in the redirection chain. This is done such that we can get the URL that users clicking on the link will actually visit. After resolving the URLs, we removed those which were links to other tweets or images/videos posted on Twitter and also duplicate occurrences of the same URL. After this filtering, *43,605* unique URLs were found.

Throughout the course of a month, we also repeatedly checked if the tweets containing the URLs do not come from accounts that are protected. We queried the accounts that the tweets in our dataset originated from, and if an account suddenly became protected, we removed the tweet posted from that account from our dataset.

### 3.1.2 Obtaining New URLs Using VirusTotal.

Everyday, the URLs after being collected and filtered, were immediately scanned using VirusTotal. We consider the day that we collected the URL as the *Day of Appearance* or *DoA* for that URL which means that the URL first appeared in our dataset on that particular day. We also filter out the duplicate URLs (and tweets), and the URL with the earliest date of posting is always preserved. Please note that, just because the URL appears on our dataset on a particular day, does not mean that it first appeared on Twitter or on the web on that day itself (i.e. if the URL is malicious it might not be a Zero-day threat). To negate this issue, we checked, for each URL, if they had already been scanned by VirusTotal. VirusTotal [80] being a very popular URL Scanning website, it is very likely that a URL that been previously scanned on VirusTotal, is not very new, and might have been encountered by a lot of users. On this basis, we removed 2,619 more URLs, bringing our final tally to 40,986 Unique URLs. However, even this approach cannot guarantee that the remaining URLs indeed appeared on the web during the time of collection only, and thus we do not use the word *Zero-day* to describe our URLs, instead choosing to address them as *DoA* term, but it can be said thatthey are more *likely* to be Zero-day threats.

## 3.2 Identifying Malicious URLs.

We used the VirusTotal Academic API to scan all the URLs on the day of their collection. Not considering the URLs that were removed due to them not being compatible with our DoA criteria, 1,472 URLs were flagged as malicious by VirusTotal. For a URL to be flagged as malicious, it has to be detected by at least one out of 80 different URL Scanning engines that are used by VirusTotal. According to [47], in some cases VirusTotal runs stripped down versions of certain URL Scanning engines and thus lags behind the label given by certain URL Scanning engines. It is recommended to cross-check VirusTotal labels with the individual URL Scanning vendors, as VirusTotal might say engine A identifies a certain URL as malicious, whereas A has changed its verdict to Non-malicious for that URL sometime ago. Thus, we checked each of these 1,472 URLs using the detected engine's website or tool to validate the VT report. In the case that the reading was not consistent and the engine had later reversed its decision from Malicious to Non-malicious, we removed the detection for that engine from the URL. In this way, 154 URLs having only 1 detection previously were removed from the set of Malicious URLs, with our final Malicious URL Set(as detected by VirusTotal) containing **1,318 URLs.**

### 3.2.1 Malicious URLs Detection Rates.

Of the 1,318 URLs that were detected by VirusTotal as malicious, 1,042 URLs had only 1 detection (out of 80), and 245 were detected by 2 engines. Only 32 URLs were detected by more than 3 engines, with the maximum being 6 engines (1 URL). The URLs were detected, on average by 1.93 engines. These detection statistics are for DoA for the respective URLs, i.e., when they first appeared in our dataset. Later on, in 6.2, we see that URLs in our dataset have much lower detection rates than other URLs which are present in the wild.

### 3.2.2 The Amount of Malicious URLs on Twitter.

We scanned the URLs as soon as we collected them, and we collected data over 15 days, thus leading to 15 separate scans. We found that, in our dataset, on an average, 3.3% (std= 0.78) of the URLs posted on Twitter were flagged as Malicious by VirusTotal on the everyday bases. The standard deviation indicates a trend that similarly lower percentage of URLs are detected as malicious over the whole duration of 15 days. This day to day detection of URLs has been illustrated in Figure 3.1.

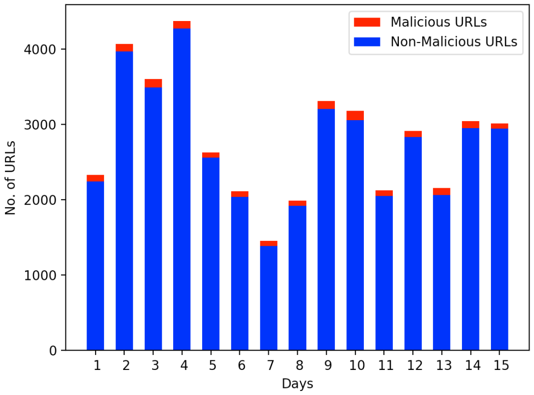


Figure 3.1: Number of malicious and non-malicious URLs appeared on our Twitter sample, from Jan 6 to Jan 20, 2020.

We do not have a conclusive statistic as to what portion of the internet is actually malicious in any given period of time. Some related works [28, 76] including a more recent study [1] suggest that an average of 1.4 million Phishing URLs are being created every month. Considering these estimated numbers, we find that 3.3% of the URLs in our database had been detected as Malicious, which is very low. This raises a few questions:

1. Since Only 3.21% (1,318 out of 40,986) URLs in our dataset were detected as malicious, are the protections provided on Twitter very effective at removing tweets containing malicious URLs?

2. With an average detection rate of 1.93 for the URLs in our dataset on DoA, what are the factors which lead to these URLs having such low detection rates? How this number compares

to regular Malicious URLs in the wild? For example, does Twitter remove URLs which have higher detection rates very quickly and so we were not able to collect them using our crawler?

### 3.2.3 Accounts posting Malicious Tweets

All the 1,332 URLs which had been detected by at least 1 VT engine had been posted over 2,638 tweets by a total of 924 unique accounts. Thus, each account posted an average of 2.85 tweets that have been detected as malicious in our dataset. But, we found 7 accounts which had posted 57 unique URLs encompassed over 93 tweets, raising their average rate to 13.28 tweets per account. Investigating these accounts we found that they frequently post malicious URLs and have been active for an average of 2.73 months. In [75], Twitter states that it may suspend accounts which post malicious links. However, in our case, we found that all 7 of the accounts were still active. We address this situation further in 4.3.1 where we take a more active approach towards measuring the factors which lead to suspension of accounts posting Malicious URLs.

## 3.3 Categorization of Malicious URLs

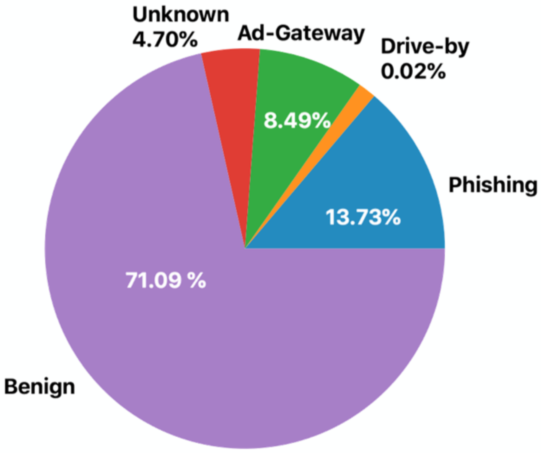


FIGURE 3.2: Categories of URLs that were detected by VirusTotal as malicious

For all the 1,318 unique URLs in our dataset that were flagged by VirusTotal, we labelled them by investigating each link manually on our test system, which was running the open source Chromium web browser. Throughout this experiment, we did not avail the use of Chromium's inbuilt Google Safe Browsing filter, and also turned off any other security tools on our system, so that our manual investigation was not hindered by warnings from these tools. Figure 3.2 provides an illustration of the breakdown of the categories of URLs that we found in our dataset(that were detected as malicious by at least one engine on VirusTotal)

Since malicious URLs tend to have a very short life span [3], we visited the websites as soon as they were collected by our crawler and detected by VirusTotal as malicious. Through our investigation, we categorized the URLs in to the following classes:

**Phishing Websites:**

Websites which try to imitate the websites of some legitimate organization and asking for user sensitive information, such as account passwords, social security numbers, payment information, etc. In our dataset we observed 181 URLs to be Phishing websites. Throughout the course of a month, Twitter removed tweets which contained 114 such URLs, with the majority of them (64) being removed within the 1st week itself. But, at the end of the month, 67 of them were still active on Twitter, distributed across 223 tweets in our dataset. Thus, 37% of the URLs that were manually classified by us as Phishing websites, were still active on Twitter. The ADR of the URLs in this category is 4.8 engines, i.e., a URL in this category has, on average a chance of being detected by 4.8 engines, with the URL with the highest detection rate has been detected by only 6 engines, while only 47 URLs having being detected by 3 or more engines. Indeed, this low detection rate has been consistent from the first day (ADR=2.07) and it only increased to and ADR of 4.8 over the course of a month. This low detection rate shows that many URL scanner engines have difficulty detecting such websites, and therefore we will further investigate these sparsely detected URLs in Section 5.

**Drive-by Download Websites:**

Websites which try to download malicious software on the user's system. For any website which automatically started downloading a file, we then proceeded to scan the file using VirusTotal to see if any antivirus engine detects them. In our manual inspection, to minimize the possibility of false positive, we filtered potential Drive-by Download URLs in the two ways: (i) cross checked the downloaded file with Malwarebazaar[40]), a popular database for hosting the latest hosting malicious samples, and the URL was only included if and only if the downloaded sample was found on the MalwareBazaar dataset. However, this is a conservative approach, since the dataset might not be comprehensive, therefore, (ii) we also checked any file that a Twitter URL downloaded and scanned them using VirusTotal, and if they had detection rate of more than 10, then we labelled them as Drive-by downloads.

We manually labeled only 19 URLs as Drive-by download websites. Twitter was able to remove 12 of such URLs over the course of the first day, and by the end of the month removed 4 more. Incidentally, only one URL in this category was detected by 7 different engines on VT, making it the most detected URL in our dataset. Drive-by downloads constitute a major portion of Malicious URLs out in the wild as well. This raises the question whether Twitter is efficient in blocking Drive-by downloads on sight, which we explore later on in Section 4.

**Malicious Gateways Websites:**

Websites which host advertisements or intrusive pop-ups, which contain elements of one or both of categories (i) and (ii), as well as automatically re-directing users to other websites, whether immediately upon visiting those websites, or by clicking anywhere on the website. These websites are often harmless in nature with respect to their content, but often confuse users by bombarding them with ad banners or popup windows which ultimately leads the users to malicious content.We labeled 112 URLs as Ad-Gateway, out of which Twitter only removed 23 over the course of a month. This is expected since most of the URLs in this category are harmless in nature, as far their own content is concerned, even though they often redirect users to other third party websites which distribute malicious content. Out of those, 62 websites were found to host e-commerce portals, whose malicious nature (if at all) is difficult to ascertain.

**Unknown:**

A few of the websites in our dataset consisted of e-commerce outlets which were detected by VirusTotal as malicious. They ask for sensitive information, such as credit card information, residential address, etc. with the promise of delivering some products/services. It is hard to evaluate whether the authors of these websites have any nefarious intention just by looking at the content of these websites. The only way to know if they are a legitimate business or not was to avail of their services and then see if the purchase obligation is met at a later date, which felt was an unnecessarily arduous and time-consuming given the scope of our work. These type of web attacks can be online scams (including tech support scams) whose attack vectors are mainly dependent on intricate social engineering tactics. While work in [37] demonstrates a strategy to detection of such websites, they are not in the immediate purview of this work Thus, and so we simply label these websites as *Unknown* and do not consider them to be Malicious for the rest of our study. These websites are not in the immediate purview of this work.

Table 3.1: Average Detection Rate (ADR) for URLs of different categories over the course of a month.

| URL Group | No. of URLs | ADR on VT | Active on Twitter after | | | |
|---|---|---|---|---|---|---|
| | | | Week 1 | Week 2 | Week 3 | Week 4 |
| Phishing | 181 (13.73%) | 4.83 | 117 | 81 | 72 | 67 |
| Drive-by Download | 19 (0.02%) | 7.25 | 12 | 7 | 5 | 3 |
| Malicious Ad-Gateways | 112 (8.49%) | 1.97 | 104 | 97 | 93 | 89 |
| Unknown | 62 (4.70%) | 1.09 | 54 | 48 | 43 | 39 |
| Benign | 937 (71.09%) | 2.09 | 919 | 908 | 895 | 872 |

**Benign Websites:**

Finally, we found the majority of the URLs (937) to be *Safe* websites, since they exhibited no malicious behaviour. Only 65 of them removed by Twitter after a month. However, interestingly, on VirusTotal the ADT was 2.09, which is higher than Malicious Ad-Gateways (1.97). Also, 246 of the URLs in this category had were detected by 3 or more engines, with the highest being 5 detections (for 2 URLs). In Section 6.1, we further investigate and characterize the URLs in this category to better understand the reasons of them being labeled as malicious by multiple URL scanner engines.

.

# Chapter 4

# Emulating Malicious Behaviour on Twitter

The URLs from our dataset that were determined as malicious on VirusTotal consistently low detection rates. Phishing is fairly predominant on Twitter [1]. Based on our manual examination of the URLs in previous Section 3.3, only 0.97% of the URLs in our entire dataset exhibited phishing behaviour. This is a low number, which can be due to: (i) *Twitter's pro-active blocking.* This is when the posted URL is blocked by Twitter immediately upon posting as the said URL exists on Twitter's blacklist; (ii) Multiple users report the tweet containing the URL on Twitter which leads to a manual investigation on Twitter. Our dataset contains a random sample of tweets that were collected using the Twitter API almost immediately upon posting, i.e., usually within a minute of the post appearing on Twitter. Therefore, there is a little chance of them being reported by multiple people in such a short amount of time. As the result, we actively tried to post malicious posts on Twitter and observed how Twitter respond to them.

## 4.1 Experiment Setup

### Twitter Account Setup.

We created an account on Twitter with the sole goal of carrying out this experiment. Since the majority of the tweets that we posted through this account contained URLs that are malicious in nature, we decided to make this account private. A private account on Twitter is one whose tweet can only be seen and interacted with by that accounts' followers and friends. Also, the owner of that account has to manually accept any follower requests. This made sure that no outside individual had accidental access to the tweets on this accounts. Since, we also wanted to check the visibility of the tweets we posted to other users, as well as if we could share (retweet) the tweets, we (the authors) followed this account using 10 other Twitter accounts created by us. Considering the ethics of this experiment, we contacted the IRB board of our Institute, who ensured us that this study does not need an IRB approval.

### Creating a Malicious Dataset.

We collected 300 URLs from three categories (phishing, drive-by download and malicious Ad Gateways): The phishing category consisted of 100 URLs, with 50 each from PhishTank and OpenPhish [46], another publicly available phishing blacklist. We only wanted URLs which are verified as phishing by their respective communities. OpenPhish posts URLs only after they have been verified by their own team, on the other hand, PhishTank [49] is open, and anyone can post URLs on their blacklist, which is then verified by other individuals registered on the website. We included URLs as phishing, only when they were verified as phishing by one or more of the top verifiers on the website, and were online at the time of posting on Twitter. The ADR for the URLs collected from PhishTank was 7.26 and for OpenPhish was higher at 9.02.

