How Many Phish Can Tweet?

Investigating the Effectiveness of Twitter's Phishing and Malware Defence System

by

Simon J. Bell

Submitted for the degree of
Doctor of Philosophy in Cyber Security

Information Security Group
School of Engineering, Physical and Mathematical Sciences
Royal Holloway, University of London

March 2020

Declaration

I, Simon Bell, hereby declare that this thesis titled, *"How Many Phish Can Tweet? Investigating the Effectiveness of Twitter's Phishing and Malware Defence System"* and the work contained within is entirely my own. I confirm that:

This doctoral thesis and the work therein was conducted under the supervision of Professor Peter Komisarczuk, Professor Kenny Paterson, and Professor Lorenzo Cavallaro. The work presented in this thesis is the result of original research carried out by myself, or in collaboration with others, while enrolled as a candidate for the degree of Doctor of Philosophy in Cyber Security at the Information Security Group, Royal Holloway, University of London. This work has not been submitted for any other degree or award at another university or educational establishment.

Simon Bell
20th March 2020

Abstract

Phishing and malware attacks continue to plague the digital world; wreaking havoc on individuals, businesses, and governments worldwide. Attacks often target popular platforms, such as Twitter: a microblogging social networking service with over 330 million active monthly users, posting more than 500 million daily tweets.

This thesis explores how well-protected Twitter users are from phishing and malware attacks. We take an empirical, data-driven approach to investigate the effectiveness of Twitter's cybercrime defence system at *time-of-tweet* and *time-of-click*. We create ***Phish**alytics*: our measurement infrastructure that collects and analyses large-scale data sets. Our data feeds include Twitter's Stream API, Bitly's Clicks API, and 3 popular blacklists: Google Safe Browsing, PhishTank, and OpenPhish. We improve internet measurement studies by addressing soundness and limitations of existing work. Our studies include characterising URL blacklists, investigating blacklist delays, and examining Twitter's URL shortener (*t.co*). We aim to better enable policymakers, technology designers, and researchers to strengthen online user security.

We provide empirical evidence highlighting the state, and scale, of cybercrime on Twitter. Key findings show over 10,000 phishing and malware URLs – publicly tweeted to more than 131 million Twitter accounts – received over 1.6 million clicks from Twitter users. Twitter's *time-of-click* defence system blocks only 12% of blacklisted URLs and web browsers miss up to 62% of non-blacklisted phishing websites. We recommend Twitter users ensure their risk appetite aligns with their cybercrime defence strategy. Furthermore, blacklists do not offer absolute protection and cybercriminals can exploit uptake delays.

Our findings suggest more can be done to strengthen Twitter's phishing and malware defence system and improve user security. However, measuring and evaluating effectiveness is complex and non-trivial. We discuss the importance of soundness, the significance of measurement study reproducibility, and the challenges of measuring an ever-changing landscape.

Acknowledgements

Firstly, I would like to acknowledge the UK EPSRC (grant EP/K035584/1) for funding the Centre for Doctoral Training in Cyber Security, at Royal Holloway, University of London. This has been a fascinating, challenging, and rewarding experience for which I am sincerely grateful.

Thank you to my supervisors: Peter Komisarczuk, Kenny Paterson, and Lorenzo Cavallaro, for guiding me through the challenges of research. Your advice and encouragement along this journey has been invaluable.

I would like to thank Twitter, Google, OpenPhish, PhishTank, and Bitly, for providing API feeds to access their data; without which this thesis would not have been possible.

Thank you to Martin Berger for originally inspiring and encouraging me to pursue a PhD in cyber security. Thank you to the numerous anonymous reviewers for their constructive criticism and feedback which ultimately contributed to our paper publications and awards.

I would like to thank everyone from the CDT cohort I was part of: Joanne, Ela, Andreas, Carlton, Greg, Giovani, Alex, and Suleman; and everyone from the S2lab, especially: Roberto, Feargus, Dusan, Sharad, Jason, Jordy, and James; it has been an extraordinary journey. I would like to thank my friends, old and new, particularly Chris, Josh, and Dom, for being there for me – especially during tough times and encouraging me to keep going; the dunking society; and my fellow volunteers at LINKS and ChildLine.

I am exceptionally grateful to my family for their continued love and encouragement throughout this journey, especially: my mother, Jane; my sister, Sam; and my fellow train and running enthusiast – and nephew, Max.

Finally, I would like to say a very special thank you to my partner, Helen, for everything you do; your love, support, and understanding got me through this.

Publications, Code, & Data

- Winner of 2 research paper awards (for categories: Best Paper and Best Student Paper): BELL, S., AND KOMISARCZUK, P. Measuring the Effectiveness of Twitter's URL Shortener (*t.co*) at Protecting Users from Phishing and Malware Attacks. In *Proceedings of the Australasian Computer Science Week Multiconference* (2020) [39].

- BELL, S., AND KOMISARCZUK, P. An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank. In *Proceedings of the Australasian Computer Science Week Multiconference* (2020) [38].

- BELL, S., PATERSON, K., AND CAVALLARO, L. Catch Me (On Time) If You Can: Understanding the Effectiveness of Twitter URL Blacklists. *arXiv preprint arXiv:1912.02520* (2019) [40].

- ***Phish****alytics*: measurement infrastructure we built to collect and analyse large-scale datasets; including phishing and malware attacks on Twitter, blacklist characterisation, and phishing detection capabilities of web browsers. Design and implementation details in Chapter 4. Codebase available on GitHub:

  `https://github.com/sjbell/phishalytics`

- Measurement data: the measurement dataset we collected during our experiments (1.5TB) is available to other researchers. See Section 4.5.1: *Ethically Responsible Data Sharing* for further details.

Contents

List of Figures

List of Tables

1 Introduction

OUTLINE
This chapter presents an overview of the thesis. We provide motivation for our research and outline our aims, research questions, contributions, and scope. We also summarise our overall approach to research and highlight a number of ethical and professional considerations. Finally, we provide context to our research, and then outline the structure of this thesis.

1.1 Motivation

Phishing and malware attacks continue to plague the digital world. From credential theft – that brings misery to individuals – to large-scale, coordinated attacks that wreak havoc on organisations and governments worldwide. Cyber attacks reach their victims by propagating through various channels and networks, such as: email, social media, and blogs.

With more than 500 million daily tweets from over 330 million active users [281], Twitter is an attractive target for cybercriminals. Twitter has come under increasing pressure to protect its users from cyberattacks. In 2010, Twitter settled a case with the US Federal Trade Commission (FTC) in which Twitter agreed to strengthen its security and carry out an independently assessed bi-annual information security audit for 10 years [86, 87, 88]. Cyberattack techniques continually evolve to avoid detection. Therefore, it is crucial to maintain effective cyber defence systems to protect internet users against phishing and malware attacks. We do this by improving our understanding of not only the phishing and malware attacks themselves – but also the defence systems that keep us safe.

Measurement studies, and longitudinal studies, improve our understanding of how effective certain defence systems are by observing specific

variables over periods of time. Results from such studies – including evaluations of effectiveness – contribute towards strengthening our defence systems and inform policymakers, researchers, technology designers, and other key actors. However, the effectiveness of studies' contribution to improving user security depends on the soundness (defined in Section 2.10) of the measurements and conclusions.

This thesis explores how well-protected Twitter users are from phishing and malware attacks. We take an empirical, data-driven approach to investigate the effectiveness of Twitter's phishing and malware defence system at *time-of-post* and *time-of-click*. As part of our research we analyse popular URL blacklists, examine the impact of delays on user security, and explore what additional protection web browsers can provide.

After defining what we mean by *effectiveness*, including a framework for measuring effectiveness (Section 2.16: *Effectiveness*), we identify limitations of existing measurement studies (Section 3.10: *Literature Review*). Our research contributes various techniques to address soundness and improve the accuracy of numerous effectiveness measurement metrics, including: blacklist delays, attack exposure (views and clicks), and volume of blacklisted URLs posted to Twitter.

Our contributions (Section 1.4: *Contributions*) also include the design and implementation of ***Phish**alytics* (Chapter 4: *Design & Implementation*); the measurement infrastructure on which we conduct our novel measurement studies and address methodological limitations of existing literature. We also improve our understanding of how the landscape we are measuring has changed over time by repeating outdated measurement studies.

We begin our investigation with an examination of 3 popular phishing and malware blacklists: Google Safe Browsing (GSB), PhishTank (PT) and OpenPhish (OP) (Chapter 5: *Blacklist Analysis Study*). This examination furthers our understanding of the 3 blacklists and provides context for our later studies. We explore key areas such as the uptake, dropout, typical lifetimes, and consider the overlap of URLs in these blacklists.

We then carry out a longitudinal measurement study (Chapter 6: *Time-of-Post Twitter Study*) to investigate Twitter's use of blacklists, delay times of these blacklists, and potential implications on user safety; analysing how many Twitter users view and click on phishing and malware attacks.

Following this, we want to understand what additional protection may be available to Twitter users that have been exposed to phishing attacks through the social network. We do this by investigating how effective web browsers' built-in phishing defence mechanisms are at protecting users from attacks (Chapter 7: *Web Browser Phishing Detection*).

Finally, we perform a longitudinal study (Chapter 8: *Time-of-Click Twitter Study*) to investigate how effective Twitter's URL shortener (*t.co*) is at protecting Twitter users from phishing and malware attacks at *time-of-click*

– and compare this to Twitter's use of blacklists and our previous study's results.

Internet measurement studies are easy to do poorly and difficult to do well [222]. The soundness of measurements – and the resulting conclusions – can be affected by various aspects such as methodology, infrastructure, and ground truth. Therefore, it is important that measurement studies address soundness, provide details of their methodology, and share the resulting data. This also improves the reproducibility of studies (see definition in Section 2.1.1: *Reproducibility*).

The internet measurement research community does not have a good culture of reproducing results [246, 26, 245, 303] and there are limited public repositories available [11, 10, 248]. Therefore, as part of our research aims (Section 1.2: *Aims*), we strengthen our measurement studies' contribution to the internet measurement research community. We share our methodology (Section 4.7: *Methodology Overview*), technical implementation details (Section 4.8: *Infrastructure & Implementation*), and resulting dataset (Section 4.5.1: *Ethically Responsible Data Sharing*) with the internet measurement community to improve reproducibility, address soundness, and aid future research.

1.2 Aims

Our primary aim is to investigate the effectiveness of Twitter's phishing and malware defence system. We do this by examining Twitter's use of blacklists and URL shortening service, *t.co*. As part of our investigation we explore the characteristics of specific blacklists, the impact of blacklist delay times on user security, we examine to what extent Twitter might be relying on web browsers' built-in security, and we analyse the effectiveness of web browsers' security.

We also define a set of aims to address soundness and improve measurement studies, thereby strengthening our measurement studies' contribution to the internet measurement research community. We aim to improve the quality, quantity, and analysis of data available to the research community for evaluating the effectiveness of Twitter's phishing and malware defence system. We present the details of our methodology and technical implementation to address soundness and improve our studies' repeatability, reproducibility, and replicability (defined in Section 2.1.1). We present our results in a format that can be used and interpreted by our intended audience (policymakers, researchers, technology designers, etc). Finally, we aim to contribute to the current understanding of how to collect and analyse internet measurements – specifically relating to phishing and malware attacks on Twitter – and to give insight into how the internet behaves.

1.3 Research Questions

Our core research questions are:

1. How effective is Twitter at protecting its users from phishing and malware URL attacks?
2. How effective are blacklists at helping Twitter to protect its users from phishing and malware attacks?
3. What do popular phishing and malware blacklists consist of? Uptake, dropout, typical lifetimes, and overlap of URLs in these blacklists
4. What is the lifetime of a URL in a phishing/malware blacklist?
5. Are blacklists affected by delays – and, if so, what impact do blacklist delays have on user security?
6. How effective is Twitter's URL shortener, *t.co*, at protecting Twitter users from phishing and malware attacks?
7. What is the comparison between Twitter's use of blacklists and Twitter's URL blocking (via *t.co*) at protecting users from phishing and malware attacks?
8. To explore the impact of phishing and malware attacks on Twitter: how many Twitter users click on publicly tweeted phishing or malware URLs that have been blacklisted?

1.4 Contributions

Our thesis makes contributions in 4 main areas:

1. Improve internet measurement studies by introducing new metrics and methodology; addressing soundness and limitations of existing work; strengthening current understanding of how to collect and analyse internet measurements of cybercrime on Twitter.
2. Measurement infrastructure (full codebase), methodology, technical implementation details, and resulting data set from our longitudinal studies; aids reproducibility and internet measurement community.
3. Provide empirical evidence, from our studies, to determine how effective Twitter's defence system is at protecting users from phishing and malware attacks. Our measurement *snapshots* also contribute towards future research by providing benchmarks for effectiveness.
4. Improve current understanding of phishing and malware ground truth by characterising and analysing 3 popular blacklists: GSB, OP, and PT.

Table 1.1 summarises the contributions of the thesis.

Section 2.16	Expand the definition of *effectiveness* – for measuring Twitter's cybercrime defence system – to include new and improved metrics; addressing soundness and limitations of existing work.
Section 2.16.2	Define an effectiveness evaluation framework.
Chapter 4	Design and implement ***Phish**alytics*: novel measurement infrastructure to conduct longitudinal measurement studies.
Section 4.5.1	Measurement data (1.5TB); available to help future research.
Section 4.7	Methodology to improve current understanding of how to collect and analyse internet measurements.
Section 4.8	**Phish**alytics technical implementation details and full codebase [37] to aid reproducibility.
Chapter 5	Novel characterisation and analysis of 3 popular phishing blacklists; including comprehensiveness and typical URL uptake, dropout, lifetime, and overlap [38].
Chapter 6	Novel, fine-grained and in-depth study into the effectiveness of blacklists at protecting Twitter users from phishing and malware attacks; *time-of-tweet* defence study [40].
Chapter 6	Novel evidence to suggest Twitter may no longer be using the GSB blacklist to protect users.
Section 6.5	Contemporary analysis of Twitter's phishing and malware defence system at time of tweet.
Section 6.5	Improve internet measurements by including URL redirection chain(s); addressing soundness and methodological limitations of existing literature.
Section 6.5.2	Improve internet measurements by introducing specialised blacklists: GSB, PT, and OP; addressing soundness and methodological limitations of existing literature.
Section 6.5.3	Improve measurement accuracy of tweeted URL blacklist delay times; addressing soundness and methodological limitations of existing work.
Section 6.5.4	Novel implementation of historical tweet context in methodology to increase accuracy of time of first tweet.
Section 6.5.5	Improve measurement accuracy of blacklisted URL clicks by defining timeframe and referrer; addressing methodological limitations of existing literature.
Chapter 7	Update and improve the accuracy of existing research into the effectiveness of web browser phishing detection; test suite comprising multiple operating systems; categorisation of URL blacklist status at time of test – to determine heuristic detection; web browser warning effectiveness framework.
Chapter 8	Novel study into the effectiveness of Twitter's phishing and malware defence system at *time-of-click*; Twitter's URL shortener (t.co) [39].

Table 1.1: Contributions of the thesis.

1.5 Scope

Undertaking longitudinal measurement studies to analyse Twitter's phishing and malware defence system is a considerable aim. Therefore, the scope of our research shall focuses on Twitter's use of *time-of-post* (or *time-of-tweet*) blacklisting and *time-of-click* filtering.

Our investigation will focus predominantly on phishing attacks; with a subsidiary exploration of malware attacks. Throughout this thesis our definition of the word "malicious" is: *intending to cause damage or steal private information from a computer system; including both phishing and malware attacks*.

We focus on 3 popular phishing and malware blacklists because these blacklists are prevalent in current popular web browsers (Section 2.6). Our studies are focused on the measurements of empirical data, such as blacklist delay times, number of URLs appearing in blacklists, number of users exposed to blacklisted URLs, etc.

We do not analyse the underlying code or deployment of specific phishing or malware attacks, since this is beyond the research scope of our thesis.

1.6 Approach

We take an empirical, data-driven approach to investigate cybercrime on Twitter by collecting and analysing large-scale datasets from a number of sources. From this real-world vantage point we explore cybercrime on Twitter vicariously; without compromising the safety of our team or Twitter's users.

Our approach to carry out our investigation consists of various measurement studies. Based on our research questions, we design a number of experiments to gather and analyse the required data. We build an infrastructure to collect and analyse this data. Through this infrastructure and our methodology we use a data-driven approach to address our research aims and questions.

1.7 Ethics

Ethics is defined as:

> *"A system of moral principles; the rules of conduct recognised in respect to a particular class of human actions or a particular group, culture, etc.; that branch of philosophy dealing with values relating to human conduct, with respect to the rightness and*

> *wrongness of certain actions and to the goodness and badness of the motives and ends of such actions."* [1]

When conducting research, it is important to define a set of ethical standards and moral principles that shall be adhered to. This thesis shall adhere to the Principles outlined by the Engineering and Physical Sciences Research Council (EPSRC) [77]; Guidelines on Research Governance, Research Ethics and Good Research Practice outlined by Royal Holloway, University of London [242]; and the Code of Conduct outlined by the British Computing Society (BCS) [48].

This thesis shall be open and honest about its methodology, experimental set-ups, and data collection processes. Comprehensive and detailed write-ups about these are found within this thesis, with specific details about the methodological setup in Chapter 4: *Design & Implementation*. One of the core aims of presenting these comprehensive write-ups is that our experiments shall be verifiable, reproducible, and replicable.

When conducting research into cyber security it is important to understand and be aware of the relevant ethical considerations. This thesis leverages access to a number of publicly available data sources in order to collect and analyse data. This data is accessed in accordance with the service providers' terms and conditions. Some of this data has been created by humans – such as tweets that have been publicly broadcast on the social network Twitter – and provided to us by Twitter. Therefore care has been taken to ensure that we process and analyse this data objectively, with well-researched and planned methodology, and in accordance with the professional considerations detailed in the next section. This thesis continues to explores a number of ethical considerations in more detail in Section 3.9.

1.8 Professional Considerations

A significant amount of research carried out for this thesis involves measuring cybercrime on the social networking platform Twitter. To do this, a number of large datasets are collected, stored, and processed, and a measurement infrastructure is designed, tested, and built in order to carry out measurement experiments. The subject of measuring cybercrime on a public social network can be a controversial one, therefore this section aims to address some of the main professional considerations of this project. The BCS [47] outlines a Code of Conduct [48] that should be adhered to when carrying out a project. The 4 core parts of the BCS Code of Conduct: *Professional Competence and Integrity; Duty to Relevant Authority; Duty to*

[1] https://www.dictionary.com/browse/ethics

the Profession; and Responsible Disclosure are applied specifically to our thesis in this section. All work carried out for this thesis shall adhere to the BCS Code of Conduct.

1.8.1 Public Interest

The measurement infrastructure created for this thesis may at times use third party software or libraries. In such cases all relevant third parties shall be referenced and credit given. With regard to Section 1(d) of the BCS Code of Conduct: this project is aimed primarily at the computer science and information security sectors. However, the outcomes of this project are not exclusive to these sector and may benefit many other sectors that want to increase their computer security awareness.

1.8.2 Professional Competence and Integrity

This thesis, and its research projects detailed herein, are being carried out as part of a university post-graduate research degree for the qualification of Doctor of Philosophy in Cyber Security. The author holds a BCS accredited Bachelor of Science (BSc) first-class honours degree qualification in Computer Science and has gained the relevant competencies and experience to build a measurement infrastructure. Ongoing training has been provided by the university throughout this project to continually improve the author's skills and competencies in key areas of cyber security research such as planning, conducting, presenting, ethics, standards, technical developments, etc.