For drive-by downloads, we collected 100 verified URLs from URLHaus [77], which is a reputed blacklist for reporting URLs that distribute malicious downloads. The ADR for these URLs was 6.91. Finally, according to our knowledge, no publicly available URL blacklist maintains a database for Malicious Ad-Gateways, and thus we manually select a list of 100 URLs which consistently hosted malicious Ads. While there is no sure way of determining when these websites appeared frequently in the wild and what was the scope of their exposure, during the time of testing all of them still hosted one or more ad links which lead to websites which we identified as malicious.

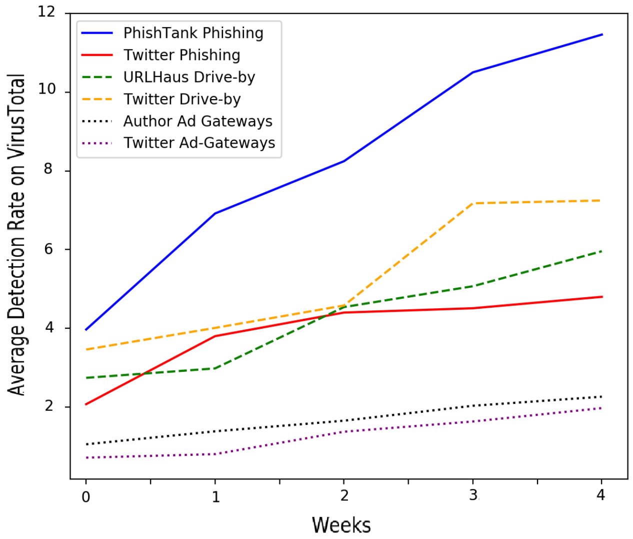
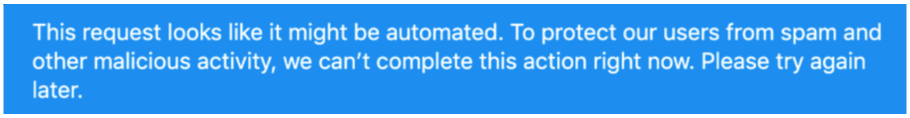


Figure 4.1: Comparison of ADR between categories of URLs collected from Twitter and random URLs found on public blacklists.

## 4.2 Twitter Response to Phishing URLs

Out of the 100 phishing URLs, Twitter prevented us from posting 41 of them immediately upon clicking on Tweet, by showing the message *"This request looks like it might be automated. To protect our users from spam and other malicious activity, we can't complete this action right now. Please try again later."* as shown in Fig 4.2. The remaining 59 URLs were successfully posted. However, upon clicking on 16 of the links, Twitter showed a warning message, informing that the website contains malicious content(Fig 4.3). The ADR for the 67 URLs (Twitter blocks and warnings combined) was 3.97 on the day they were collected. Later, over the course of a month, Twitter removed 24 more URLs, including the 16 which it had warned about on the first day. The remaining 43 URLs which could be posted and were not detected by Twitter, had a much lower ADR at 2.16. The ADR of all the 100 Phishing URLs (both detected and undetected by Twitter) rose to 11.47 after a month. In Table 3.1, we saw that by the end of a month, the ADR for Phishing URL rose to only 4.83. We compared the ADR between malicious URLs that we found in our Twitter dataset with random URLs that we collected from public blacklists for all the three categories and we found that the ADR of phishing URLs are significantly different between the ones collected from Twitter and the ones collected from public blacklists, with the former being much lower (4.8) than the later (11.47). This data has been graphically illustrated in Figure 4.1. The URLs of the two other categories show similar ADRs between the Twitter and random URLs. This warrants further investigation into these type of URLs which we do in Section 5. Therefore, we further investigated these URLs that remained undetected by Twitter.

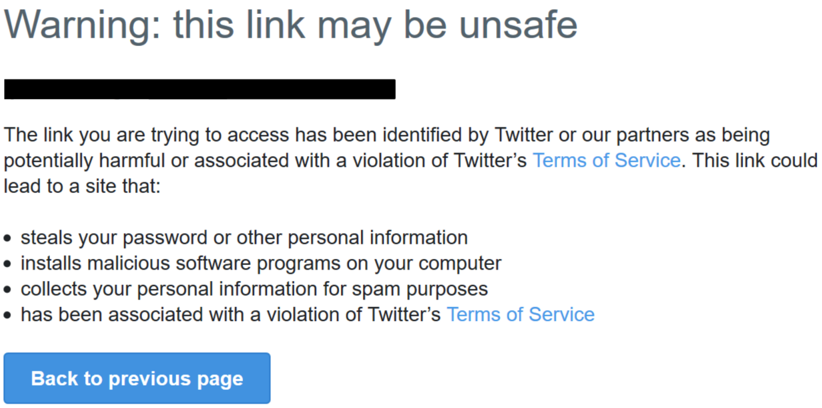


This request looks like it might be automated. To protect our users from spam and other malicious activity, we can't complete this action right now. Please try again later.

Figure 4.2: Message generated by Twitter when it prevented us from posting a URL

**Web form URLs can remain undetected.**

Out of the 100 phishing URLs, there were 18 web form URLs who maintained a combined ADR of only 0.78 during the initial collection time (within an hour of their first appearance on PhishTank, with 8 of them being detected by one engine only, and 7 of them have not been detected at all). When we attempted to post these URLs on Twitter, *none* of them were blocked immediately, and we could successfully post all of them. Note that we used the full URLs for Google forms (e.g., *docs.google.com/forms/...*).



Warning: this link may be unsafe

[redacted]

The link you are trying to access has been identified by Twitter or our partners as being potentially harmful or associated with a violation of Twitter's Terms of Service. This link could lead to a site that:

- steals your password or other personal information
- installs malicious software programs on your computer
- collects your personal information for spam purposes
- has been associated with a violation of Twitter's Terms of Service

Figure 4.3: Warning generated by Twitter when it suspected that the user was trying to visit an unsafe website

We then monitored the tweets over a month, and surprisingly Twitter did not remove or block access to any of them. However, the ADR of these URLs increased to 2.39 at the end of the month with prominent URL scanning engines, such as Kaspersky, Bitdefender, ESET and Fortinet acknowledging them as phishing. However, in Section 6, we explore a more plausible reason for these URLs getting detected by URL scanners and blacklists, where we find a correlation between their detection, and them being posted on PhishTank. Also, Google diligently removes these forms as long as the potentially malicious content/questions, such as asking for password, credit card number, social security number, was in plain-text format. 4 of the URLs in this dataset used certain evasion tactics (discussed in Section 5), and none of them were removed by Google, with only 2 of them were detected by one engine.

### Mislabeling of Shortening Form URLs.

When we used shortened links for Google forms (e.g., *forms.gle/...*), Twitter initially allowed us to post all 18 links without any interference. However, when we clicked on these links, Twitter warned us that these links might be unsafe to visit.

Then, we experimented with using 5 more shortened form URLs on a benign web form. Twitter allowed posting all 5 of these URLs, but again warned us that all the links can be unsafe. Furthermore, we used 5 more links which had started with forms.gle, but this time these URLs did not exist in reality. An example for this is *forms.gle/this-is-not-a-real-link*. Twitter again warned that all 5 of these links can be unsafe.These experiments show that Twitter has a a tendency of labelling Google Form URLs, which are shortened using URL shortening services, irrespective of them being Malicious or not, or even the URL existing in the first place.

## 4.3 Twitter Response to Other Malicious URLs

**Drive-by Downloads.**

Out of the 100 URLs containing drive-by downloads, Twitter blocked 37 by preventing from posting them, while we were warned against visiting 8 of them. The ADR for the 45 detected URLs was 6.16, while the 55 undetected URLs had an ADR of 5.8. Twitter removed 17 more URLs (including the 8 it had warned us about on the first day), over the course of a month.

**Ad-Gateway URLs.**

Out of 100 Ad-Gateway URLs, Twitter was able to block 55 of them, while also giving a warning for 15 URLs. The ADR for the detected URLs were 2.25, while the 30 undetected URLs had an ADR of 2.3.

Since many of these malicious URLs do not show up on many URL blacklists, it might be the case that Twitter removes those URLs that are reported by users, and therefore others remain undetected. This is consistent with our findings in Section 4.4, where we found Twitter is efficient in removing URLs which are consistently reported by it's users.

### 4.3.1 Twitter's URL Removal Strategies

In summary, in the first day of posting these URLs, Twitter blocked 133 out of our 300 URL across the three categories outright, preventing from posting them, 167 URLs still were still posted on the account, all of which were malicious in nature. Over the course of a month, we also observed the number of removed tweets from our account. Ideally, these tweets should disappear. However, all 167 of the tweets were available to view from our account from which we posted the tweets in the first place. We also tried to view these tweets using the other 10 accounts that follow our account, and we could view all but 63 of them. On further investigation we found that Twitter *hides* the tweets which it detected later on, and never removes them. We argue that this is a suitable measure, since technically any user other than the poster would not be able to see these URLs. However, it is notable that even though our account had posted many tweets with malicious URLs, Twitter *had not* suspended the account.

## 4.4 User Reports and Account Suspension

Twitter provides an option to users to report tweets due to various reasons, with one of them being *Suspicious or Spam*, under which users have the option of choosing *Includes a link to a potentially* *harmful, malicious or phishing site.* We refer to this later option as *Reporting Choice* or *RC* from here on after.

**Experiment Setup.**

We created 3 more private accounts (*Malicious Posters or MP*), with each account being followed by 10 other accounts(Followers). We then posted all 167 tweets (which we refer to as *MSNB* or *Malicious Set of URLs not blocked by Twitter*, sporadically over a period of two weeks on each of these accounts. We used a randomizer which chooses $n$ number of tweets everyday for each account, from the MSNB and posts them using that MP account. The randomizer keeps track of previously posted tweets for a MP account, so that all 167 tweets can be posted for these accounts. While the process of posting is being done, another Randomizer program runs for each follower account, which randomly picks $x$ number of tweets posted by the MP accounts for a particular day, and reports them to Twitter by selecting the *Reporting Choice*.

**Analysis.**

Twitter suggests that it uses the reporting mechanism to identify and suspend offensive accounts [5]. We also observe the same behavior, where after a period of two weeks, all three of our MP accounts had been suspended on grounds of violating the usage clauses of Twitter [70]. All the MP accounts were suspended before they can post all 167 tweets. More specifically, two of the MP accounts were suspended within the first 5 days, whereas the sole remaining account took 8 days to be suspended. We define *Tweet Report Rate (TRR)* as the cumulative total of tweets reported by the Follower accounts divided by the cumulative total of the tweets posted by an MP till a particular day. Using this metric, we track how reporting the tweets leads to Twitter suspending the account. TRP has a maximum value of 1, which means all posted tweets have been reported.

In Fig 4.4, we see the system design of the procedure which is followed by the MP Accounts and the Followers that work in this experiment.

Figure 4.5 shows the increasing rate at which Followers reported the tweets for all 3 MP accounts, and the Anchor points (the terminating circles of the line in the figure) indicate at what duration they were blocked. MP accounts 1 and 3 show similar patterns of reports against their tweets when they were suspended, while MP account 2 lasted a little while longer. But all the accounts were blocked when the TRP rate increased steadily and were close to 1, suggesting that Twitter suspends an account when a large percentage of the tweets posted by that account have been reported.

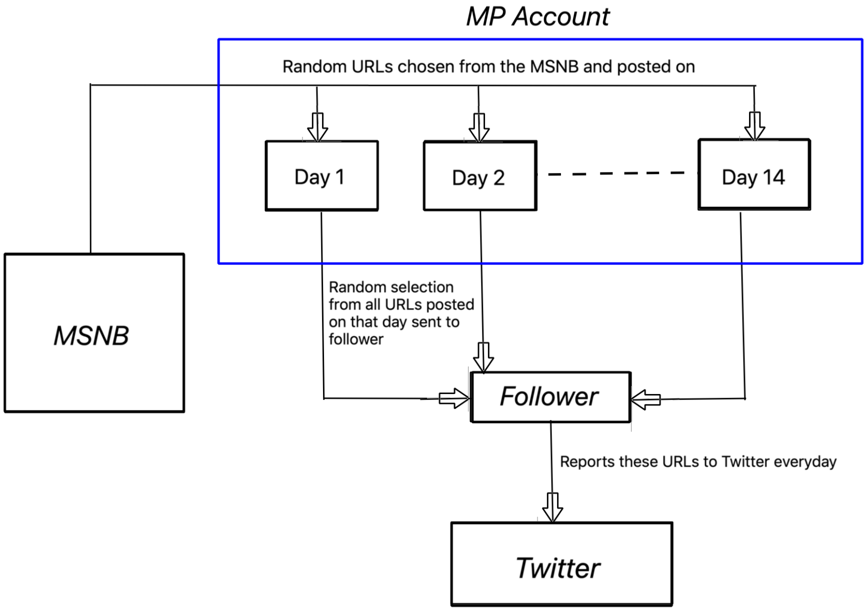

MP Account
Random URLs chosen from the MSNB and posted on
Day 1
Day 2
Day 14
MSNB
Random selection from all URLs posted on that day sent to follower
Follower
Reports these URLs to Twitter everyday
Twitter

FIGURE 4.4: System design for simulating Twitter account suspension using MP and Follower accounts

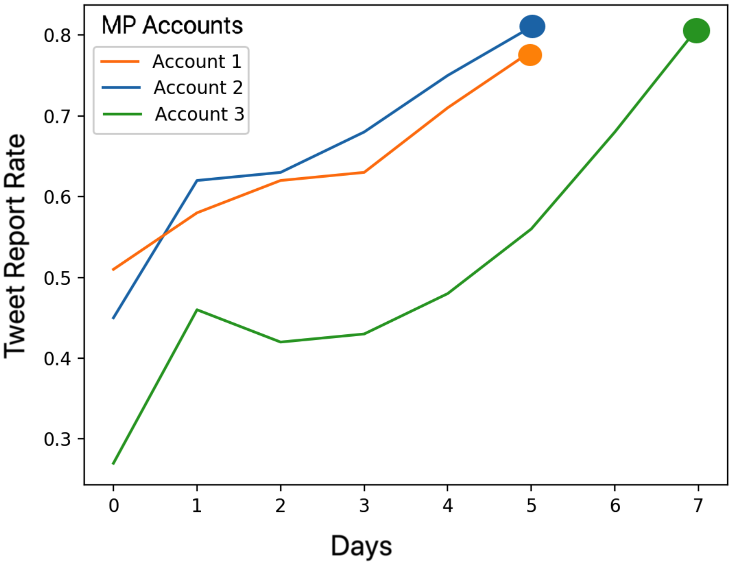


FIGURE 4.5: Tweet Removal Rate (TRP) for the three accounts.

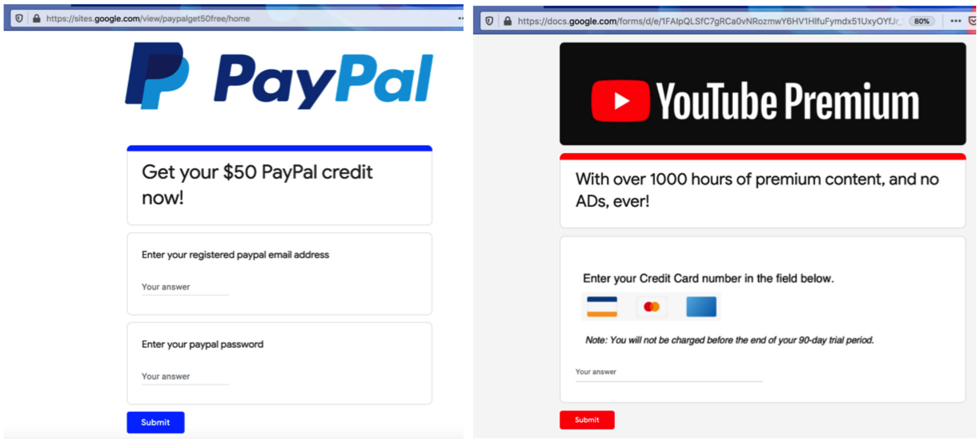
**Results.**

While Twitter suspend users based on user reports, we argue that the malicious content posted by these accounts should be removed before their followers are exposed to them, because reporting tweets is a manual process, and might take a long time, which potentially puts users at risk before the malicious account is suspended. Also, while we cannot conclusively state that Twitter will exhibit the same behaviour towards more popular accounts, these results are consistent with our findings in Section 4.3.

# Chapter 5

# Evasion Tactics used by Form URLs

From our URL set, out of 181 URLs which were determined to be Phishing, 59 URLs were Webforms hosted on Google. The design of these form of phishing websites is simple and they are easy to implement. Attackers create a web form on Google Forms or Google Sites and write the Form prompts/fields, such that users are asked to share personal information, such as email addresses, passwords, credit card Numbers, social security numbers, etc. Usually these web forms pretend to be originating from a popular commercial brand or product in the attempt to fool unsuspecting users visiting them. Fig 5.1 shows two examples of Phishing attacks which are hosted as Web forms on Google Sites and Google Forms.



https://sites.google.com/view/paypalget50free/home

Get your $50 PayPal credit now!

Enter your registered paypal email address

Your answer

Enter your paypal password

Your answer

https://docs.google.com/forms/d/e/1FAIpQLSfC7gRCa0vNRozmwY6HV1HlfuFymdx51UxyOYfJr

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 90-day trial period.*

Your answer

Figure 5.1: Example of common Phishing Web Forms

We tried visiting each of these URLs, and found 43 of them still alive, while the others showed a *"We're sorry. You can't access this item because it is in violation of our Terms of Service,"* suggesting Google has already removed them. Considering their respective *Day of Appearance* in our dataset, they were removed on an average of 2.19 hours after being posted on Twitter. This indicates that these forms of phishing attacks are removed quickly by Google. We assume that this is due to the presence of suspicious text, such as *'password'* or *'credit card number'* which trigger an automatic removal response. From the remaining 43 URLs, 14 of them had simple text content and Google removed them within the next 2 days.

## 5.1 Use of Images for Obfuscation

In our dataset, 29 web forms still continued to remain online even after a month. On manual inspection, we found that instead of using Textual features, 17 of these URLs had resorted to using images which showed the text. For example, instead of asking for *Password*, the field contained an image which had *Password* written inside it. Figure 5.2 shows one such example which imitates a popular telecom operator. The images closely resemble the background color of the form, making them virtually unnoticeable to the casual observer. These URLs were detected by only 5 unique engines on VirusTotal after a month, and the full URLs were never blocked or removed by Twitter.

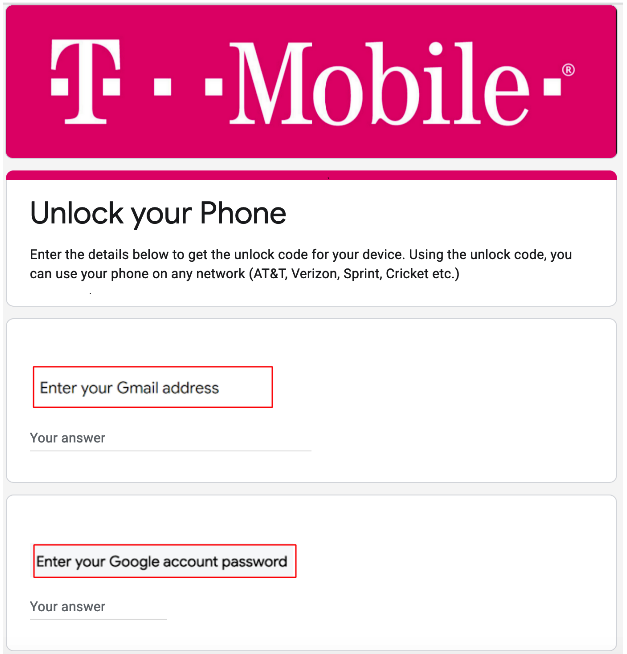
Since our sample size is small (with 17 URLs), we decided to collect 40 more such Google Form URLs (all of which were more than a week old during collection) from PhishTank, consisting of 20 each from the two categories of: *Simple Text*, and *Image-field*. Only 4 of these 20 PhishTank URLs were removed from Google even after a week of their first appearance on PhishTank, whereas for the 20 URLs which used simple text based features, 17 had already been removed by Google when we collected them. Therefore, this obfuscation method is successful in avoiding removal by Google's filtering systems.

**Possible Mitigation Strategies.**

Considering that Google develops TensorFlow [68] it is possible for them to analyze text within image content with the the help of Optical Character Recognition (OCR), which they should implement in their Google Forms and Google Sites ecosystem to mitigate the image based obfuscation strategy.

## 5.2 Use of Special Characters for Obfuscation

In our dataset, 12 Web form URLs used special characters in place of certain letters in the text. The URLs used the special character '*', '$', '@' inserted between the letters which constituted the word 'Password', 'Social Security Number', 'Credit Card Number,' etc. Thus, we collected 20 more of such URLs from Phishtank. Only 2 of them had been removed by Google, even when all of them were older than a week. We found several words such as 'Pa**word', 'Cred!t C@rd No.' and 'S$N'. Figure 5.3 provides one such example where special characters have been used instead of letters. While this form of obfuscation appears to be successful in avoiding detection,



Unlock your Phone

Enter the details below to get the unlock code for your device. Using the unlock code, you can use your phone on any network (AT&T, Verizon, Sprint, Cricket etc.)

Enter your Gmail address

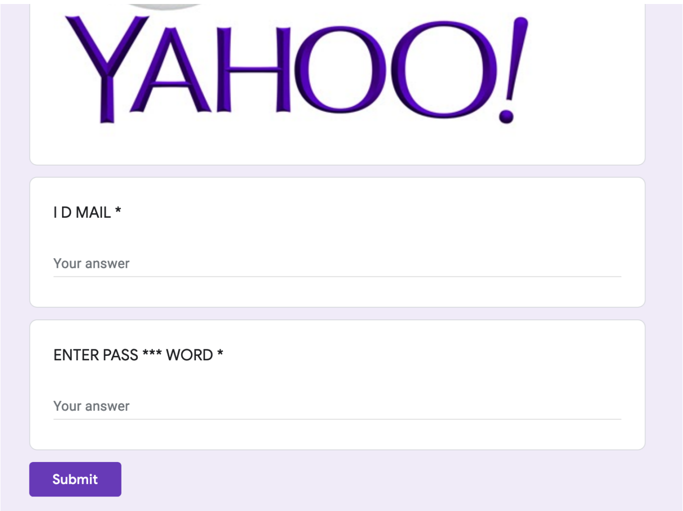
Your answer

Enter your Google account password

Your answer

Figure 5.2: Example of a phishing Web-form, which uses images for field text instead of text to avoid removal. Image content is highlighted in Red.

we argue that it is less dangerous than the image obfuscation technique, because there is a higher probability of users being suspicious of the phishing form if they see text is interjected with these special characters.



I D MAIL *

Your answer

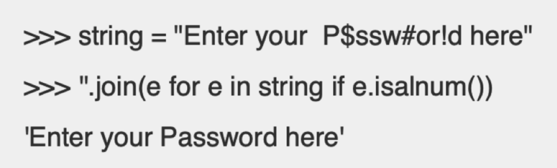
ENTER PASS *** WORD *

Your answer

Figure 5.3: Example of a phishing web-form which uses special characters in between text fields to avoid removal.

**Possible Mitigation Strategies.**

The simplest mitigation strategy that can be proposed for this obfuscation method is filtering out the special characters which appear to hinder the judgement of the text detection tools used by Google to detect these web forms. Regular expressions can accomplish this task by stripping the form text in such a way that only the plain-text remains. Fig 5.4 shows one such simple code snippet written in Python which takes the obfuscated text and generates a plain text output which clearly indicates that the fraudulent form asks for your password. This output will then make it easier for the web-form hosting service to detect and remove these phishing attacks.



```
>>> string = "Enter your  P$ssw#or!d here"
>>> ".join(e for e in string if e.isalnum())
'Enter your Password here'
```

Figure 5.4: A Python code snippet which can filter out special characters from a text and provide the plaintext as output.

Table 5.1: The number of Phishing Form URLs being removed by Google, and Twitter, where ADR shows the average detection rate on VirusTotal.

| Approach | Total | Google | Twitter | ADR |
|---|---|---|---|---|
| Text | 20 | 17 | 6 | 1.68 |
| Image | 20 | 4 | 0 | 1.09 |
| SC | 20 | 2 | 0 | 1.21 |

**Summary.**

In Table 5.1, we show the removal/blocking statistics for the Form Phishing URLs of all the three categories of fields i.e. text, image and special characters, across Google, Twitter and VirusTotal, after a week of them being present on the PhishTank database. We found that Twitter was only able to block 6 URLs for the simplest Form phishing websites (Text), while missing all URLs from the two obfuscation techniques.

# Chapter 6

# Dependence on Public Blacklists

During our manual inspection of URLs that were detected as malicious by VirusTotal (in Section 3.3), we found that 932 *benign* URLs were detected as malicious by VirusTotal. While these URLs might be false positives, they were still detected by VirusTotal by more than one scan engine, and 65 URLs were removed by Twitter over the course of a month. In this section, we investigate these *Benign* URLs by cross-examining them on public blacklists, such as PhishTank.

## 6.1 Cross-Examination of Benign URLs on Public Blacklists

Many URL blacklists are crowd-sourced and allow anyone to register on their websites and post links to their blacklists. As the result, it is possible for anyone, including malicious actors, to submit legitimate websites to these services, and have 'un-verified' URLs still show up on their blacklists. Since several URL Scanners often use public blacklists, including PhishTank [27], our goal is to determine whether there is a co-relation between benign URLs appearing on PhishTank and them being detected by URL Scanners, i.e., can URLs on PhishTank lead to false postives in URL Scanners.

### Verifiers on PhishTank.

Certain PhishTank users known as 'verifiers' evaluate all URLs posted on the website as either 'valid phishes' or 'invalid' (i.e., not phishing). PhishTank maintains separate lists for both valid and invalid phishing websites, and labels a URL with a specific label only if enough verifiers vote for that label. We searched for our remaining 872 *Benign* URLs in the *invalid* list that had not been removed by Twitter after a month. For having more confidence in the labels, we only considered the ratings of the top 10 verifiers on the website who have a combined total of 11.4 million votes.

### Presence of Benign URLs in PhishTank Invalid List.

We found 287 URLs on the PhishTank *invalid* list. Thus, almost 33% of the Benign URLs has been submitted at-least once on PhishTank and verified as Invalid(Benign). Yet they were still being detected by URL Scanners as Malicious even a month after their first appearance in our dataset. PhishTank does not provide any metadata about these invalid URLs in a developer friendly format. Therefore, we had to implement a web crawler based on the Selenium engine to browse through the invalid lists and collect the URLs. The URLs are listed on PhishTank from the latest submission first and so on through several pages. Due to the massive volume of URLs that reside on the website (estimated to be in millions based on the total number of URLs rated by the top verifiers), it was not possible to collect all of them. Thus, our aim was to be able to collect URLs which were posted at least a week before when we first started collecting our data (Jan 2nd), and started from Jan 21st 2020 and went back till Dec 24th 2019, collecting 1,872 URLs in the process. Thus, the number we have reported (287 Twitter URLs found in the invalid list), might have been larger, if we had been able to obtain all the PhishTank invalid URLs.

**Cross-Examining Random Invalid Phishtank URLs on VirusTotal.**

To further investigate if URL scannners use information from popular blacklists, we cross-examined a sample of invalid Phishtank URLs on VirusTotal. Through 29th January to 2nd Febraury 2020, we collected 100 random URLs, which were verified as *invalid* by one or more of the top 10 verifiers on the website. We then manually investigated them and identified 8 of them as malicious website, while we could not make a decision for 4 URLs. To be on the safer side, we eliminated all of these 12 URLs from further consideration. The remaining 88 URLs were then scanned using VirusTotal, and out of them, 64 URLs had at least one detection, with 229 unique detection instances were observed between 21 different URL scanning engines. VirusTotal assigns two labels for a detection by an engine: Malicious or Phishing. We find several prominent engines detect the *Benign* URLs, which have already been labeled as *invalid* on PhishTank. We also include the breakdown of the detection rates into phishing and malicious, since it may be argued that PhishTank only hosts phishing URLs, which is not the case according to the URL scanners on VirusTotal. During our manual analysis we find that URLs submitted on PhishTank consist of URLs similar in functionality to all the 5 categories we have found in Section 3.3.

Table 6.1 provides the distribution of the 10 engines with the highest number of detections. We found that CleanMX, which is a public URL blacklist, detects 49 out of 100 *Benign* URLs as malicious. Interestingly, CleanMX is also the top URL submitter on PhishTank with 2.9 million submissions to date. Of the 88 *Benign* URLs, CleanMX had submitted 21, all of which obviously detected as malicious URLs by VirusTotal. However, out of the remaining 67 URLs, CleanMX labels 28 of them as malicious. This hints that CleanMX might be using PhishTank submissions to boost their own system. We further investigate this in Section 6.2, where we check if posting a

Table 6.1: Distribution of phishing and malicious detection rates by 10 engines with the most detections for the *Benign* URLs flagged as *invalid* by PhishTank. (Total URLs=100).

| **Engine** | **Malicious** | **Phishing** | **Total** |
|---|---|---|---|
| CleanMX | 27 | 22 | 49 |
| ESET | 19 | 13 | 32 |
| CyRadar | 12 | 17 | 29 |
| CRDF | 6 | 9 | 15 |
| Fortinet | 4 | 8 | 12 |
| BitDefender | 4 | 6 | 10 |
| Sophos | 2 | 7 | 9 |
| Comodo | 3 | 4 | 7 |
| G-Data | 5 | 2 | 7 |
| Kaspersky | 3 | 3 | 6 |

set of safe and new websites get detected right after being posted on PhishTank before they are even verified.