As part of this thesis, significant research has been carried out to explore existing studies that relate to this topic along with an analysis of gaps in this existing pool of knowledge. This existing pool of knowledge provides a standard for some of the methodology and experimental setups carried out in this thesis. With regard to Section 2(e) of the BCS Code of Conduct: this thesis respects and values alternative viewpoints and, seeks, accepts and offers honest criticisms of work. This thesis contributes to the research community: comprehensively-tested and peer-reviewed novel methodology and measurement frameworks to enhance existing knowledge and to further the research field.

1.8.3 Duty to Relevant Authority

This thesis has been carried out with due care and diligence in accordance with the requirements of Royal Holloway, University of London, all relevant laws, and the terms and conditions set out by the various service providers that provide data for this thesis. Advice has been sought and permission

granted to carry out all work related to this thesis. All personal information related to this thesis shall remain confidential and not be disclosed unless required by law such as under the Regulation of Investigatory Powers Act 2000 [121].

1.8.4 Duty to the Profession

This project has been carried out to a high standard and in accordance to the standards set out in the BCS Code of Conduct along with high standards and professional cyber security research.

1.8.5 Responsible Disclosure

The measurement infrastructure that has been designed and built as part of this thesis uses various software, including a number of libraries. Any bugs or vulnerabilities discovered shall be reported to the relevant software vendors – as part of responsible disclosure guidelines [252].

1.9 Context

This section provides context to our research area. We begin with an overview of cybercrime, explore relevant legislation relating to phishing and malware attacks, present conviction rates for prosecutions under the Computer Misuse Act 1990 (CMA), and examine complexities of jurisdiction and cybercrime attribution. We then explore the prevalence of phishing and malware attacks by summarising data from numerous sources. Finally, we examine web browser usage statistics to understand historical trends and contemporary figures.

1.9.1 Cybercrime

Cybercrime is an umbrella term that includes many types of crimes which either take place online or whereby technology plays a key role in the crime. The National Cyber Security Strategy [120] defines 2 closely linked, but distinct ranges of criminal activity within this umbrella term:

- **Cyber-dependant crimes:** crimes that can only be committed through a computer
- **cyber-enabled crimes:** traditional crimes which can be increased in scale or reach by the use of computers

Interpol [131] defines “pure cybercrime” as:

> *"crimes against computers and information systems, where the aim is to gain unauthorised access to a device or deny access to a legitimate user".*

Many traditional forms of crime have evolved to leverage modern technologies and the internet in order to maximises criminals' profits in the shortest time. Many of these "cyber-enabled" crimes are not new (e.g. theft, fraud, illegal gambling, sale of fake goods, etc) but have adapted to this new, and constantly changing, digital environment.

The National Crime Agency (NCA) lists the 4 most common cyber threats [202] as:

- **Hacking**: including of social media and email passwords
- **Phishing**: bogus communications asking for security information and personal details
- **Malicious software (i.e. malware)**: including ransomware through which criminals hijack files and hold them to ransom
- **Distributed denial of service (DDoS)**: attacks against websites; often accompanied by extortion

The main UK legislation relating to offences or attacks against computer systems, such as hacking or denial of service, is the Computer Misuse Act 1990 (CMA) [118]. Sections 1 to 3 of the CMA includes 3 criminal offences, each punishable by 12 months' imprisonment or an unlimited fine:

1. Causing a computer to perform a function with intent to secure unauthorised access to computer material
2. Unauthorised access with intent to commit or facilitate commission of further offence
3. Unauthorised acts with intent to impair the operation of a computer

Additionally, section 3(1) of the Investigatory Powers Act 2016 [121] (IPA), which came into force on 27 June 2018, makes it an offence to intentionally intercept a communication in the course of its transmission – whether this be by public or private telecommunication system or a public postal service.

Offences under sections 170 to 173 of the Data Protection Act 2018 (DPA) [119] may be committed alongside cyber-dependant crimes. This may occur, for example, when phishing attacks share personal information about their victims such as usernames and passwords which are obtained as a result of the attack. Many phishing attacks may additionally fall under the Fraud Act 2006 due the underlying dishonesty and deception of these attacks. Phishing attacks may also commit offences under the Theft Act

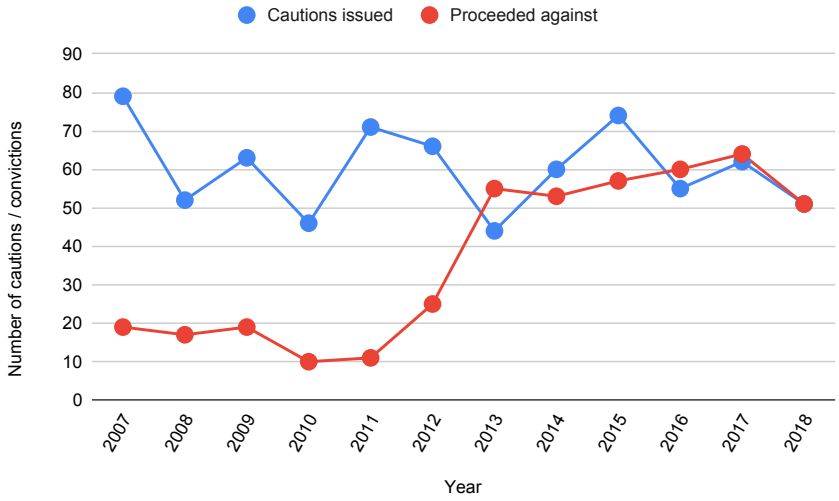


Figure 1.1: Computer Misuse Act 1990 Cautions and Charges, 2007-2018. Graph data source: Ministry of Justice [185, 186].

1968, Theft Act 1978, Forgery and Counterfeiting Act 1981, and Proceeds of Crime Act 2002 (POCA).

However, it is important to note that if an offender accesses data, reads it, then uses the information for his/her own purposes, then this is not an offence contrary to the Theft Act. Data (i.e. information obtained from computer storage) does not fall within the definition of property in section 4 of the Theft Act 1968 and cannot be stolen (Oxford v Moss (1979) 68 Cr App R183 DC). However, such an act would likely constitute an offence under section 1(1) of the CMA. Additionally, if the act were motivated by the intent to commit or facilitate the commission of further offences, it would constitute an offence contrary to section 2(1) of the CMA.

If a number of criminals are involved in a phishing attack then this may fall under section 1 of the Criminal Law Act 1977, or common law conspiracy to defraud. The act of setting up false social networking accounts or aliases could also amount to criminal offences under the Fraud Act 2006 if there was a financial gain, as under section 8 possession or making or supplying articles for use in frauds includes any program or data held in electronic form [122].

A 2019 investigation by *The Register* [92] found that 90% of hacking prosecutions in the UK in 2018 resulted in convictions and that 16% of offenders were sent to prison. Data obtained from HM Courts and Tribunals Service [185, 186] showed 441 total prosecutions under the CMA between 2007 and 2018, as shown in Figure 1.1. 79 people – 24% of the total prosecuted in that period – were found not guilty at court or otherwise had their cases halted. Of the guilty, 16 per cent were given immediate

custodial sentences. That number increases to 45% if suspended sentences are included. In 2018, 51 cautions were issued as well as 51 criminal court cases. In those 51 prosecutions, 45 defendants were found guilty, a rate of around 90%. The 2013 increase in prosecutions may relate to when Home Secretary Theresa May withdrew her extradition order against hacker Gary McKinnon [34]. This may have signalled a turning point when the UK decided it would rather prosecute at home than extradite suspects. The most common fines ranged from between £300 and £500 – with one fine over £10,000. The most frequent prison sentence lengths ranged between 6 to 9 months and 18 to 24 months. Current UK sentencing laws automatically halve prison sentences in favour of release on licence, with release from prison usually being a bit earlier again than half the headline figure [57].

Jurisdiction of cybercrime difficult and complex due to the borderless nature of cybercrime. For example, the act of a cybercrime may be carried out from a country that does not recognise said act as a crime. However, the recipient victims and/or consequences of such an act may be located in a country that *does* recognise the act as a crime. Additionally, different countries (e.g. UK/US) may have differing laws around cybercrime which can result in prosecution challenges and extradition complications (such as Gary McKinnon [137], Elliott Gunton [182], etc).

Cybercrime attribution is often a difficult and complex task. This is partly due to the distinction between nation states and criminal groups becoming increasingly blurred, and also because it can be relatively easy for criminals to maintain true anonymity online by leveraging services such as the dark web to cover their tracks. The FBI [80] explains that those carrying out cyber attacks range from *"computer geeks looking for bragging rights, to businesses trying to gain an upper hand in the marketplace by hacking competitor websites, from rings of criminals wanting to steal personal information and sell it on black markets, to spies and terrorists looking to rob our nation of vital information or launch cyber strikes"*.

As we have seen, there are various laws which cover numerous cyber crimes. We have also seen how a single phishing or malware attack may breach a number of laws at the same time. Although there are many laws in place to protect against cyber attacks, such as the CMA, some argue that these laws are outdated and can do more harm to ethically motivated cyber defenders, security researchers and journalists than actually preventing criminals carrying out cyber attacks [218].

1.9.2 Phishing & Malware in Numbers

This section aims to provide an overview of the current state of phishing and malware attacks in terms of number of recorded attacks. It can be difficult to pinpoint the exact number of attacks at any given time, due

to most attacks going unreported. However, a number of organisations have a good vantage point to observe and quantify phishing and malware attacks that occur in the wild. Therefore we include statistics from some of these organisations in an attempt to build a picture of the current state of phishing attacks.