Trailing CleanMX is ESET, a popular commercial antivirus vendor, with misdetection rate of 32 out of 100 *Benign* URLs. ESET has been certified as an efficient URL Scanning engine by AVComparatives [10], one of the most reputed independent organizations which tests the efficacy of security products. However, ESET has a high false positive rate towards *Benign URLs.* Similarly, all the other listed vendors in Table 6.1, including Fortinet, BitDefender, Comodo, G-Data and Kaspersky, are highly acclaimed vendors, and are still susceptible to mislabeling *Benign* URLs.

**The Case of Trusted Domains.**

Of the 24 URLs which were not detected by any engine on VirusTotal, we found that 9 of them were webpages hosted under popular vendors - Google, Yahoo, Apple and Bank of America. For the remaining URLs, we checked their Page Rank on Alexa, and all of them where included in the Top 500 of Alexa's list. This vaguely suggests that most URL Scanners have a white-list of the most popular websites which they do not detect, despite being posted on PhishTank.

## 6.2 Evaluating Benign and New URLs on Phishtank

We found that a huge majority of *Benign* URLs posted on PhishTank and also marked as Benign (*invalid*) were detected as malicious, by as many as 21 unique URL Scanners. To investigate the relationship between URL posts on PhishTank and them being detected as malcious by scan engines, we implemented a new experiment.

**Creating a Dataset.**

We collected 20 different URLs from various sources which we investigated as being benign. We made sure that these URLs are not hosted on those Web hosting services which are frequently mislabeled by URL scanning engines. Also, we did not use any URL Shortener tools. The URLs consisted of 7 blogs, 4 ecommerce websites, 7 news websites, and 2 banks originating outside the US. Out of the 20 URLs, 5 of them were present in Alexa top 500 [69].

**Experiment Setup.**

We initially scanned all the websites on VirusTotal and found them to be clean (no detections). Outside of the 5 websites included in Alexa's Top 500 List, the other 15 URLs are not very popular, and might not have the volume of traffic that are expected from popular websites. Sahingoz et al. [47] found that URL scanning vendors frequently crawl the web for discovering harmful websites. Also, when the users of their products visit an unknown URL, it automatically triggers an investigation for that URL. While we cannot control how URL scanning vendors randomly crawl through the Internet to find new websites, we can design our experiment in such a way that a few users visit these websites frequently while they have the tools from these vendors installed. We accomplished this by implementing 10 virtual machines across 3 different systems. All of these virtual machine systems run a copy of Windows 10, and we installed 5 of the most popular commercially available antivirus tools that we found in Table 6.1, i.e., ESET, Fortinet, BitDefender, Sophos and Comodo. Next, we implemented an automated script using the Selenium engine, which visited all the 20 websites frequently everyday on each of these VMs. Note that the Web browser protection/filter was turned on for each of the 5 products, and we had opted to share anonymous data to improve detection (An option provided by AntiVirus vendors in their user-interface). Thus we have every reason to believe that the websites were indeed being investigated by the vendors, due to their multiple visits everyday. We ran this setup for a week. Afterwards we ran another scan over the VirusTotal for the URLs, and it found no detections.

**Uploading URLs on PhishTank.**

The next part of our experiment was to upload these URLs on PhishTank's database. Immediately after uploading the URL, PhishTank slots them to 'Recent submissions,' i.e., submissions which have not been rated by the community yet. We waited for a period of 2 hours, before scanning all 20 URLs on VirusTotal again. During this time, the posted URLs had not been validated by any verifier on PhishTank. 9 of the 20 URLs were detected with at least one detection. CleanMX detected 7 of these URLs, while ESET proceeded to detect 5 of them. The other engines which participated in the detection were Comodo (4), BitDefender (2) and Fortinet (1). Incidentally, out of the 5 URLs that are included in Alexa's 500 list, only one of them was detected by CleanMX, while the rest had no detections. To verify VirusTotal's readings, we further visited the detected websites using the VM systems which had ESET, BitDefender, Fortinet and Comodo installed, and indeed, the URLs were blocked by those vendors now as well. This experiment shows that the URL scanner engines are pro-active and even before URLs being verified by the reviewers, or even after them being invalidated by the reviewers, still detect *Benign* URLs as malicious. This strategy increases their false positive rates, and more importantly provides an opportunity for the attackers, who can add legitimate URLs to the popular blacklists, with the goal of negatively affecting those websites.

## 6.3 Posting Obfuscated Phishing URLs on PhishTank

In Section 5, we found a new type of phishing attack, which leverages the trustworthiness of Google's domains, and posts web-forms which ask for private information from the users. We found that these URLs are very rarely detected by URL Scanners, and protection against them is reliant on how soon Google investigates and removes them. However, since these websites frequently use evasion tactics that we found in Chapter 5,they can avoid detection by Google as well. This leaves these types of phishing attacks available on the web for more than a month at a time. We also found the partial dependence of prominent URL scanners on PhishTank, which leads to misdetection of Benign websites. In this section, we explore the opposite end of the spectrum, i.e., if posting this form of undetected attack on PhishTank can help the URL scanners in detecting them.

**Experiment Setup.**

For this experiment, we created 10 Web forms, and posted each of them on Google Forms. Each form had two different URLs, with the intention being to post one of them on PhishTank, while the other is not. We scanned the URLs beforehand on VirusTotal, which detected nothing. We then proceed with our VM setup and visited the URLs for a week. Since all of these Forms were created with the obfuscation methods we have found, they had not been removed after a week, and VirusTotal still had not detected them.

### Uploading on PhishTank

We uploaded these URLs on PhishTank, and scanned them again after 2 hours. This time, 14 of the URLs were detected by 8 different engines, including CleanMX, CRDF, BitDefender, ESET, Comodo and Fortinet. On the other hand, the other part of the pairs of the Forms which had not been posted on PhishTank still returned zero detection on PhishTank. *Thus, this shows that the URLs were detected only because of them being posted on PhishTank, and not due to them being malicious.*

From these observations, we draw a new possible Attack technique by which attackers can pollute a benign URL's verdict given by the Anti-Phishing Tools.

### Ethics.

The phishing URLs that we created on Google Forms for this experiment contained fields which asked for personal information from the visitor. However, we did not collect any data that had been entered in these forms. After completing our experiment, we deleted these forms, and with them, the data that was posted in them. Also, it is not possible to access the Forms accidentally, since they can only be accessed using a unique link that is not indexed by any search engine. Also, for the Form URLs that were posted on PhishTank, we discarded any user data that the verifiers had entered immediately after they had checked the URLs.

### Possible Attack Strategy on Benign URLs

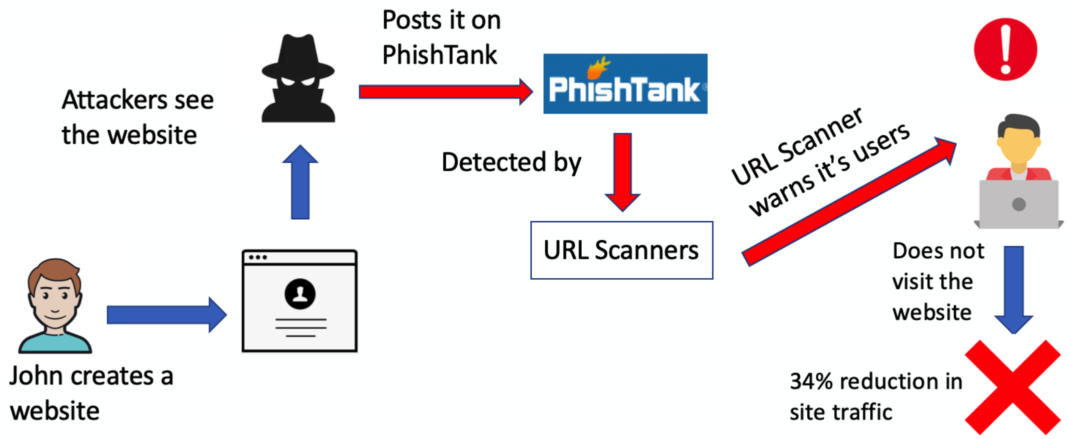

Posts it on PhishTank
Attackers see the website
PhishTank
Detected by
URL Scanner warns it's users
URL Scanners
Does not visit the website
John creates a website
34% reduction in site traffic

FIGURE 6.1: A scenario where an attacker can leverage URL Scanner's dependence on PhishTank and poison the verdict of a Benign website

Figure 6.1 shows a scenario where attackers can leverage the reliance of these Anti-Phishing Tools on PhishTank, and use it to make them misdetect benign URLs.

John creates a website which is used as an online store front for his e-commerce business. This website is benign. Now, an attacker wants to make sure that John gets lesser visitors than what he expected. Since any individual can post URLs on PhishTank's database, the attackers submits the URL for John's website on PhishTank. Soon after, several Anti-Phishing tools detect John's website as Malicious, and for the users who are using these Anti-Phishing Tools, if they try to visit John's website, they are warned that the website might be unsafe. If the users trust this rating, then they do not visit this website. In 7.2 we see that these misinformed ratings(in this case False Positives) lead to a 34% reduction in website traffic. Thus 34 out of every 100 potential customer would not visit John's website, which would ultimately harm his e-commerce business. This example is one of several ways that attackers can use take the advantage of the reliance of Anti-Phishing tools on public blacklists, and make them flag a Benign URL as malicious. The analysis of the scope of this attack, and how well-spread it is on the web is something we leave for future study.

# Chapter 7

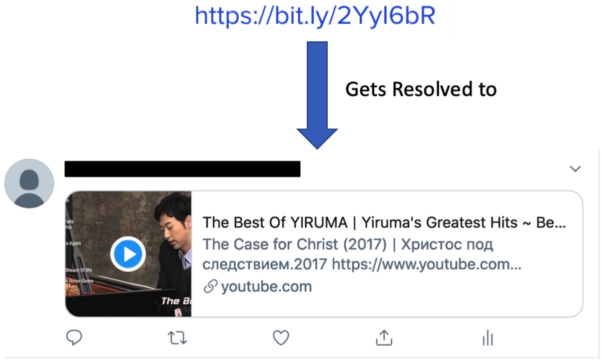
# The Case of Benign URLs

## 7.1 Misinformation Factors

In our manual inspection of URLs being detected by VirusTotal, we found 71% of the URLs in our dataset which did not exhibit any malicious behaviour, yet 65 of them were detected and removed by Twitter within a month. Checking our dataset of the 937 URLs we had labeled as *Benign*, we found that 29 out of these 65 URLs were using URL shortening services, and 15 were using a free web-hosting domain. Thus, we expanded our investigation for all the URLs in the *Benign* set and found 131 URLs which were using a URL shortener, while 112 of them were using free webhosting domains. The average detection rate for both sets was very low, with SUs having an ADR of 1.76, whereas for WHD URLs the ADR was 1.53. The SUs were detected by 11 unique URL Scanners and blacklists, out of which Comodo(4),Fortinet(3) ESET(3), Fortinet(2) were popular Antivirus solutions running on consumer and enterprise systems. Considering the fact that users frequently use both URL Shorteners [39], Web Hosting services[2], such benign URLs being detected by URL Scanners warrants a closer look into this situation.

### 7.1.1 URL Shortening Links.

We found that the 131 URLs were shortened by 7 different URL shortening services, including *bitly.com*, *bit.ly*, *rebrand.ly*, *ow.ly*, *us.to*, *chng.it*, and *shorl.com.* Bit.ly and Bitly.com had detected the most URLs combined (47), followed by *rebrand.ly* (33). The detection statistics are better illustrated in Table 7.1. We also found that Twitter 'warns' against opening links from 4 of these URL providers However, this warning is not consistent, i.e., it does not warn against every URL which uses one of these URL shorteners. Out of 131 such URLs, Twitter showed a warning for only 35 URLs. While we could not find any strong factor towards Twitter warning against them, we found 31 of the warned URLs to have very long URL strings, with all of them being larger than 192 characters. Since social networking websites such as Twitter attempt to resolve all Shortened Links to display a preview(An example is shown in Fig 7.1), it is possible that it could not resolve these 31 URLs, and thus warned us against them. Proving this assumption, would however, require a large scaled analysis to find if Twitter really mislabels Shortened URLs which we leave again,



[redacted]

The Best Of YIRUMA | Yiruma's Greatest Hits ~ Be...
The Case for Christ (2017) | Христос под следствием.2017 https://www.youtube.com...
youtube.com

FIGURE 7.1: An example of Twitter resolving a Shortened URL and generating a Link preview when it is posted.

TABLE 7.1: URL Shorteners - DOT : Detected on Twitter WOT: Warned on Twitter. DR: Detection rate on VirusTotal.

| **Host** | **URLs** | **DOT** | **WOT** | DR |
|---|---|---|---|---|
| bit.ly | 29 | 0 | 0 | 2 |
| bitly.com | 18 | 9 | 12 | 1 |
| rebrand.ly | 33 | 17 | 8 | 2 |
| ow.ly | 29 | 7 | 3 | 2 |
| us.to | 10 | 2 | 3 | 2 |
| chng.it | 7 | 0 | 0 | 3 |
| shorl.com | 5 | 0 | 0 | 2 |

for future persual. Since attackers often use URL shortening services to mask malicious links [38], It is possible that URL Scanning vendors and Twitter choose to blacklist some of them. However, they might also block legitimate URLs, which have been shortened by such services.

### 7.1.2 Web Hosting Domains.

We found 112 URLs being hosted across 5 free web hosting domains, including *000webhost*, *Wix*, *hpage*, *freehostia*, and *atspace*. It is easy to determine which websites are hosted using these services since any website hosted by them has the URL format *sitename.freeservername.com* or equivalent. One provider - *000webhost* housed 56 of these URLs, suggesting that it is a popular service for users looking to host a website for free. The detection statistics for the detected domains are shown in Table 7.2. The average detection rate for these URLs was 1.53. These web hosting services allow users to build a website free of cost, and attackers frequently use these services for hosting phishing and other malicious websites. Therefore, it is plausible that URL scanners have blacklisted URLs of these domains. But the harm done by mislabeling these domains is much lower than that of URL Shorteners, since any user is looking to build a proper website would avail the use of a paid hosting service with a custom domain. On the other hand, Twitter did not warn us against any of these URLs suggesting that Twitter is not prone to mislabeling URLs which are hosted on free web hosting services.