Anti-Phishing Working Group

The Anti-Phishing Working Group [17] (APWG) is an international organisation dedicated to the prevention and reduction of phishing activity. APWG produce a number of trend reports [18] each year to highlight the current threat level of phishing activity.

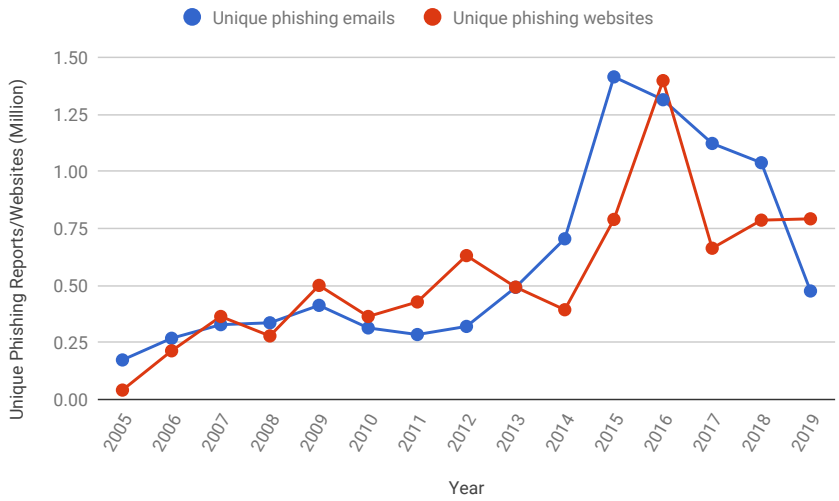


Figure 1.2: Number of unique phishing e-mail reports (campaigns) received from consumers and number of unique phishing web sites detected between 2005 and 2019 inclusive. Graph data source: APWG [18].

Figure 1.2 shows the total number of unique phishing emails, along with the total number of unique phishing websites, reported to APWG per year between 2005 and 2019. We see that, in 2005, there were a total of 173,063 unique phishing emails reported to APWG, which consisted of 40,947 unique phishing websites. In 2018 there were a total of 475,369 unique phishing emails, consisting of 791,766 unique phishing websites.

We also see a sharp increase in the number of phishing emails, rising from 704,178 in 2014 to 1,413,978 in 2015. Although the number of phishing emails reported by APWG has been decreasing since 2015, the numbers still remain significantly higher than prior to 2015. We also see a sharp rise in the number of unique phishing websites, from 393,160 in 2014 to 789,068 in 2015 and again to 1,397,553 in 2016. This number does fall

back to 662,795 in 2017 but continues its upward trend to 785,920 in 2018 and 791,766 in 2019. The graph shows a general upwards trend, illustrating that the number of phishing emails and websites has been increasing since APWG started to monitor the attacks in 2005.

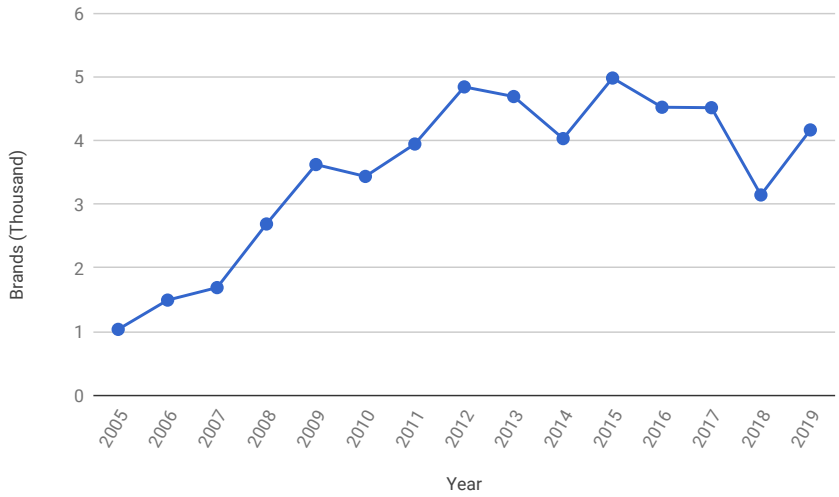


Figure 1.3: Number of brands targeted by phishing campaigns between 2005 and 2019 inclusive. Graph data source: APWG [18].

APWG also provides statistics for the number of brands that are targeted each year by phishing campaigns. Figure 1.3 shows the total number of brands targeted by phishing campaigns per year, detected by APWG, between 2005 and 2019. In 2005 there were 1,037 brands targeted. In 2019 there were 4,165 brands targeted. Again, we see an upwards trend showing that the number of brands being targeted by phishing campaigns continues to increase. The APWG Q4 2019 Trends Report [19] shows that the most targeted industry sectors of phishing emails were SaaS/Webmail (36%), Payment (27%), and Financial Institution (16%).

Figure 1.4 shows the percentage of phishing attacks hosted on HTTPS websites between Q1 2016 and Q4 2020. We see that, in Q4 2020, 84% of phishing sites were using SSL certificates (HTTPS) in an attempt to further convince users that they are legitimate websites. Up from 58% in Q2 2019.

Overall, these statistics from APWG show that the number of phishing attacks continues to grow. Therefore research into phishing attacks plays an important role in understanding and reducing the impact of such threats.

Google Safe Browsing Transparency Report

Google Safe Browsing provides a Transparency Report [106] which presents an overview of websites listed in the Google Safe Browsing blacklist (the

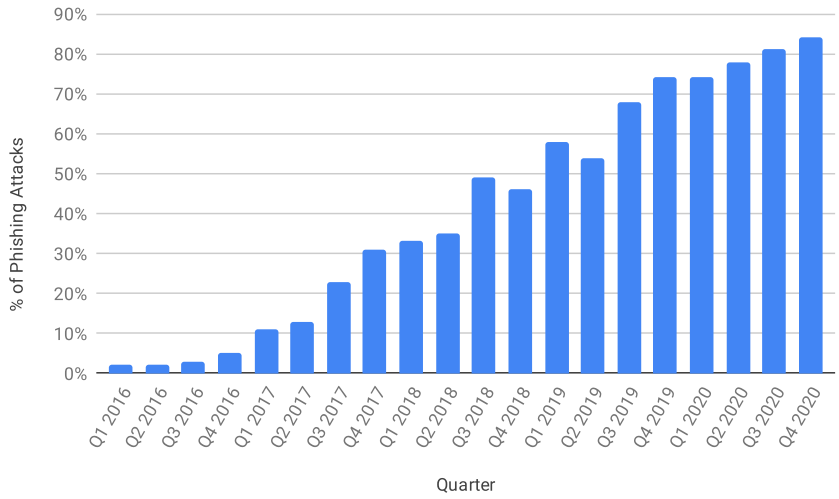


Figure 1.4: Percentage of Phishing Attacks Hosted on HTTPS between Q1 2016 and Q4 2020. Graph data source: APWG [18].

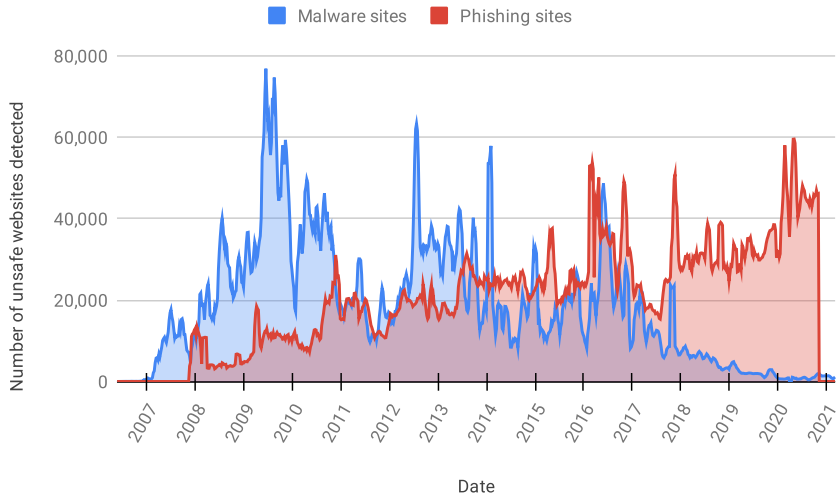


Figure 1.5: Unsafe websites detected per week, 21st May 2006 to 16th March 2021. Major ticks on the x-axis represent 1st January for that year. Graph data source: Google Safe Browsing Transparency Report [106].

blacklist itself is discussed in further detail in Section 2.6.1: *Google Safe Browsing*).

Figure 1.5 shows the total number of unsafe websites detected per week from 21st May 2006 to 16th March 2021. Overall, we see a general upwards trend in the number of phishing websites detected per week, with some noticeable peaks of 52,924 to 53,732 in February to March 2016; 49,156 in

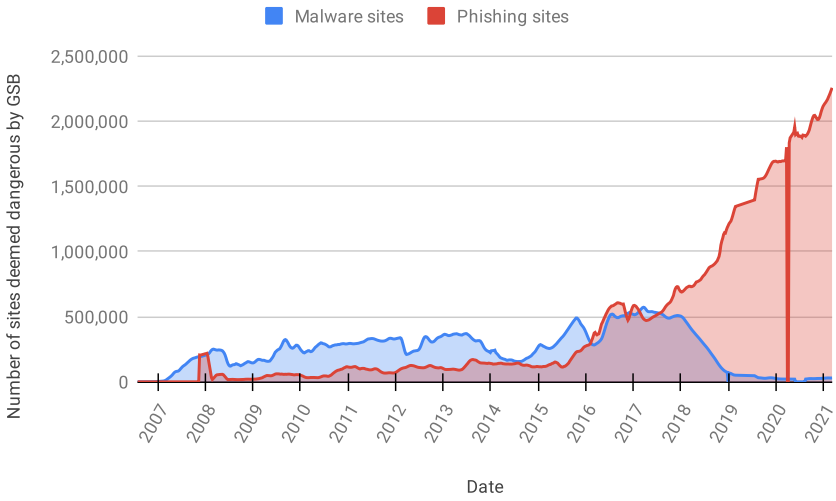


Figure 1.6: Number of sites deemed dangerous by Safe Browsing, 21st May 2006 to 16th March 2021. Major ticks on the x-axis represent 1st January for that year. Graph data source: Google Safe Browsing Transparency Report [106].