Table 7.2: Web Hosting Domains Table - DOT : Detected on Twitter WOT: Warned on Twitter. DR: Detection rate on VirusTotal.

| **Host** | **URLs** | **DOT** | **WOT** | **DR** |
|---|---|---|---|---|
| 000webhost | 56 | 0 | 14 | 2 |
| Wix | 25 | 0 | 4 | 1 |
| hpage | 20 | 0 | 7 | 1 |
| freehostia | 4 | 0 | 0 | 2 |
| atspace | 7 | 0 | 0 | 1 |

## 7.2 Impact of Anti-Phishing Warnings on Benign websites

Previously, we found that Twitter mislabeled several legitimate URLs by warning users, when they try to visit them. This occurs due to 2 different factors, URL Shortener Bias and Webhosting domain bias. We found 7 popular URL Shorteners, and 5 Web Hosting domains which suffer from such mislabeling. Also, from we found 27 Benign URLs in our Benign, which Twitter warns against, but have been invalided on PhishTank by top verifiers. In this section, we empirically examine the impact of these warnings on web traffic of these benign URLs through Twitter, i.e., if seeing the warnings makes users visit the websites associated with the URLs more or less.

### 7.2.1 Account Setup.

We created three Twitter accounts, (which we refer to as *Impact account* or *IA*), with the sole intention of growing a follower base of real users, who will then be able access the benign websites that trigger Twitter's warnings. To attract followers for our account, in an approach similar to [58], we posted several tweets about eight different topics, as well as liked popular posts related to them. This was done using a bot. we created a bot which utilizes the Twitter API [73] to post (both original and retweets) regarding the topics from the top tweets feed, as well as liking them, at regular intervals. We ran this bot for the duration of two weeks, and found that our accounts were being followed by 153, 124 and 89 followers, respectively. Since we only want to evaluate the link visitation behaviour of real users, we removed suspected bot accounts using Botometer [14], using threshold of $s > 0.43$ [78], where $s$ is the score of the account. We blocked 31, 12, and 15 accounts which were bots, so that they do not get access to the benign URLs we post later on. We made our accounts private, so that users outside of our followers base do not get access to the Benign URLs.

### 7.2.2 Posting Benign Tweets.

We selected 40 random benign URLs, with 20 of them being Regular URLs (RU) not belonging to any bias category, and 20 of them which had been detected due to Web Hosting Bias (WH).

We put each of these URLs into two sets: (i) No warning set and (ii) Warning set.

The *No Warning* set includes: a) RU URLs; b) WH URLs which were shortened by bit.ly, a URL shortener which Twitter does not mislabel.

The *Warning set* includes: a) RU URLs which were shortened by one of the 4 URL shorteners that Twitter mistakenly warns against; b) WH URLs.

We then posted tweets using URLs from both sets on the three accounts and monitored their activity using the Twitter Analytic platform [**twitter˙ana**] to check for the visitation statistics for the links over a period of two weeks.

### 7.2.3 Results.

After two weeks we found that, all URLs in the *No Warning* set had been visited for a total of 5,790 times across the three accounts by 203 unique followers. This is nearly 45% more visits than those in the warning set, despite both sets having the same collection of URLs. The warning set only had 3,216 URL visitations by 155 unique followers across the 60 URLs in the *warning set*. Twitter Analytics does not provide any identifying information regarding which exact user visited which link, thus we cannot provide any descriptive statistics regarding any particular user behaviour. However, we could analyze which links were visited, and which were not. We elaborate this in Table 7.3. We found that users reported tweets containing 14, 21 and 6 URLs from our WA set in IA1,IA2 and IA3 respectively, suggesting that they believed that the tweet was really malicious after seeing the warning from Twitter. Twitter however informed us that they had investigated the tweets and found no violations, and thus our account was not suspended. At the end of two weeks, we also lost 83 followers across the three accounts. We assume that due to these warnings, accounts might potentially lose followers, as the latter might believe that the account is consistently posting malicious tweets, as has been demonstrated in our study in this section. However, it is not clear if these users unfollowed our account solely because of seeing the malicious tweets, as there can be several other reasons [67, 39] have explored how users unfollow due to various socio-economics reasons including offensive tweets. But, no work has been done to evaluate similar trends for malicious tweets, it is hard to determine the impact of such tweets on user impressions towards those accounts. However, considering that the accounts did only lost 5 followers together (2,0,3 for IA1,IA2 and IA3 respectively) before the WA set URLs were posted, despite posting tweets at a similar pace, and only lost the followers when the WA set URL tweets were posted, we can assume that they had a role in influencing the decision of the users to unfollow our account.

Table 7.3 shows that all accounts have consistently lost visitations when they posted URLs which had the characteristics of the WA set. Thus, we can say that Twitter's warnings against safe URLs impacts the traffic of the concerned websites. Table 7.4 also shows that all URLs have consistently lost visitations no matter their category.

Table 7.3: Account statistics for all three impact accounts after posting URLs from NW and WA set NW: No warning set visit, WA: Warning set visit. TF: Total followers, FL:Followers lost

| Account | NW visits | WA visits | % of visits lost | TF | FL |
|---|---|---|---|---|---|
| **Account 1** | 2,347 | 1,272 | 45.9 | 153 | 34 |
| **Account 2** | 2164 | 1132 | 47.7 | 124 | 18 |
| **Account 3** | 1299 | 814 | 37.4 | 89 | 21 |

Table 7.4: Statistics for each URL type with respect to: NWV - No warning set visits, and WAV: warning set visits

| URL source | Total | NWV | WAV | % visits lost |
|---|---|---|---|---|
| **Shortened URLs** | 20 | 2,773 | 1,469 | 47.03 |
| **Web hosting domains** | 20 | 3,017 | 1,747 | 42.11 |

### 7.2.4 Validating the Results.

In this section, we have determined that URLs which are misdetected by Twitter, and are posted by Twitter accounts have an effect on users visitations for those URLs. We further support our results in 8.1, where we analyze the responses of 120 respondents over M-Turk who try to investigate a URL after seeing that Twitter generates a warning for it.

#### Ethical Considerations.

The study conducted in this section is based on users visiting benign websites which do not attempt to mislead users or pose any harm to their personal or private data. Also, due to our reliance on the Twitter Analytics platform for obtaining the data regrading the link visitations of the posted URLs, we did not have access to any uniquely identifiable individual data. Also, throughout our study we did not collect any private data using any third-party tools. The only data used for analysis are the link visitation statistics, and tracking the loss of followers from the IA accounts.(Identifying the users who unfollowed our accounts was also not tracked, as we had not maintained any identifiable data for the follower accounts in the first place).

### Misinformation Scenarios

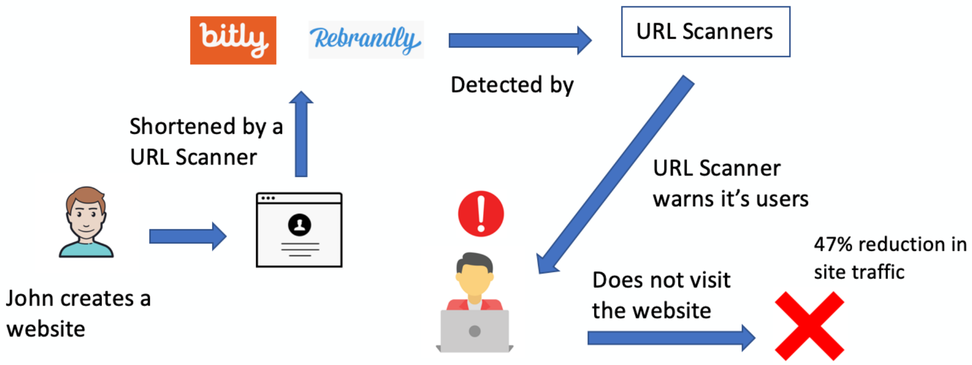

bitly
Rebrandly
Detected by
URL Scanners
Shortened by a URL Scanner
URL Scanner warns it's users
John creates a website
47% reduction in site traffic
Does not visit the website

Figure 7.2: A scenario where a benign website is detected as Malicious due to URL Shortener Bias

Similar to the attack scenario in 6.3, Benign URLs might also be a victim of the conservative detection approaches of URL Scanning Tools. But instead of an attacker manually submitting the URL on PhishTank, for URL Shortening Bias, every URL that is hosted under one of the 7 URL Shorteners (that are misdetected), will be flagged as Malicious. As you see in Fig 7.2, this time, John shares his e-commerce website to others by using a URL Shortening service. Now the URL Shortener is detected by several engines, and if one of those URL Scanning engines is used by a user, he/she would be less inclined to visit the website. This bias or misinformation can cause a 47% reduction in website traffic, i.e. John can lose 47 out of 100 potential customers.

Similarly, in 7.3, John has hosted his e-commerce website on a free web hosting domain that is frequently detected by these tools. This time, it will lead to a 42% reduction in web-traffic.

Thus, these misinformation factors can cause severe damage to web traffic of the misdetected website, which can be very problematic to the owner of the website.

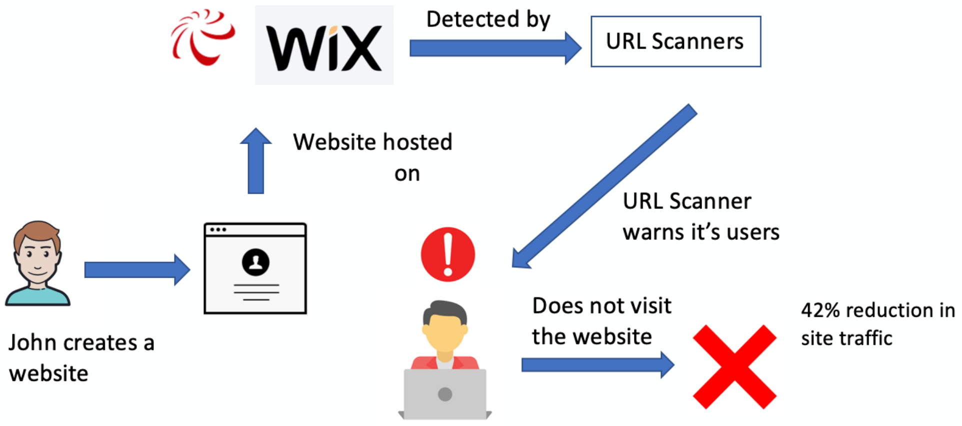

WiX
Detected by
URL Scanners
Website hosted on
URL Scanner warns it's users
John creates a website
Does not visit the website
42% reduction in site traffic

FIGURE 7.3: A scenario where a benign website is detected as Malicious due to Web Domain Bias

# Chapter 8

# Determining User Behaviour towards Misinformation

## 8.1 Survey Study

In the previous sections, using data-driven methods and emulative experiments, we showed that two different sources of information for identifying phishing websites, i.e. VirusTotal and Twitter, can misinform users, which might result in: (i) Accidental exposure to real threats, and (ii) Significant decrease in web traffic to the mislabeled websites. To corroborate these findings, we conducted an IRB approved survey study over 120 respondents using the Amazon Mechanical Turk (M-Turk) platform. The respondents are asked to investigate several URLs, and thus, in the process, we obtain information about what features Internet users investigate to identify malicious URLs, and how the misinformation provided by different sources influence their perception towards these websites. Specifically, we focus on the following research questions:

### Research Question 1 (RQ1)

Does the reports by detection tools, i.e., VirusTotal and Twitter, influence the perception of users visiting some websites?

### Research Question 2 (RQ2)

What features do Internet users investigate to identify malicious URLs?

To answer research questions *RQ1* and *RQ2*, we define five hypotheses:

### Hypothesis 1 (H1)

User participants have significantly different responses before and after they see the VirusTotal scores or the Twitter warning for a URL.

**Hypothesis 2 (H2)**

Depending on the URL, user participants have significantly different responses before and after they see VirusTotal scores or the Twitter warning for a URL.

**Hypothesis 3 (H3)**

User participants have significantly different responses before and after they see the *mislabeled* VirusTotal scores or Twitter warnings for a URL.

**Control Factors:**

There are some factors that can contribute to the responses provided by study participants. For example, if they see the first URL has been detected by many scan engines, and then the other URLs that are only detected by a couple of them, then they might have a higher trust in the verdict for the URL with higher detection rate. Also, if they have knowledge about some scan engines, e.g., have heard their names or have used them before, then they might have a higher trust in the results provided by them. To investigate the influence of such information on the responses, we define two more hypotheses:

**Hypothesis 4 (H4)**

Higher detection rates provided by VirusTotal results in users trusting VirusTotal verdict differently.

**Hypothesis 5 (H5)**

Showing the names of the scan engines that detected a URL make users trust VirusTotal verdict.

### 8.1.1 Study Design:

The survey asks study participants to investigate the maliciousness of 7 URLs, and choose from 3 labels: *Malicious*, *Not Malicious*, and *I am Not sure*. Then, they will be asked to investigate the maliciousness of the same 7 URLs, while having access to additional information about the URLs, which are provided by detection tools, i.e., VirusTotal and Twitter. Our goal is to study if the participants change their verdict after observing some information. For all questions, respondents were also asked to provide a short text response about their decision towards the website.

Table 8.1 shows the description for the URLs. We chose to include URLs that are correctly detected (true positives), incorrectly detected (false positives), and incorrectly not detected (false negatives). We added URLs that are correctly detected to simulate real life scenarios, where in many cases detection tools provide correct information. We also chose a verity of URLs, including Banking websites,

#### Comparison Groups

To investigate how factors such as observing a URL with higher detection rate, or observing the name of engines can impact the responses, we defined three groups:

#### Group 1

This group can also be considered as control group or comparison group, consists of the study participants whose survey include a URL with a very high detection rate, and they do not see the name of scan engines that have detected the URLs as malicious.

#### Group 2

This group consists of the study participants who see first URL in their survey having a higher detection rate of 14/80. However, it does not include the name of scan engines that have detected the URLs as malicious.

#### Group 3

This group consists of the study participants where the URLs in their survey are identical to those in Group 1, however, it also shows the name of the scan engines that detect a particular URL as malicious, and then proceeds to ask the user if they are familiar with one or more of the engines that detected the URL as malicious. Additionally, at the end they are also asked if they actively use a URL Scanning Tool. To validate our *H4* we compare the responses of *Group 1* and *Group 2*, and To validate our *H5* we compare the responses of *Group 1* and *Group 3*.

### 8.1.2 Survey Environment

Since the majority of the survey would include respondents visiting websites which were malicious and cause accidental damage to their private information, we built a secure environment for the respondents which had them connecting to a virtual machine (VM) located running on a system in our research lab using the *TeamViewer* remote-access software. When starting the survey on Amazon Mechanical Turk respondents were given a unique access token which they used to access the virtual machine. The virtual machine ran a custom built user interface based on *Openbox* running on top of the Ubuntu 18.04 operating system. This interface gave users access to only the open-source Chromium web browser. We disabled Google Safe Web Browsing protection on this web browser so that users can conduct an unbiased investigation through the URLs that were provided. Connecting to this VM using the access token automatically launched a survey hosted on *Qualtrics* survey platform, which provided them with the URL links and the corresponding questions of the survey. Browsing these websites inside the virtual machine made it impossible for any infection to occur on the respondent's own system. Throughout the survey, we also warned the respondents every-time while they accessed the URL, that they should not enter any personal information on these websites.