October 2016; and 50,905 in November 2017. Interestingly, the number of malware sites has decreased from 48,641 in May 2016 to 2,806 in October 2019.

Figure 1.6 shows the number of sites deemed dangerous by Safe Browsing from 21st May 2006 to 16th March 2021. Again, we see the number of phishing websites continues to grow, hitting 1,592,706 in October 2019. Whereas the number of malware websites has reduced, from a peak of 572,377 in March 2017 to 26,942 in October 2019.

It is not clear why there has been a fall in the number of malware websites detected by GSB – Google do not provide an explanation for this decrease. However, we do see in these graphs, as in the APWG graphs, that the number of phishing attacks continues to increase over time.

Google does provide explanations for some visible changes to Safe Browsing, such as in September 2012 when one of their "third-party phishing feeds that provides Safe Browsing with data is temporarily removed because of concerns with too many false positives". Google also states that "Phishing protection in Safe Browsing obtains some of its data from third-party feeds. However, these feeds must meet stringent standards to ensure very low false positive rates." [2]

Interestingly, Google states that, as of February 2021:

[2] https://support.google.com/transparencyreport/answer/7381518?hl=en-GB&ref_topic=7381457

> *"Data charts for **Unsafe websites detected per week** and **How we identify malware** stopped being updated and were archived."* [3]

Google's Safe Browsing Transparency Report [106] archived its malware section on the 8th April 2020, stating:

> *"Due to an evolving threat landscape and shift in malware behaviour, the data in these charts is no longer representative."* [4]

The data shown in our Figures 1.5 and 1.6 was recorded on the 17th March 2021 from the public GSB Transparency Report API endpoints [104] and [103], respectively. Data from [104] stopped reporting phishing sites from the 8th November 2020 (as seen in Figure 1.5). The API data endpoint for *unsafe websites detected per week* [104] reported a peak of 188,831 phishing sites on 23rd February 2020 – 3 times the number of phishing detections ever recorded by GSB in a single day. However, the figure was subsequently altered by Google to the data seen in Figure 1.5.

AV-Test

AV-Test is an independent research institute for IT, based in Germany. They provide information on malware statistics, such as can be seen in Figure 1.7. In contrast to GSB's statistics, data provided by AV-Test shows that the total number of malware threats continues to rise year-on-year and has not decreased.

This section has explored data from number of different organisations that provide statistics on the number of phishing and malware attacks. This gives us a sense of how prevalent these attacks are and provides motivation for research into such attacks.

1.9.3 Web Browser Usage Statistics

As previously mentioned, this thesis leverages Google Safe Browsing as a source of *ground truth* for detecting phishing and malware websites. Part of the motivation for using GSB is its prevalence in popular web browsers: Chrome, Safari, Firefox, Opera, and Vivaldi. To this end, it is important to understand the usage statistics and market share of web browsers. We use two key metric providers as a source for these statistics: W3Counter and Statcounter.

Figure 1.8 shows the market share of the most popular web browsers between May 2007 and January 2020, provided by W3Counter. In May

[3] https://support.google.com/transparencyreport/answer/7381518

[4] https://transparencyreport.google.com/archive/safe-browsing/malware

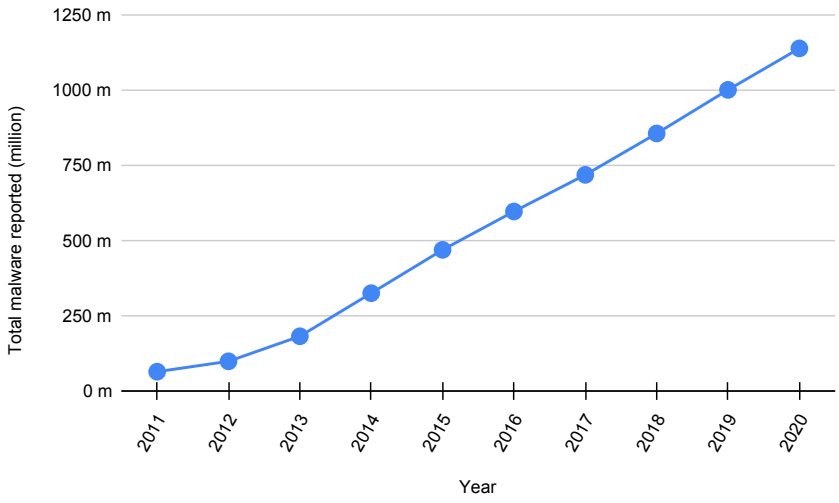


Figure 1.7: Total Malware reported between 2011 and 2020. Graph data source: AV-test [24].

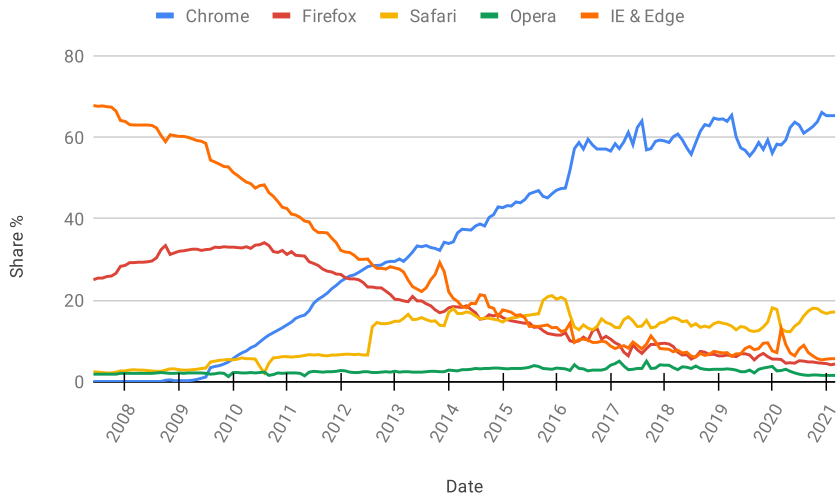


Figure 1.8: Web Browser Usage Trends from May 2007 to February 2021. Major ticks on the x-axis represent 1st January for that year. Graph data source: W3Counter [287].

2007, Microsoft's Internet Explorer (IE) was the most popular web browser with market share of 67.6%, followed by Firefox at 25%, Safari at 2.4%, and Opera at 1.8%. In January 2020, Google's Chrome was the most popular web browser with a market share of 58.2%, followed by Safari at 17.7%, Microsoft's Internet Explorer (Edge) at 7.1%, Firefox at 5.5%, and Opera at 2.6%.

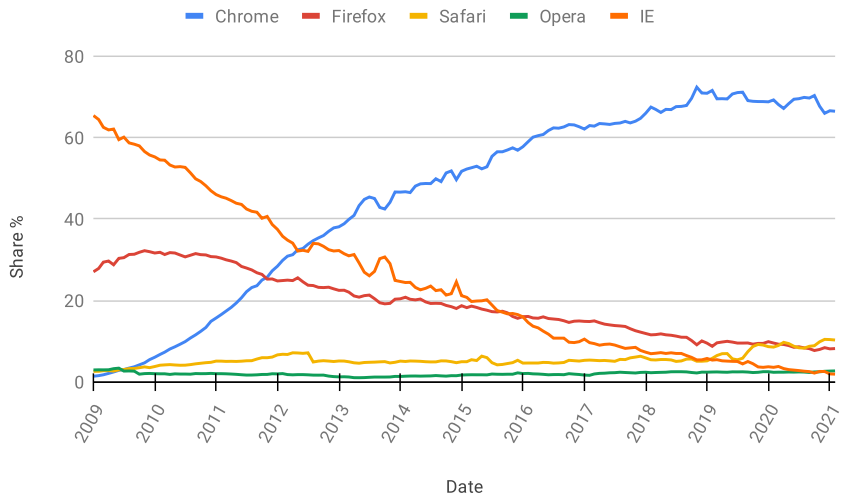


Figure 1.9: Desktop Browser Market Share Worldwide, January 2009 to February 2021. Major ticks on the x-axis represent 1st January for that year. Graph data source: Stat Counter [257].

Figure 1.9 shows the market share of the most popular web browsers between January 2009 and January 2020 provided by Statcounter. In 2009, Microsoft's Internet Explorer was the most popular browser with a 65.41% market share, followed by Mozilla's Firefox at 27.03%, Opera at 2.92%, Safari at 2.57%, and Chrome at 1.38%. In January 2020, Google's Chrome had become the most popular browser with 68.78% market share, followed by Firefox at 9.87%, Safari at 8.64%, IE at 3.7%, and Opera a 2.49% market share.

As we can see, the two data providers paint a similar picture: since 2013, Chrome has overtaken Internet Explorer to become the most popular web browser, followed by Firefox, Safari, and Opera.

It is also interesting to understand how the operating systems that these web browsers run on have changed over time. Figure 1.10 shows the most popular operating systems between January 2009 and January 2020, as measured by Statcounter. We see that in 2009, Windows was the most popular operating system with a market share of 94.8%, followed by OS X (Apple) at 3.66%. In January 2020, Android was the most popular operating system with a market share of 39.67%, followed by Windows at 35.23%, iOS at 14.5%, and OS X at 7.71%. This trend shows how, since 2009, mobile operating systems (*i.e.* smart phones) have overtaken desktops to become the most popular form of computers. The mobile versions of Chrome, Safari, Firefox, Opera, and Vivaldi continue to use GSB for phishing and malware protection.