### 8.1.3 Survey Structure

The structure of the survey is divided into two parts:

**Consent form and Demographic Information:**

Respondents were asked questions regrading their age, time spent on the internet and field of employment. This was to make sure that we have a diverse set of respondents. We also asked them about how often they encounter malicious websites, and what was the source of the exposure.

**URL Analysis:**

Respondents were presented with 7 URLs. , with each of them being either a) Malicious, but has a feature which prevents them from being detected by the majority or all URL Scanning engines b) Non-Malicious, but has features which result in them being detected by 1 or more URL Scanning engines. Both a) and b) result in the URL Scanners providing some form of misinformation which can potentially confuse users. The respondents were asked to rate each of 7 URLs twice, once without the guidance of VirusTotal reports, and next time with the guidance of VirusTotal reports. Then, after evaluating each URL, they were asked to write a short textual response to explain the reason for their verdict.

This was done such that we can determine the impact of VirusTotal, and in turn, the affect of the URL Scanning engines on the user's verdict.

**Evaluating URLs without Detection Tools Guidance.**

We allowed participants to explore the websites on our VM as much as they wanted. As stated earlier, before visiting each website, they were given a warning not to enter any personal or uniquely identifiable information on these websites. If investigating the website required them to enter information (such as name, email address, etc.), they were encouraged to enter random or invalid text on these websites.

After investigating the website, users have to select one of three options which determine their verdict for the website:i)Malicious ii)Non-Malicious iii)Unsure. Irrespective of which option they select, they then have to provide a short text-based response on why the reason towards the website that they just investigated.

### Evaluating URLs without VirusTotal guidance

The respondents are first shown all the 7 URLs without informing them, in any way, if they are malicious or non-malicious. This lack of information encourages the user to visit these URLs and investigate themselves. Since the study takes place inside a VM, they could explore the websites as much as they wanted, without causing any damage to their physical systems. As stated earlier, before visiting each website, they were given a warning not to enter any personal or uniquely identifiable information on these websites. If investigating the website required them to enter information(such as Name, Email address etc.), they were encouraged to enter random and invalid text on these websites. After investigating the website, users have to select one of three options which determine their verdict for the website:i)Malicious ii)Non-Malicious iii)Unsure. Irrespective of which option they select, they then have to provide a short text-based response on why the reason towards the website that they just investigated.

### Evaluating URLs with Detection Tools Guidance.

The participants were presented with the same 7 URLs in a random order, with each URL now bearing a VirusTotal score along with it. which shows how many engines detected this URL as malicious. The participants see the message *This URL has been detected as malicious by 'n' engine*, where n is the number of engines that detected the URL as malicious. Note that the VirusTotal scores shown in this section were accurate, with each detection on VirusTotal being cross-checked with the respective engine's blacklist to make sure VirusTotal was not lagging behind on the detection label for that corresponding engine [47]. After seeing the VT score, the respondents are again encouraged to visit the URL and investigate it again. After investigating the URL, the respondents are asked to again rate the website as Malicious/Non-Malicious/Unsure. They are further asked again to give a response behind their verdict. After seeing the VT score, the user is again encouraged to visit the URL and investigate it again. After investigating the URL, the users is asked to again rate the website as Malicious/Non-Malicious/Unsure. They are further again asked to give a response behind their verdict. The main objective of this section is to determine how the reveal of the VT scores impacts the users with respect to changing their verdict. There lies the possibility that the user who had correctly identified a website as Malicious might later change their decision after seeing the VirusTotal score. The converse situation might also happened i.e. the user might label a non-malicious website as malicious after seeing the VT score. Depending on the Study group(See Section 4.2), users are either asked if they are familiar with the URL Scanners that detected the URL as malicious.

### General Protection Behaviour.

Finally, the respondents were queried on their behavioral habits while dealing with malicious websites, such as what they do when encountering one.

We chose to ask these questions at the end of survey because in the multiple choice options, we were providing some sources that they can use for detecting malicious URLs, and therefore, observing these sources could have affected how they approach labeling the URLs provided to them. The survey also asks if they are familiar with the concept of URL scan engines. If the user says *Yes*, they are asked to state the URL scan engines that they use.

### Time Constraint.

Surveys hosted on M-Turk must have a time-constraint, i.e., the maximum time a user is allowed to complete the survey and submit it. To determine the most optimum time for completing the survey, we availed the help of 8 volunteers who took our surveys as a trial run. We choose the volunteers in such a way that 4 of them claimed to be technologically pro-efficient and were working in a STEM field, while the other 4 were not so acquainted with the day to day usage of the technology, and worked in non-STEM fields. The average time for completion of the survey was 35 minutes. Thus, to be on the safer side, we estimated that the respondents to our survey would not need more than 40 minutes. Since participants in Group 3 have 7 extra (but brief choice-based) questions, we allotted them an extra 5 minutes.

Respondents had to adhere to these survey time limits, and complete answering all the questions in the survey. At the end of the survey, a unique survey code was generated for them, which they had to enter on their MTurk portal.

### Choosing and Labelling the URLs.

For choosing the URLs, the first preference was given to those which were already present in our dataset. We used a randomizer program to choose a particular URL and put it in a set for manual investigation by moderators. Each URL is then investigated and labelled manually by 3 individuals, separately. The URL is only included in the survey, if and only if all the 3 moderators

Table 8.1: URLs included in the survey and their characteristics

| URL | URL Group | Malicious? | VT Verdict | Reason for Verdict on VT |
|---|---|---|---|---|
| 1 | Bank Phishing Website | Yes | Malicious | Imitating legitimate bank website |
| 2 | Technology Blog | No | Malicious | Top level domain misdetection |
| 3 | Malicious Ad-Gateway | Yes | Malicious | Aggressive popups |
| 4 | Online Petition | No | Malicious | Misinformation in public blacklist |
| 5 | Phishing form(Basic) | Yes | Non-Malicious | Simple Google Forms Phishing |
| 6 | E-Commerce website accessed through Twitter | No | Malicious | URL Shortener Misdetection |
| 7 | Phishing form(Sophisticated) | Yes | Non-malicious | Sophisticated Google Forms Phishing |

decide on the same label. This approach was suitable for 5 of our 7 URLs since they were either non-malicious (and thus not taken down by the author of the website or the web host provider), or they were malicious, but they were not detected very quickly and thus stayed alive for the duration of our study. However, for the remaining 2 URLs, bank phishing website and Google phishing form (Basic), they had to be replaced, with the former being replaced thrice, and the later once, because they were taken down quickly. In the case of replacement, 10 fresh URLs with similar detection rate on VirusTotal are chosen, and labelled by the moderators. The URL that they all agree on to be malicious, is then included in the survey for that day. To make sure all URLs in the survey are functional and working as intended, we visited them everyday at the start of that day's batch of respondents who took the survey, and then intermittently during the survey period for that day (when other respondents were also taking the survey).

### 8.1.4 URLs Descriptions

Our final set of URLs include:

**URL 1:**

This URL (and it's replacements) were *Bank phishing URLs* which imitated legitimate banking websites, with the sole intention of stealing the login credentials of unsuspecting users.

**URL 2:**

This URL is a legitimate *Technology blog*, which is mis-detected by URL Scanners due to it being posted on a public blacklist (PhishTank)

**URL 3:**

This URL is a malicious Ad-Gateway. It portrays itself as a news website, but subjects the respondents to a large number of malicious popups, most of which lead to other malicious websites.

**URL 4:**

This URL contains a petition hosted on change.org. It is mis-detected by VirusTotal due to *URL shortener bias.*

**URL 5:**

This is basic phishing page hosted under Google forms. Despite being malicious, it is mis-detected by both URL Scanners and Twitter due to them trusting the top level domain(Whitelisting) Google.

**URL 6:**

This URL is a product page from an e-commerce website with a long URL (greater than 192 characters), which is then shortened by a URL shortener which is often mislabeled on Twitter as Malicious. This URL triggers a warning from Twitter, when respondents try to visit it.

**URL 7:**

This is a more sophisticated form of URL 5, which uses better graphics and design. This URL is similarly mis-detected due to Whitelisting trusted domains.

**Compensation and Ethical Considerations.**

We recruited the participants through Amazon M-Turk. We only considered individuals for participants if they were a) Residing in the US, b) Had a 95% approval rating on M-Turk, and c) Had more than 1,000 approved HITs (Surveys completed).

After collecting 147 responses, we had to discard 27 of them, which had low quality responses and could not be able to contribute to our study. Through M-Turk, Upon successful completion of the survey, the respondents were compensated monetarily ($4) for their participation. The study and compensation scheme was approved by our institution's ethics review board (IRB). During the study, the respondents had to visit several websites which were potentially dangerous, and we took very strict precautions to protect the respondent's privacy. Our safety measure are better elaborated in 8.1.2. We also did not collect any identifiable user data and warned the respondents to not enter any of their personal information on these websites.

### 8.1.5 Data Analysis

We start by testing *H4* and *H5* to see if factors such as observing a URL with higher detection rate, or observing the name of engines impact the participants' responses.

**Variables.**

We define two variables: *Group* and *Verdict*. The *Group* variable is categorical, which can get categories *Group 1* and *Group 2* for *H4*, and *Group 1* and *Group 3* for *H5*. The *Verdict* variable shows the counts for each of three possible verdicts, i.e., *malicious*, *not malicious*, and *I am not sure*.

**Statistical Test.**

To validate each of these hypotheses, we created the contingency table, which include two rows and three columns, filling with the counts of each of verdicts for each group (either Group 1 and Group 2 for *H4*, or Group 1 and Group 3 for *H5*). We used Pearson's Chi-Squared test to determine whether there is a statistically significant difference between the frequencies of verdicts across different groups.

### Analysis of *H4*

In the case of *H4*, our assumption is that, if respondents see a higher detection rate from VirusTotal for the first URL in *Group 2* (with 14 detection's), they might have a different perception of how VirusTotal scores work, and might have higher or lower trust when they see the VirusTotal verdicts for other URLs (with lower detection rates).

**Results.**

Our Chi-Squared test was rejected as the p-value was greater than the threshold (0.05). We also repeat the test, for each URL, and we found that for all the URLs, except URL 5, the p-value was higher than the threshold (0.05). Thus, we *cannot* reject the null hypothesis. This suggests that *seeing a higher VirusTotal score does not change the responses.*

### Analysis of *H5*

In the case of *H5*, our assumption is that, if respondents have a different trust level in the URL scan engines, then seeing the name of them might change their perception of websites.

Table 8.2: Chi-Squared Test for *H5* using responses by 40 respondents per group.

| **URL** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| **p-values** | 0.97 | 0.06 | 0.25 | 0.42 | 0.35 | 0.08 | 0.87 |

**Results.** In Table 8.2, we see that, when we compared the responses between the two groups, the p-value generated for all the URLs, are higher than the rejection threshold (0.05). Thus, we *cannot* reject the null hypothesis. This suggests that *seeing the name of URL scan engines do not change the responses.*

### Analysis of *H1*

. Our findings from *H4* and *H5* suggest that respondents are not influenced by higher VirusTotal detection's, or with their familiarity with the scan engines. Therefore, for testing other hypotheses, we merge the responses provided by all the groups (i.e., 120 participants).

### Variables.

We define two variables: *Exposure* and *Verdict*. The *Exposure* variable is categorical, which identifies if the URL is provided with additional information, i.e., the results of detection tools. This variable can be "yes" and "No." The *Verdict* variable shows the counts for each of three possible verdicts, i.e., *malicious*, *not malicious*, and *I am not sure*.

### Statistical Test.

To validate *H1*, we created the contingency table, which include two rows and three columns, filling with the counts of each of verdicts for being or not being exposed to additional information. We used Pearson's Chi-Squared test to determine whether there is a statistically significant difference between the frequencies of verdicts when users receive the additional information.

### Results.

Considering all the verdicts for all URLs, we found the p-value for the Chi-Squared test to be above 0.05, where we cannot reject the null hypothesis. This finding suggests that exposure to the VirusTotal verdict has no effect on the overall verdicts.

### Analysis of *H2*

Our survey URLs include different types of websites, with some being legitimate and others being malicious. Our assumption is that exposure to VirustTotal results might have a different impact on the verdicts of different types of URLs. To validate *H2*, we do the Chi-Squared test on the responses of each URL separately.

Table 8.3: Chi-Squared Test for *H2* using responses by 120 respondents.

| **URL** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| **p-values** | 0.59 | 0.27 | 0.46 | 0.03 | 0.03 | 0.38 | 0.01 |

**Results.**

Table 8.3 shows the results. In this case, the results are different, in the way that we found 3 URLs (4, 5 and 7) which have consistently shown p values lesser than 0.05 for all three groups. This indicates that there is a correlation between user verdict and them being exposed to the VirusTotal scores, thus proving our hypothesis correct. On the other hand, we found URL 1 and 3 have high $p$ values across all three groups.

Therefore, our previous assumption was correct,that seeing VirusTotal scores and then rating the websites has a co-relation with respect to the respondent. *But*, the co-relation might be dependent on the specific URL, as we have seen in Table 8.3.

Now, we asked ourselves, why do some URLs have lower p-values, but others have high p-values. This difference can be due to the fact that the URLs are characteristically different. Some might have certain characteristics or features which users put more confidence in, while others have features which users do not trust, and thus rely more on VirusTotal in that case. In the proceeding sections, we seek to determine it by using the metric *Impact Factor* metric in 8.1.7.

### 8.1.6 Analysis of Changes in Verdicts

**Analysis of *H3***

Our assumption is that exposure to VirustTotal results might have a different impact when the VirustTotal results are false and misinforming. To validate *H3*, we divide the 7 URLs to False Postive-False Negative(FP-FN) and True Positive(TP) sets, and run the Chi-Squared test on the responses of these groups. The TP set consists of URLs 1 and 3. These URLs are malicious in nature, and both VirusTotal and Twitter detect them correctly. The FP-FN set consists of the rest of the URLs (2,4,5,6,7). Here URLs 2, 4 and 6 are FPs, and URLs 5 and 7 are FNs.

**Statistical Test**

To validate *H3*, for each of the FP-FN and TP sets, we created the contingency table, which include two rows and three columns, filling with the counts of each of verdicts for being or not being exposed to additional information. We used Pearson's Chi-Squared test to determine whether there is a statistically significant difference between the frequencies of verdicts when users receive the additional information.

### Results

Based on Chi-Squared test, we could not reject the null hypothesis for the URLs in the TP set with $p = 0.46$. However, we could reject the null hypothesis for the URLs in the FP-FN set with $p = 0.07$. This finding shows that users are more reliant on VirusTotal and Twitter when these tools misinforms the user.