As we have seen, web browsers that employ GSB to defend against cyber

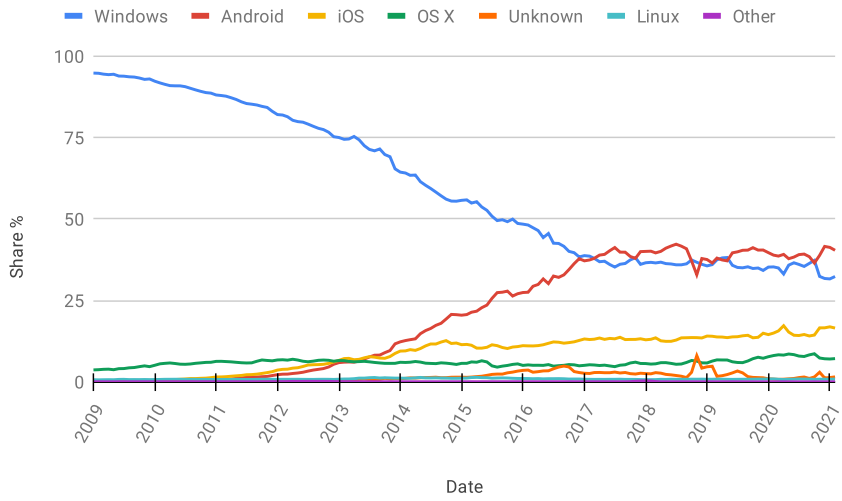


Figure 1.10: Operating System Market Share Worldwide, Jan 2009 to Jan 2020. Major ticks on the x-axis represent 1st January for that year. Graph data source: Stat Counter [258].

attacks (Chrome, Safari, Firefox, and Opera) are some of the most popular web browsers – used by people to browse the web. This makes GSB an important and highly relevant blacklist to study, since any findings from this research may impact these popular web browsers.

1.10 Thesis Structure

The rest of this thesis is structured as follows:

Chapter 2 - *Background*: defines key concepts and topics, including: measurement studies, phishing and malware attacks, blacklists, URL shorteners, a brief introduction to Twitter, defence against phishing & malware attacks, and APIs. We define *effectiveness*, our set of effectiveness evaluation metrics, and present our effectiveness evaluation framework.

Chapter 3 - *Related Literature*: split into 2 parts:

1. **Summarises existing literature:** to understand relevant background knowledge, explore various methodologies, justify our research methodology, and provide context to our research findings.
2. **Reviews existing studies:** to identify and address methodological limitations.

Chapter 4 - *Design & Implementation*: covers the design and implementation of our measurement system, called ***Phish**alytics*; including the architecture, interface, and specific parts of our system. We describe the experiments, methodology, and technical implementation of our measurement studies.

Chapter 5 - *Blacklist Analysis Study*: presents our study of popular blacklists: GSB, OP, and PT. This chapter is an edited version of our research paper: BELL, S., AND KOMISARCZUK, P. An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank. In *Proceedings of the Australasian Computer Science Week Multiconference* (2020).

Chapter 6 - *Time-of-Post Twitter Study*: presents our study into blacklists on Twitter and the impact of blacklist delays on user security. This chapter is an edited version of our research paper: BELL, S., PATERSON, K., AND CAVALLARO, L. Catch Me (On Time) If You Can: Understanding the Effectiveness of Twitter URL Blacklists. *arXiv preprint arXiv:1912.02520* (2019).

Chapter 7 - *Web Browser Phishing Detection*: presents our study on the effectiveness of popular web browsers' phishing detection technology. Our aim is to assess whether Twitter can rely on the web browsers to protect users from phishing attacks. This chapter is an edited version of our CDT summer project entitled: *Browsing for Phish: An Analysis of Web Browser Phishing Detection Technology* (2015).

Chapter 8 - *Time-of-Click Twitter Study*: presents our study on Twitter's URL shortener (*t.co*). This chapter is an edited version of our multi-award-winning research paper: BELL, S., AND KOMISARCZUK, P. Measuring the Effectiveness of Twitter's URL Shortener (*t.co*) at Protecting Users from Phishing and Malware Attacks. In *Proceedings of the Australasian Computer Science Week Multiconference* (2020).

Chapter 9 - *Discussion*: discusses our results from Chapters 5 to 8 and addresses key points raised in earlier chapters.

Chapter 10 - *Conclusion*: brings this thesis to a close; summarising our main findings and discussions.

2
Background

OUTLINE
This chapter defines key concepts and topics, including: measurement studies, phishing and malware attacks, blacklists, URL shorteners, a brief introduction to Twitter, defence against phishing & malware attacks, APIs. Towards the end of this chapter we present our definition of *effectiveness* and our effectiveness evaluation framework.

2.1 Measurement Studies

Measurement studies, and longitudinal studies, are types of research design commonly used in the fields of social science, health care, and economics that involve repeated observations of the same variables (e.g. people) over a specified period of time. Measurement studies provide a crucial role in establishing facts in a specified environment; a scientific way to determine the current state of something. These facts can contribute towards evidence-based decisions, advice, policies, technology designs, etc.

Internet measurement studies [221, 159, 65, 109, 51, 187] typically focus on observing specific aspects of computer networks to improve our understanding of the internet's structure and behaviour. Aspects of the internet that we might want to measure include: structure (e.g. topology, routing, proxies, wireless, etc), traffic (e.g. transport, end-to-end performance, etc), users and applications (e.g. WWW, DNS, social networks, etc), failures, and nefarious behaviour (e.g. pattern attacks, port scans, phishing, malware, etc).

In this thesis we conduct numerous internet measurement studies that focus on applications (specifically, the social network: Twitter), users, and nefarious behaviour. We measure empirical data (tweets, blacklisted URLs,

etc) to make observations about the state, and scale, of cybercrime on Twitter. In later chapters we will define our data sources for capturing empirical data, and our methodology for measuring the data and resulting observations.

Let us consider an example: a highly contagious computer virus is spreading across the internet; causing global havoc. The virus's main transmission vector is drive-by-download. You want to determine how prevalent the malware is on websites.

A measurement study could be conducted whereby all public-facing websites on the internet are checked for the specific malware strain. By visiting all public-facing websites, and checking them for the malware strain, we can start to understand how many websites are spreading the strain and what impact this may have (such as: number of users infected, transmission rate, ease of spread, etc).

In our example, the measurement study would define a methodology to ensure its research approach is scientific and accurate. This methodology would define which websites are to be visited (how do we visit all public-facing websites on the internet?) along with how we will check if the malware is present on each website (e.g. signature match, honey-client machine, etc). The results of said measurement study can contribute towards our understanding of how prevalent the virus is and what the impact may be.

Notice we say that the measurement study can **contribute** towards our understanding of the virus – the study might not provide a full or independent understanding of the virus. In reality, we probably cannot visit every single public-facing website on the internet. This could be due to any number of factors, such as websites availability during the measurement study, timing of malware appearance on each website, etc. Therefore, with these considerations in mind, we understand that the measurement study should provide a reasonable *snapshot* to help us understand the prevalence of the malware at the time the study was carried out.

It is important to assess the soundness (defined in Section 2.10) of measurement studies and the resulting conclusions. Soundness can be affected by various aspects such as methodology, infrastructure, and ground truth of the study. Measurement study observations may not be absolute; they might require context, understanding of background concepts and literature, etc. Table 2.1 lists some important considerations of common measurement study objectives.

In this thesis we conduct numerous measurement studies to assess the effectiveness of Twitter's phishing and malware defence system. We improve longitudinal studies by providing an effectiveness evaluation framework, discussing soundness, and addressing methodological limitations of existing work. As in our example, we will define the scope and purpose of our meas-

Objective	Consideration
Establish facts in a specified environment.	Facts must be taken in context of the scope and methodology of the study.
	How sound are the measurements and the study's conclusion(s)?
Provide a snapshot of the specified environment in relation to research aims.	Characteristics of environment, and the implications thereof, must be understood (e.g. pace of evolution).
	Perspective is important; two measurement studies of the same environment may produce different results and draw different conclusions depending on their metrics, observational position, aims, etc.
Contribute to decisions, advice, and policies.	Any conclusions drawn must understand the results in context of existing literature, related topics, general considerations of measurement studies, etc. This typically gives rise to a discussional approach to results and conclusions.
	Evaluate soundness of measurements and conclusions. Be aware of absolute claims, dichotomous thinking, and measurement fallacies (e.g. quantitative fallacy: conclusion based solely on quantitative observations – or metrics – that ignores all other observations, context, etc).

Table 2.1: Important considerations of common measurement study objectives.

urement studies (based on our research questions) and the methodology of our studies.

The results of our measurement studies can help us to understand how effective Twitter's phishing and malware defence system is – and how well protected Twitter users are from these attacks. The results of our studies are targeted at all audiences and may be of particular interest to researchers, policymakers, technology designers, etc.

It is important to define the aims of our measurement studies. For this, we take inspiration from a well-respected, specialist conference in our field: the Internet Measurement Conference (IMC). IMC is an annual conference focusing on Internet measurement and analysis, sponsored by ACM SIGCOMM and ACM SIGMETRICS in cooperation with USENIX. Papers at IMC aim to "contribute to the current understanding of how to collect

or analyse Internet measurements, or give insight into how the Internet behaves".

The core aims of our measurement studies are to:

- Contribute towards answering our research questions (defined in Section 1.3)
- Improve the quantity of data available
- Improve the quality of data available
- Improve the analysis of said data
- Present our methodology details and technical implementation to improve our studies' repeatability, reproducibility, and replicability
- Present results in a format that can be used and interpreted by our intended audience
- Contribute to the current understanding of how to collect and analyse internet measurements, and give insight into how the Internet behaves

2.1.1 Reproducibility

An important consideration when conducting any measurement study is to ensure that its results are reproducible [222, 246, 26, 245, 303]. This helps the research community and ensures results are verifiable. ACM defines the following 3 terms [3] relating to reproducing experimental results:

Repeatability

Same team, same experimental setup.

The measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat her own computation.

Reproducibility

Different team, same experimental setup.

The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.

Replicability

Different team, different experimental setup.

The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

2.2 Deception

Deception is the act of attempting to convince a person (or persons) into believing something that is not true. An attempt at deceptive behaviour has been successful when the recipient (of the deceiver) believes a given lie to be true. In the case of phishing, deception is used as a social engineering technique to manipulate the victim into unknowingly sharing their sensitive data with the attacker. Deception is typically seen in the offline world where human-to-human interactions occur. It is also in these face-to-face scenarios that deception detection techniques can be used [70].

In the digital world, techniques for deception detection such as facial cues and body language are more challenging to deploy. However, this new arena for deceptive behaviour brings with it a myriad of new detection techniques which can be used to identify the authenticity of the deceiver (such as IP addresses, cryptographic signatures and other auditing breadcrumbs).

2.3 Phishing

Phishing is a term used to describe the action of an attacker attempting to steal sensitive information from her victim in the digital world using social engineering techniques. This sensitive information can include usernames, passwords, addresses and credit card details. In the case of phishing, the attacker will masquerade as an official organisation – such as the victim's bank or online service providers such as Facebook, Google, Twitter, etc – in order to trick the user into giving away their sensitive information.

The term "phishing" originates in the 1990s on America Online (AOL), the number one provider of internet access at the time. A group of hackers, called the *warez* community, created an all-in-one hacking tool, called AOHell [66]. This tool contained many features such as: a credit card number generator, email bomber, instant messaging bomber, fake account generation, etc. The included algorithm to generate random credit card numbers was often used to create fraudulent AOL accounts. In 1995 AOL

stopped the random credit card generators, which caused the *warez* group to move onto alternative methods.

The AOHell tool included a number of "phishing" tools that allowed hackers to steal passwords and credit card details of AOL users through automated social engineering techniques. Hackers would use the platform's instant messenger system to contact users whilst posing as AOL employees. These messages would lure victims into verifying their accounts or confirming billing information. The tool would send messages, that appeared to come from AOL staff, to AOL users with messages such as:

> "Hi, this is AOL Customer Service. We're running a security check and need to verify your account. Please enter your username and password to continue."

Hackers would then use these compromised accounts to carry out spamming and fraudulent activities. These compromised accounts could also be used on the black market to exchange pirated software and other goods. It is believed that the AOHell tool contained the first recorded mention of the term "phishing" [239]. On January 2, 1996, a Usenet group dedicated to AOL used the term "phishing" to describe what the *warez* group was doing and to warn the AOL community about the attacks. AOL eventually included the term "phishing" in its emails and messaging software to warn users about these attacks [184].

The word "phishing" is a combination of the words "fishing" and "phreaking". Analogous to the sport of fishing, bait (in the form of emails, messages, websites etc) were used to reel in fish (sensitive data) from the sea of internet users. Phreaking was a term coined by John Draper (also known as Captain Crunch), a notorious telephone hacker from the 1970s, that was used to describe the act of hacking into telephone systems – the original form of hacking. Hackers that were fishing for user's sensitive data would often replace the letter "f" with "ph" to describe their actions as part of the hacking community.

Many hackers would use AOL chat rooms to discuss pirated software and trade stolen accounts. However, AOL quickly implemented an automated system to remove accounts that were found to be using certain keywords that could be linked to these activities. As a result, the hacking community would use the characters "<><" as a substitute for any of these keywords. Since the chat rooms consisted of large amounts of HTML, and the "<" and ">" characters feature predominantly in HTML code, the automated filters were unable to detect its use.

Since the days of AOL phishing, attackers have moved to other platforms – such as email, SMS, and social media – to lure victims in where it is much harder for the attackers to get caught. An example of a contemporary phishing attack might be an email pretending to be from Facebook asking

a user to verify their account. A URL contained within this email might direct the user to a spoof Facebook login page, whereby the user inputs their username and password.

With its humble roots in the *warez* and hacking community, by rebellious teenagers stealing passwords for fun, phishing has evolved into one of the most prevalent and costly computer security threats – affecting people, corporations, and governments worldwide.

2.3.1 Types of Phishing

The main types of phishing include:

- Vishing: voice phishing, phone alls
- Smishing: SMS phishing
- Search Engine Phishing: fake websites targeting specific keywords
- Spear Phishing: typically involves more sophisticated social engineering techniques and more detailed information about the victim than a regular phishing attack. An example of a spear phishing attack would be one that targets the central executive officer (CEO) of a company. The spear phishing attacker might know the CEO's name, address, bank provider and personal assistant's (PA) name. The attack could therefore masquerade as the CEOs PA in order to convince the CEO to send an amount of money to the attackers bank account. This attack may be more convincing to the victim, in terms of deception, because more information can be used to convince the victim that the lie is real.
- Whaling: similar to spear phishing, but even more targeted. Often aimed at high-level executives such as CEOs, CFOs, and COOs
- Pharming: cache poisoning against DNS to change IP address associated with a website to redirect to phishing website
- Evil Twin: bogus wireless access point to create fake WiFi
- Email Spoofing & Brand Impersonation: emails pretending to be from someone or an organisation
- Website / Brand Phishing URLs: websites which look legitimate, such as a Facebook login page. URLs often hidden/disguised e.g. via URL shortener, redirects, misspellings. Includes subdomain attack, pop-up phishing (login box pops up)
- HTTPS phishing: with the advent of easy-to-deploy TLS certificates, phishers will use HTTPS in an attempt to further deceive their victims

2.3.2 Phishing Websites

The main form of phishing that is addressed in this thesis is phishing websites. These are websites that are designed to look like an official organisation (such as a bank, social media website, auction website etc). Victims often arrive at these phishing websites via emails that use authoritative language and time sensitive techniques (such as "your account has been compromised, you must take action now to recover your data"). These initial emails and messages play a key role at "warming up" the victim and initiating the social engineering technique. These attacks are effective, not only because of the use of authoritative language but also because the websites themselves look convincing to users [72].

Phishing attacks on Twitter have been known to involve malicious users promoting tweets to lure in their victims by promising verification status on the app. Victims would then hand-over their Twitter username, password, phone number, and credit card information to these criminals [45].

2.4 Malware

Malware stands for malicious software and is a term used to describe any software that has nefarious intentions. These intentions can include attempting to disrupt, damage, or gain authorised access to a computer system, gathering sensitive information, etc. Malware is an umbrella term often used to describe the following types of programs:

- Viruses: programs that replicate by inserting their code into other programs, files or boot sectors of the hard drive often with the aim to corrupt or modify the target system
- Worms: programs that spread by infecting other computers usually via a computer network with the aim to harm the computer network often by consuming bandwidth. Unlike viruses, worms do not generally attach themselves to other programs or corrupt files on the target system
- Trojans: programs disguised to look like useful applications but actually have malicious intent such as causing loss or theft of data and possible system harm. Trojans are generally non-self-replicating. The term is derived from the ancient Anatolia story of the wooden horse used to trick the defenders of Troy
- Spyware: software that collects information that is often then sent or sold to a third party
- Ransomware: software that prevents its user from using the computer system by demanding some ransom be paid to the malware author.

An example of ransomware is cryptoviral extortion which encrypts a user's hard drive, preventing access to the entire system unless the ransom is paid

- Adware: software that displays adverts on the infected system without the user's consent. Not to be confused with advertising-supported software whereby the terms of the software state that adverts will be displayed and the user agrees to this
- Malvertising: using legitimate adverts or advertising networks to covertly deliver malware to its victims. Clicking on an advert may take its victim to a malicious website or install malware on their computer. Malware may be embedded in the advert itself end execute automatically without user interaction, know as "drive-by-downloads"
- Scareware: also known as rogueware, this software is designed to look like a genuine virus with the aim of tricking the user into paying for fake antivirus software to remove it
- Fileless malware: although not strictly a different category of malware, these types of malware programs exploit and spread in memory or through other "non-file" operating system objects such as scheduled tasks, APIs, registry keys, etc
- Hybrids: most modern malware is a combination of the above types, including parts of worms, trojans, viruses, etc. This include rootkits or stealth programs which attempt to modify the underlying operating system to gain root access, escalated privileges, take control, and hide from antivirus and antimalware detection programs
- Bots & Botnets: a computer that has been infected with malware so it can be controlled remotely by criminals. Once infected and controlled remotely, a bot computer (also known as a zombie computer) can be instructed by its commander to carry any number of nefarious activities – this is know as command-and-control (C&C). Multiple bots controlled by the same C&C are known as a botnet. These botnets are often used in distributed denial of service attacks (DDoS) which can wreak havoc on unsuspecting organisations' websites. Bots are often hybrids, starting out as trojans or worms during the initial infection but then carry out further attacks, as instructed by C&C

Malware attacks on Twitter have included drive-by-download links contained within tweets, cross-site scripting attacks [21], and Android malware that is controlled by tweets [78].

2.5 Sybil Attack

Named after Flora Rheta Schreiber's 1973 book *Sybil*, about the treatment of Sybil Dorsett (*a.k.a.* Shirley Ardell Mason) for dissociative identity disorder (also known as multiple personality disorder), the sybil attack involves a node in a peer-to-peer network operating under multiple identities at the same time. The attacker can subvert the reputation system of a network by creating a large number of false identities and leveraging them to gain a disproportionately large influence within said network. A network's vulnerability to a Sybil attacks depends on how "cheaply" identities can be created, how easily an entity can input to the network when said entity is lacking a chain of trust linking them to a trusted entity, and if the network treats all entities in the same way. Sybil attacks can occur on social networks when an attacker (or group of attackers) control large numbers of fake profiles in order to manipulate the platform [290].