We find that URLs in the TP set i.e. URLs 1 and 3 have very high p-values. On the other hand URLs in the FP-FN set have all have lower p-values, with URLs 4,5,7 being lower than our rejection threshold. Thus, we can say with some degree of confidence, that users are more reliant on VirusTotal and Twitter when these tools misinform the user. Meanwhile, URL 6 had a higher p-value despite being in the FP-FN set due to having a very The only outlier was URL 6, for which Twitter warns against the website due to URL shortener bias. From Table 8.5 we found that more than 50% of respondents were confident about Twitter's warning, and did not change their verdict, even after seeing the VirusTotal warning. This is compounded by the fact that 2 engines in VirusTotal also detected this URL as malicious. Thus, users who already incorrectly labelled this URL as malicious after seeing Twitter's warning, were further validated by VirusTotal's mislabeling. We further strengthen our findings in the Rate of Change analysis below.

### Analysis of Rate of Change

In this analysis, we consider the rate of change of verdicts i.e. how many verdicts changed after seeing the VirusTotal score. A higher rate of change means that respondents were influenced more by VirusTotal, and a higher rate of change for URLs in the FP-FN set would support our finding from the previous analysis, that, respondents were more unsure about their responses for URLs for which VirusTotal or Twitter misinforms in their detection.

Table 8.4: Rate of Change for TP and FP-FN sets

| **URL** | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| **TP** | 11% | NA | 18% | NA | NA | NA | NA |
| **FP-FN** | NA | 28% | NA | 37% | 33% | 18% | 47% |

### Results.

We found the p-value for the Chi-Squared test to be below 0.05, where we can reject the null hypothesis. This finding suggests that exposure to the VirusTotal verdict when they are false has an effect on the respondents' verdicts.

Table 8.4 shows that URL 1, which is a Bank Phishing URL, has the lowest change of verdict among all the URLs, suggesting that respondents were more aware of this URL to be actually Phishing. The URLs 4, 5 and 7 had the most verdict changes from *negative* to *positive* (URL 4), and *positive* to *negative* (URLs 5 and 7). The only outlier was URL 6, which had a lower rate of change. Here, Twitter warns against the website in URL 6, due to URL shortener bias. From Table 8.1.7 we find that more than 50% of respondents were confident about Twitter's warning, and did not change their verdict, even after seeing the VirusTotal warning. This is compounded by the fact that 2 engines in VirusTotal also detected this URL as malicious. Thus, users who already incorrectly labelled this URL as malicious after seeing Twitter's warning, were further validated by VirusTotal's mislabeling. Thus, looking at the Rate of Change for this URL is irrelevant, and we can safely say that misinformation from two sources impact the respondents in rating this URL.

For URLs 2 and 4, which are a Tech Blog and a Change.org petition Many users label them as benign, as they are actually benign, however, many of them change their verdict after seeing the VirusTotal scores, which labelled both of them incorrectly as Malicious. This contributes heavily to users being confused, and later on changing their verdict after seeing the VirusTotal scores, which labelled both of them incorrectly as Malicious.

In contrary, URL 5 and 7 are phishing websites that are created on the *Trusted sites*. Interestingly, we see a large number of verdicts change after they see that none of the VirusTotal engines have detected these URLs as malicious.

Thus, we see that for the URLs that both VirusTotal and Twitter misinform the users (FPs or FNs), users become unsure of their responses and start relying more on VirusTotal or Twitter's verdict. The problem in this scenario is that, in this way, users might accidentally access a malicious website, and on the other hand, refrain from visiting legitimate websites which are detected due to several misinformation factors that we have discussed earlier in this paper.

### 8.1.7 Impact of Website Features

In this section, we try to answer the *RQ2*, and identify the features that Internet users use to investigate the legitimacy of websites. For that purpose, we analyze the qualitative responses that the participants provided by justification for their labels to URLs. The goal is to determine which features in a website make the respondent more or less confident in their own ability of providing a verdict towards them. Being less confident towards a feature would mean they would be more reliant on VirusTotal, and thus their verdict would change, given VirusTotal's own verdict is different from theirs.

### Coding the Responses.

We manually coded all short answers provided by the $120 * 7 = 840$ responses obtained from labeling the URLs before exposure to detection tools results. Each response was labelled as one of eight different features. A feature is a one-word substitute which summarizes the characteristics which made the user rate the URL as malicious or not malicious. A URL might have multiple features.

### Features that are used to label responses

#### Imitation

This category says that the websites were imitating some other popular website.This feature contributed to respondents labelling the URL as *Malicious.*

#### SSL cert

Discussing the existence of SSL certificate. This feature contributed to respondents labelling the URL as *Malicious* and *Non-Malicious.*

#### Safe site

Respondents could not find anything wrong with the website. This feature contributed to respondents labelling the URL as *Non-Malicious.*

#### Popups

Respondents encountered pop-ups or redirects. This feature contributed to respondents labelling the URL as *Malicious.*

#### Twitter

Respondents believing the warning given by Twitter for a particular URL. This feature is only seen in URL 6. This feature contributed to labelling the URL as *Malicious.*

#### Search

Respondents search for the website on search engines such as Google, Bing, DuckDuckGo, etc. This feature is only seen in URL 6. This feature contributed to labelling the URL as both *Malicious* and *Non-malicious.*

### News

Respondents believe that the website delivers only news or blog content. This feature contributed to labelling the URL as *Non-Malicious*.

### Trusted site

Respondents trust the website which hosts the URL. This feature contributed to labelling the URL as *Non-Malicious*.

### Feature Influence.

Factor influence ($FI$) is defined as total number of participants that have mentioned a particular feature and have not changed their verdicts due to that feature $P_F(N)$ divided by total number of participants that have mentioned a particular feature ($P_F$).

$$FI = \frac{P_F(N)}{P_F}$$

Factor influence determines by how much a feature played a role in the respondent *not* changing their verdict after being exposed to the VirusTotal scores. A score of 1 for $FI$ for a feature means that this feature has not influenced the respondents to change their verdict.

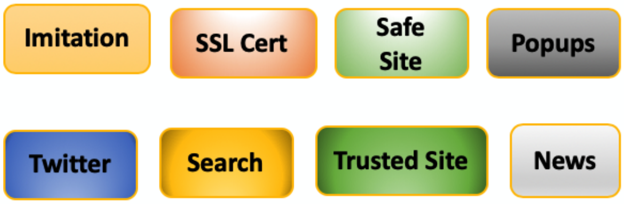


Figure 8.1: List of all impact factors that we have considered

### Results.

Our observations have been illustrated in Table 8.5, where the $FI$ is the average FI over all 120 participants in a group. We found that 8 factors induce confidence in respondents up to some degree. To check the effectiveness of each factor throughout our URL set, we average the scores. However, as we can see from the Table, Each feature contributes disproportionately throughout the URLs i.e. some URLs are more affected by one feature. The last column in Table 8.5 shows the URLs where a feature has the most influence on their verdict. (Such as a phishing form would have more influence from the Imitation factor rather than News). From this analysis, we find that *Imitation* has the most influence on URLs which are phishing (0.41), which means 41% of users identify that phishing websites are imitating or trying to steal personal data, and are confident with their decision, so that the VirusTotal score do not influence their decisions. This observation lines up perfectly with Table 8.4, where we see that URL 1, which is a Bank Phishing website, has the least no of verdict changes across all three groups. However, URL 5 and 7 were also phishing websites, but for them, the *Trusted Site* feature also contributes since those URLs are hosted under a trusted domain (Google Forms and Google Sites). But individual Imitation scores for these URLs are lower than the average (0.34 and 0.30 respectively), again suggesting that respondents are more influenced by the Trusted Site factor for these websites than Imitation, and thus we see a lot of verdicts change for this URL. We also found that the *Twitter* feature has a very high (0.52) $FI$. Twitter triggers a warning for only URL 6, and we see that 52% of users cite the URL as malicious, and are confident of their verdict even after seeing the VirusTotal scores. However, URL 6 is a false positive, and hence the dependence on Twitter leads to misinformation regarding this particular URL. This is consistence with our finding in Section 7.2, such that if users receive a warning from Twitter for a website, the website receives much less web traffic. Thus, this further supports our notion that Twitter generated warnings for legitimate URLs has an impact on respondent (user) behaviour.

Table 8.5: FI scores of all URLs

| **Factors** | **URLs(No.of respondents=120)** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **Sel. URLs** |
| **Imitation** | 0.6 | 0.1 | 0.10 | 0 | 0.34 | 0.02 | 0.31 | 1,5,7 |
| **SSL Cert** | 0.2 | 0.2 | 0.2 | 0.1 | 0.2 | 0.1 | 0.3 | All |
| **Safe site** | 0.074 | 0.2 | 0.1 | 0.13 | 0.2 | 0.22 | 0.14 | 2,4,6 |
| **Popups** | 0.02 | 0.1 | 0.42 | 0.1 | 0.03 | 0.1 | 0.04 | 3 |
| **Twitter** | 0 | 0 | 0 | 0 | 0 | 0.52 | 0 | 6 |
| **Search** | 0.11 | 0.2 | 0.1 | 0.2 | 0.1 | 0.8 | 0.04 | All |
| **News** | 0 | 0.3 | 0.13 | 0.04 | 0 | 0.01 | 0 | 2,3 |
| **Trusted site** | 0 | 0.03 | 0.01 | 0.53 | 0.2 | 0.03 | 0.22 | 5,7 |

**Summary.**

We can summarize our findings from this study as follows:

1. Factors such as showing higher VirusTotal detection rates for URLs or providing the name of URL Scan engines have no impact on user verdict.

2. Respondents were confused about 3 different URLs after they saw the VirusTotal score.

3. Respondents are more confident about some URL features than others as evidenced by the Impact Factor metric.

4. Respondents rely on VirusTotal verdict the most when the URL has some features which they are not confident about; and

5. Respondents tend to change their verdicts more for URLs which were misinformed by both VirusTotal and Twitter.

## 8.2 Survey Questionnaire

### 8.2.1 Demographic Information

1. Please choose the appropriate options for the questions below:

    - 18-20
    - 21-29
    - 30-39
    - 40-49
    - 50-59
    - 60 or older

2. Please specify your race:

    - White
    - Black or African-American
    - American Indian or Alaskan Native
    - Asian
    - Native Hawaiian or other Pacific Islander
    - From multiple races
    - Some other race (please specify)

3. Gender

    - Male
    - Female

- Other

4. Education

   - Less than high school degree
   - High school degree or equivalent (e.g. GED)
   - Some college but no degree
   - Associate degree
   - Bachelor's degree
   - Graduate degree

5. What is your background in the technological field?

   - I do not use technology that often
   - I use technology on a day to day basis, but I do not work in a STEM field,and I am not proficient in technological aspects
   - I do not work in a STEM field, but I am very proficient in technological aspects.
   - Computer Scientist/Engineer
   - Scientist/Engineer in another field of study

6. How much time do you spend online everyday?

   - Less than an hour
   - 1-2 hours
   - 2-3 hours
   - 4-5 hours
   - 6+ hours

7. How often do you encounter a malicious website?

   - Every day
   - At least once a week
   - At least once a month

- At least a few times a year
- At least once a year
- Never

8. How were you exposed to them? (Select all options that apply)
   - I clicked on a Link in an email
   - I clicked on an Advertisement on another website
   - It was one of the search results on Google/Bing/Yahoo Search etc.
   - It was posted on social media

### 8.2.2 URL Analysis

**Before VirusTotal report is shown:**

*The question structure is used for Questions 1-7 i.e. for evaluating the URLs before the VirusTotal score is shown.* For this question, you will be required to visit the URL given below, observe it then rate the URL based on the options below : **Note:** You are allowed to use any resources or tools which will aid you in rating this website.

Please visit this URL:

**http://xxxx.xyz/yyyy**

*Note that the real URLs used in this study can be Malicious in nature, and thus, they have been hidden.*

Do you think this website is malicious?

- It is malicious
- It is not malicious
- I am not sure

Please write in short, why you think this website is or isn't Malicious? *Text box*

**After VirusTotal report is shown:**

*The question structure is used for Questions 8-14 i.e. for evaluating the URLs after the VirusTotal score is shown. Parts of the question that are exclusive to Group 3 are noted seperatley.*

The following URL has been detected as Malicious on VirusTotal by X out of 80 engines.

Please visit this URL to investigate: **http://xxxx.xyz/yyyy**

After seeing the VirusTotal score and investigating the URL again, do you think it is malicious?

- It is malicious
- It is not malicious
- I am not sure

Please write in short, why you think this website is or isn't Malicious? *Text box*

*Exclusive section for Group 3 only:*

The URL Scanners which detected this URL as malicious are X, Y and Z.

Please tell us how familiar you are with the URL scanning engines that detected this URL as malicious.

X

- I have used this engine
- I have heard about this engine
- I just checked this engine online
- I have not used or heard of this engine before

*Similar pattern for Engins Y and Z also*

### 8.2.3 Attention Questions

*These Questions are interjected between the other ones to make sure that the respondent is actually taking the survey diligently. The survey consisted of two attention questions, and failing any one of them would result in termination of the survey.*

Attention 1 For this question, you will be required to visit the URL given below, observe it then rate the URL based on the options below :

Note: You are allowed to use any resources or tools which will aid you in rating this website. Please visit this URL:

http://this-is-just-an.attention&checkurl.com

- It is malicious
- It is not malicious
- I am not sure

### 8.2.4 Internet Behaviour

What do you do when you encounter a website that looks suspicious to you?

Select all that apply

- I close the browser window
- I run a scan on my system using my Antivirus
- I check the link using one or more URL Scanning Tool
- I search for articles/forum posts about the website to see what others feel about it

Select your familiarity with URL Scanning Tools from the options below:

- I was not aware of URL Scanners before taking this survey
- I am aware of URL Scanning Tools but don't use them
- I check suspicious link using one or more URL Scanning Tool(s)

# Chapter 9

## Conclusion and Future work

In this work, we have evaluated more than 3k new URLs that both URL Scanners and Twitter had detected as Malicious. We found that 71% of these URLs were actually benign in nature(3.3), and upon further investigation in Sections 6.1, 6.2,7.1.1 and 7.1.2, we found three factors - URL Shortener Bias, Web Hosting bias and PhishTank which result in the detection tools detecting these Benign URLs. We also found through our emulative study in Sections 7.2, that posting these misdetected URLs on Twitter results in an average decline of 42% in web traffic to these Benign URLs. For Benign URLs which were misdetected due to being posted on PhishTank, we found a possible approach with which attackers can poison the detection verdict of Benign websites in 7.2.4. Meanwhile, only 13% of our dataset contained Phishing URLs, and a majority of them were not removed and were sparsely detected by URL Scanners even after a month(3.3). We found a new type of Phishing Attack which leverages the trust of popular domains to keep themselves from being detected. These attacks also use two obfuscation methods - Image based and Character based, which we have explored in 5.1 and 5.2. We also found that URL Scanners are partially dependent on PhishTank to detect this Phishing Attack.

We also found that Twitter is often reliant on manual user reporting for removing newer Malicious URLs. We see this trend especially in accounts which rapidly post large volumes of malicious content, with Twitter relying on the slow process of users reporting these accounts, rather than suspend them based on their behaviour4.4.

Finally, in our survey study, we obtained an idea about the impact of the misinformation propagated by both the URL Scanning tools and Twitter. We found that respondents frequently put more confidence on some website characteristics(Factors), in which case they were not reliant on the detection verdict of the URL Scanning tools, with the converse occurring when websites had factors which respondents had little confidence in. To measure this relation, we introduced a new metric in 8.1.7, *Factor Influence* and proved that the impact of these factors is indeed very significant 8.1.7. In 8.1.6, we also find that respondents were unsure about your their own investigation of a website the most, when the detection tools provided a wrong verdict for it.

### Limitations and Future Work

Our work while being able to successfully establish several characteristic of both Twitter and URL scanning engine's performance against phishing and legitimate URLs, has a few limitations. Firstly, our dataset only consists of tweets that had not been detected by Twitter during the time of data collection, thus our work is focused on features of URLs which Twitter is generally not adept at detecting. Twitter's black-box nature with regards to blocking URLs makes it difficult to determine the scale of phishing or other forms attacks which are attempted by the attackers on the platform. We try to replicate such scenarios with a smaller dataset of URLs collected from public blacklists, but our findings are not conclusive enough to generalize Twitter's detection capabilities against all types of Malicious URLs.A much larger dataset with more instances of these URLs will be helpful in generalizing our results more.

We also found a new phishing attack technique, Web-form URLs hosted on trusted domains such as Google Forms and Google Sites to be a newer form of threat which is present on both Twitter and public blacklists suggesting their popularity. With the lack of proper documentation regarding this new form of phishing attacks, future work can determine how widespread this attack is, and how it impacts regular users.

Also, while we found that Twitter is reliant on user reports to block accounts which regularly exhibit malicious behaviour, we did not study the scale at which such malicious accounts which have not been suspended can propagate malicious URLs through their followers and friends network. We leave this work for future study. For the categories of URLs that we have identified that frequently avoid detection by Twitter, we have not evaluated the rate of their propagation through the network of users in our dataset, limiting our study only to the URLs we had initially collected, and the ones we acquired later on from Public blacklists. While we manually observed all detected malicious URLs, we had initially used VirusTotal to differentiate malicious URLs from legitimate ones, and due to VirusTotal sometimes running a stripped down version of the respective scanning engines, it is possible that some malicious URLs were not detected during our initial scans, though we try to rectify this issue by frequently scanning the URLs and cross-verifying them with the engines.

We were conservative in stating what percentage of legitimate URLs are misdetected by URL scanners due to the former being posted on Public blacklists. This is due to the fact that we only investigated one public URL blacklist in this regard - PhishTank. It is possible that URL scanners rely on other blacklists as well, and this phenomenon of misdetection can be more widespread than what we have found in our experiment.

In our survey study, we have found how respondents are affected by verdicts from both VirusTotal and Twitter. Our survey population was diverse, having respondents from all ages and professional backgrounds, including their technological proficiency. Yet, we did not determine if VirusTotal ratings affect respondents who are more knowledgeable in the field of technology and those who are not differently. We leave this analysis for future work.

# Bibliography

[1] *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/l. 2020.

[2] *101+ Web Hosting Stats and Facts To Help Choose A Better Host*. https://hostingtribunal.com/blog/web-hosting-statistics/. 2020.

[3] *84% of Phishing Sites Last for Less Than 24 Hours]*. https://www.infosecurity-magazine.com/news/84-of-phishing-sites-last-for-less/. 2020.

[4] Neda Abdelhamid, Fadi Thabtah, and Hussein 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.

[5] *About suspended accounts*. https://help.twitter.com/en/managing-your-account/suspended-twitter-accounts. 2020.

[6] Hassan YA Abutair and Abdelfettah Belghith. "Using case-based reasoning for phishing detection". In: *Procedia Computer Science* 109 (2017), pp. 281–288.

[7] Ahmed Aleroud and Lina Zhou. "Phishing environments, techniques, and countermeasures: A survey". In: *Computers & Security* 68 (2017), pp. 160–196.

[8] Nadia Alshahwan et al. "Detecting malware with information complexity". In: *Entropy* 22.5 (2020), p. 575.

[9] *Amazon Mechanical Turk*. https://www.mturk.com/. 2020.

[10] *Anti-Phishing Test July 2013]*. https://www.av-comparatives.org/tests/anti-phishing-test-july-2013/. 2020.

[11] Simon Bell, Kenny Paterson, and Lorenzo Cavallaro. "Catch Me (On Time) If You Can: Understanding the Effectiveness of Twitter URL Blacklists". In: *arXiv preprint arXiv:1912.02520* (2019).

[12] *Bitdefender TrafficLight]*. https://www.bitdefender.com/solutions/trafficlight.html. 2020.

[13] Michael Bossetta. "A simulated cyberattack on Twitter: Assessing partisan vulnerability to spear phishing and disinformation ahead of the 2018 US midterm elections". In: *arXiv preprint arXiv:1811.05900* (2018).

[14] *Botometer*. https://botometer.iuni.iu.edu/. 2020.

[15] David A Broniatowski et al. “Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate”. In: *American journal of public health* 108.10 (2018), pp. 1378–1384.

[16] Jan-Willem Hendrik Bullée et al. “On the anatomy of social engineering attacks—A literature-based dissection of successful attacks”. In: *Journal of investigative psychology and offender profiling* 15.1 (2018), pp. 20–45.

[17] Marcus Butavicius et al. “Breaching the human firewall: Social engineering in phishing and spear-phishing emails”. In: *arXiv preprint arXiv:1606.00887* (2016).

[18] Ferhat Ozgur Catak and Ahmet Faruk Yazı. “A Benchmark API Call Dataset for Windows PE Malware Classification”. In: *arXiv preprint arXiv:1905.01999* (2019).

[19] J. Charlton et al. “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.

[20] Chao Chen et al. “Investigating the deceptive information in Twitter spam”. In: *Future Generation Computer Systems* 72 (2017), pp. 319–326.

[21] Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. “A survey of phishing attacks: their types, vectors and technical approaches”. In: *Expert Systems with Applications* 106 (2018), pp. 1–20.

[22] Sharvari Prakash Chorghe and Narendra Shekokar. “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.

[23] *CRDF Threat Center]*. https://threatcenter.crdf.fr/. 2020.

[24] *ESET Security]*. https://www.eset.com/us/. 2020.

[25] *FortiGuard]*. https://fortiguard.com/webfilter. 2020.

[26] Edwin D Frauenstein and Stephen 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.

[27] *Friends of PhishTank*. https://www.phishtank.com/friends.php. 2020.

[28] *Google: 10 percent of sites are dangerous]*. https://www.cnet.com/news/google-10-percent-of-sites-are-dangerous/. 2020.

[29] Brij B Gupta et al. “Fighting against phishing attacks: state of the art and future challenges”. In: *Neural Computing and Applications* 28.12 (2017), pp. 3629–3654.

[30] Surbhi Gupta, Abhishek Singhal, and Akanksha 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.

[31] Reza Hassanpour et al. “Phishing e-mail detection by using deep learning algorithms”. In: *Proceedings of the ACMSE 2018 Conference*. 2018, pp. 1–1.

[32] *How Twitter wraps URLs]*. `https://developer.twitter.com/en/docs/basics/tco`. 2020.

[33] Amir Javed, Pete Burnap, and Omer Rana. “Prediction of drive-by download attacks on Twitter”. In: *Information Processing & Management* 56.3 (2019), pp. 1133–1145.

[34] Marianne Junger, Lorena Montoya, and F-J Overink. “Priming and warnings are not effective to prevent social engineering attacks”. In: *Computers in human behavior* 66 (2017), pp. 75–87.

[35] Chris Kanich, Stephen Checkoway, and Keaton Mowery. “Putting Out a HIT: Crowdsourcing Malware Installs.” In: *WOOT*. 2011, pp. 71–80.

[36] *Kaspersky Threat Intelligence Portal]*. `https://opentip.kaspersky.com/`. 2020.

[37] A. Kharraz, W. Robertson, and E. Kirda. “Surveylance: Automatically Detecting Online Survey Scams”. In: *2018 IEEE Symposium on Security and Privacy (SP)*. 2018, pp. 70–86.

[38] *Link Shorteners in Phishing Attacks*. `https://bit.ly/3dj4Zq3`. 2020.

[39] Suman Kalyan Maity, Ramanth Gajula, and Animesh Mukherjee. “Why Did They #Unfollow Me? Early Detection of Follower Loss on Twitter”. In: *Proceedings of the 2018 ACM Conference on Supporting Groupwork*. GROUP ’18. Sanibel Island, Florida, USA: Association for Computing Machinery, 2018, pp. 127–131. ISBN: 9781450355629. DOI: `10.1145/3148330.3154514`. URL: `https://doi.org/10.1145/3148330.3154514`.

[40] *Malware Bazaar ]*. `https://bazaar.abuse.ch/`. 2020.

[41] Jian Mao et al. “Phishing-alarm: robust and efficient phishing detection via page component similarity”. In: *IEEE Access* 5 (2017), pp. 17020–17030.

[42] Rima Masri and Monther 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.

[43] David Modic and Ross Anderson. “Reading this may harm your computer: The psychology of malware warnings”. In: *Computers in Human Behavior* 41 (2014), pp. 71–79.

[44] Mahmood Moghimi and Ali Yazdian Varjani. “New rule-based phishing detection method”. In: *Expert systems with applications* 53 (2016), pp. 231–242.

[45] Adam Oest et al. “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.

[46] *OpenPhish*. `https://openphish.com/faq.html`. 2020.

[47] Peng Peng et al. “Opening the blackbox of virustotal: Analyzing online phishing scan engines”. In: *Proceedings of the Internet Measurement Conference*. 2019, pp. 478–485.

[48] Tianrui Peng, Ian Harris, and Yuki 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.

[49] *PhishTank*. https://www.phishtank.com/faq.php. 2020.

[50] Paul Prasse et al. "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.

[51] Babak Rahbarinia, Marco Balduzzi, and Roberto Perdisci. "Exploring the long tail of (malicious) software downloads". In: *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE. 2017, pp. 391–402.

[52] Routhu Srinivasa Rao and Alwyn Roshan Pais. "Detection of phishing websites using an efficient feature-based machine learning framework". In: *Neural Computing and Applications* 31.8 (2019), pp. 3851–3873.

[53] Elissa M Redmiles, Sean Kross, and Michelle L Mazurek. "How well do my results generalize? comparing security and privacy survey results from mturk, web, and telephone samples". In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019, pp. 1326–1343.

[54] Ozgur Koray Sahingoz et al. "Machine learning based phishing detection from URLs". In: *Expert Systems with Applications* 117 (2019), pp. 345–357.

[55] 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.

[56] Ameya Sanzgiri, Jacob Joyce, and Shambhu Upadhyaya. "The early (tweet-ing) bird spreads the worm: An assessment of twitter for malware propagation". In: *Procedia Computer Science* 10 (2012), pp. 705–712.

[57] John Seymour and Philip Tully. "Weaponizing data science for social engineering: Automated E2E spear phishing on Twitter". In: *Black Hat USA* 37 (2016), pp. 1–39.

[58] M. Shafahi, L. Kempers, and H. Afsarmanesh. "Phishing through social bots on Twitter". In: *2016 IEEE International Conference on Big Data (Big Data)*. 2016, pp. 3703–3712.

[59] Himani Sharma, Er Meenakshi, and Sandeep Kaur 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.

[60] Hossein Shirazi, Bruhadeshwar Bezawada, and Indrakshi Ray. "" Kn0w Thy Doma1n Name" Unbiased Phishing Detection Using Domain Name Based Features". In: *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*. 2018, pp. 69–75.

[61] Gunikhan Sonowal and KS Kuppusamy. “PhiDMA–A phishing detection model with multi-filter approach”. In: *Journal of King Saud University-Computer and Information Sciences* 32.1 (2020), pp. 99–112.

[62] Jayesh Sreedharan and Rahul Mohandas. *Systems and methods for risk rating and proactively detecting malicious online ads*. US Patent 9,306,968. Apr. 2016.

[63] Alex Sumner and Xiaohong Yuan. “Mitigating Phishing Attacks: An Overview”. In: *Proceedings of the 2019 ACM Southeast Conference*. 2019, pp. 72–77.

[64] Bo Sun et al. “Automating URL blacklist generation with similarity search approach”. In: *IEICE TRANSACTIONS on Information and Systems* 99.4 (2016), pp. 873–882.

[65] Choon Lin Tan, Kang Leng Chiew, KokSheik Wong, et al. “PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder”. In: *Decision Support Systems* 88 (2016), pp. 18–27.

[66] Yasuyuki Tanaka, Mitsuaki Akiyama, and Atsuhiro Goto. “Analysis of malware download sites by focusing on time series variation of malware”. In: *Journal of computational science* 22 (2017), pp. 301–313.

[67] Liyang Tang and Zhiwei Ni. “Emerging opinion leaders in crowd unfollow crisis: a case study of mobile brands in Twitter”. In: *Pattern Analysis and Applications* 19.3 (2016), pp. 731–743.

[68] *Tensorflow*. https://www.tensorflow.org/. 2020.

[69] *The top 500 sites on the web ]*. https://www.alexa.com/topsites. 2020.

[70] *The Twitter Rules*. https://help.twitter.com/en/rules-and-policies/twitter-rules. 2020.

[71] Hsin-yi Sandy Tsai et al. “Understanding online safety behaviors: A protection motivation theory perspective”. In: *Computers & Security* 59 (2016), pp. 138–150.

[72] *Tweepy - An easy-to-use Python library for accessing the Twitter API*. https://www.tweepy.org/. 2020.

[73] *Twitter Developer]*. https://https://developer.twitter.com/en/docs. 2020.

[74] *Twitter now blocking bad URLs, but imperfectly]*. https://www.computerworld.com/article/2526696/update--twitter-now-blocking-bad-urls--but-imperfectly.html. 2020.

[75] *Twitter:Our approach to blocking links]*. https://help.twitter.com/en/safety-and-security/phishing-spam-and-malware-links. 2020.

[76] *Up to Three Percent of Internet Traffic is Malicious, Researcher Says]*. https://www.csoonline.com/article/2122506/up-to-three-percent-of-internet-traffic-is-malicious--researcher-says.html. 2020.

[77] *URLHaus*. https://urlhaus.abuse.ch/about/. 2020.

[78] Onur Varol et al. "Online human-bot interactions: Detection, estimation, and characterization". In: *Eleventh international AAAI conference on web and social media*. 2017.

[79] *VirusTotal API]*. https://support.virustotal.com/hc/en-us/articles/115002100149-API. 2020.

[80] *VirusTotal]*. https://www.virustotal.com/gui/home/. 2020.

[81] Fengguo Wei et al. "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.

[82] Ping Yi et al. "Web phishing detection using a deep learning framework". In: *Wireless Communications and Mobile Computing* 2018 (2018).

[83] Shuofei Zhu et al. "Measuring and Modeling the Label Dynamics of Online Anti-Malware Engines". In: *29th {USENIX} Security Symposium ({USENIX} Security 20)*. 2020.