2.6 Blacklists

A blacklist is defined as a set of elements to be blocked; an access control list. An example of a blacklist would be an email client that blocks known spam senders (e.g. spam@phishy.org). Any emails received from these senders would be marked as spam and possibly moved to an appropriate spam folder.

Phishing blacklists are a popular defence strategy that aim to protect people from phishing attacks. These blacklists typically contain known phishing URLs, providing an access control list which is used to prevent users from visiting these dangerous websites. For these phishing blacklists to be effective they need to be updated quickly and regularly to protect users from emerging phishing attacks. URLs should be removed from a blacklist when their website is no longer a threat – so as not to impact website visitors once it is safe – but also added again if that same website becomes a threat in the future. The number of URLs contained in a blacklist can contribute to its effectiveness; a small set of niche blacklisted phishing URLs will not provide a user with full protection compared to a large and comprehensive blacklist with a wide net. It is also important to understand the inner-workings of these blacklists as this can help determine how effective they will be at protecting people against phishing attacks.

Our study focuses on 3 popular phishing blacklists: Google Safe Browsing, PhishTank, and OpenPhish. These 3 blacklists are used by the web browsers Chrome, Safari, Firefox, Opera, email provider Yahoo! Mail, anti-virus providers McAfee, Kaspersky, Virus Total and Strong Arm, and online reputation and internet safety service web browser plugin Web Of Trust.

Other blacklist sources are available, however, this thesis focuses on these 3 due to their use in popular web browsers and services. Another popular blacklist, used by other research studies, is VirusTotal. Whilst we did consider leveraging the VirusTotal blacklist in our studies, one of its main limitations that made it infeasible for use in our studies is that it is rate limited to 4 requests per minutes. This means we would not be able to process the large volume of URLs that we can with the other 3 blacklists.

2.6.1 Google Safe Browsing (GSB)

Launched in 2007, Google Safe Browsing [102, 207] is a URL blacklist that contains both malicious and phishing [298] URLs and is used by the web browsers Google Chrome, Safari, Firefox, Opera, and Vivaldi to protect users from dangerous websites. We focus on GSB in our study because of its prominence in popular web browsers: already in 2012 GSB was protecting 600 million users from dangerous websites [294]. In 2015 GSB began using the term "Social Engineering" to categorise phishing websites which also encompass additional types of deceptive content. Google defines a social engineering web attack as occurring when either: *"the content pretends to act, or looks and feels, like a trusted entity – like a bank or government"* or *"the content tries to trick you into doing something you would only do for a trusted entity – like sharing a password or calling tech support"* [108]. During the week commencing 3rd September 2017 the total number of sites deemed dangerous by GSB was 573,433 phishing and 500,245 malicious. During that week GSB detected 24,756 new phishing sites and 6,312 new malware sites.

GSB defines malware websites in its blacklist as being either *compromised* or *attack*. A compromised website is a legitimate website that has been hijacked to either include, or direct users to, malicious content – such as drive-by-downloads [232]. An attack site is a website that has intentionally been set up to host and distribute malware [106]. During the week commencing 3rd September 2017, GSB identified 5,981 new compromised websites and 335 new attack websites.

GSB provides two APIs for accessing its blacklist: *Lookup* and *Update*. The *Lookup* API provides a remote service whereby URLs to be checked are sent to Google's servers and a response is returned for each URL stating if the URL is in the blacklist. The *Update* API provides the user with a local copy of the blacklist; this local copy is stored as a database of SHA-256 URL hash prefixes, the majority of the hash prefixes being 4 bytes. URLs are encrypted in GSB to ensure privacy [93]. To perform a URL blacklist lookup, the URL hash prefix is checked in the local database and, if there is a prefix match, then the full URL hash is retrieved from Google's servers to determine if there is a match on the full hash.

As we saw, in Section 1.9.2: *Phishing & Malware in Numbers*, Google Safe Browsing provides a Transparency Report [106] to illustrate the volume of malicious and phishing websites contained within its blacklist.

2.6.2 PhishTank (PT)

Launched in October 2006, PhishTank [226] provides a community based phishing website reporting and verification system. Users of the website can submit URLs of suspected phishing websites; the PhishTank community then vote as to whether these URLs are phishing or not. PhishTank is used by the web browser Opera, online reputation and internet safety service web browser plugin Web Of Trust, email provider Yahoo! Mail, and antivirus providers McAfee and Kaspersky[227]. The PhishTank blacklist of approved phishing URLs can be downloaded as a JSON file.

2.6.3 OpenPhish (OP)

Launched in 2014, OpenPhish [215] is the result of a 3 year research project on phishing detection that uses autonomous algorithms to detect zero day phishing websites. Our study has access to the academic feed. OpenPhish is used by the antivirus companies Virus Total and Strong Arm. The OpenPhish blacklist can be downloaded as a JSON file.

2.7 Ground Truth

Ground truth is a term often used in statistics and machine learning that means checking the results of a classifier for accuracy against real-world data. In this thesis our *ground truth* comprises our set of 3 blacklists (GSB, PT, and OP). That is, we only classify a URL as phishing or malware if said URL resides in at least 1 of these 3 blacklists. This means that if a genuine phishing or malware URL does not reside in 1 of the 3 blacklists (i.e. that URL is a false negative), we will not classify said URL as phishing or malware. In Section 9.11 we discuss the impact of ground truth on our studies.

2.8 Benchmarking

Benchmarking involves comparing the performance metrics of one system with another. In measurement studies, an existing study may provide a benchmark for future studies. This comparison can can help determine what has changed. For example: when evaluating the effectiveness of a given system, an existing study may conclude that the system is, say,

40% effective. A later study may determine that the same system is 60% effective. In this example, benchmarking the 2 studies may suggest that the system's effectiveness has increased, from 40% to 60%. However, when benchmarking, it is important to carefully examine the soundness, methodology, and contexts of both studies – and ensure the entire set-up is identical. Even small changes may cause unexpected changes to the measurements which could affect the benchmarking.

2.9 Snapshot

Measurement studies create a *snapshot* of the measurement environment at the time the study was conducted. Analogous to how a photograph creates a *snapshot* of a particular moment in time (including details such as people, buildings, etc), a measurement study's *snapshot* contains details of the environment; such as metrics, methodology, measurements, figures, etc. Measurement study *snapshots* contribute in many ways to various areas, such as future research, benchmarking, auditing, etc.

2.10 Soundness

Soundness is defined as:

> *"the quality of being based on valid reason or good judgement."*[1]

In deductive reasoning, an argument is *sound* if it is both valid and all of its premises are true. For example, the following syllogism argument is valid, but its premise is not true, therefore it is not not *sound*:

> All birds can fly.
> Penguins are birds.
> Therefore, penguins can fly.

Whereas the following argument is both valid and sound:

> No reptile has fur.
> All snakes are reptiles.
> Therefore, no snake has fur.

Paxson (2004) [222] defines soundness, in the context of internet measurement studies, as:

[1] https://en.oxforddictionaries.com/definition/soundness

> *"developing confidence that the results we derive from our measurements are indeed well-justified claims. By this we mean that we have a solid understanding of the strengths and limitations of the measurement process on which we base our results; and, likewise, a solid understanding of the quality of the chain of analysis supporting the results."*

2.11 Risk Appetite

ISO Guide 73:2009: Risk management – Vocabulary [133], defines risk appetite as:

> *"[The] amount and type of risk that an organisation is willing to pursue or retain."*

Risk appetite refers to how much risk an organisation is willing to accept in pursuit of its objectives, before action should be taken to reduce the risk. Different organisations will likely have different risk appetites, depending on the structure and type of the organisation.

2.12 URL Shorteners

As the name suggests, URL shortening services provide a way to reduce the length of URLs. This is often useful when character limits are enforced on platforms such as Twitter or when wanting to share a short and more memorable URL. For example, the long and unmemorable Pure Profile URL:

`https://pure.royalholloway.ac.uk/portal/en/persons/simon-bell(6786cfa5-1fc4-4744-8b2f-6a06dbd27ba8).html`

Can be shortened, using the URL shortener *Bitly*, to the shorter and more memorable URL:

`https://bit.ly/simon-bell-pure-profile`

Making the shortened URL ideal for sharing at conferences, networking events, or sharing on social networks, such as Twitter. This shortened URL simply forwards to the original URL. Some of the URL shortening services, such such as *Bitly*, also provide analytical data to track how many clicks the URL receives, demographics of visitors, etc. Some of the most popular and well-known URL shortening services include: *Bitly*, *TinyURL*, *Owly*, etc. Many organisations have their own URL shorteners such as Google's *Goo.gl*, Facebook's *fb.me*, YouTube's *youtu.be*, and Twitter's *t.co*. There are also URL shortening services that allow users to generate revenue from shortened URLs, such as *Adf.ly*, by displaying adverts to visitors of the shortened URLs.

2.13 Twitter

Twitter is a microblogging social network platform; allowing its users to broadcast – or *tweet* – messages to other users of the platform. Twitter users can *follow*, and be *followed* by, other Twitter users. By default, Twitter user accounts are set to public; anyone can view a user's tweets. Twitter users can also chose to set their profiles to *private*, whereby their tweets can only be seen by approved followers.

Since its creation in 2006, Twitter has gained over 974 million users with 330 million active users per month posting 500 million tweets per day [281, 22]. Among these Twitter users are many high profile celebrities, politicians, heads of state and societal influencers whom attract large numbers of followers [282]. Due to this large user base, Twitter makes an attractive target for malicious users aiming to carry out phishing and malware attacks to exploit people. One of the main ways these attacks are carried out is by leading victims to a malicious site, by including one or more URLs in a tweet, whereby the attack can occur.

Twitter has come under increasing pressure to protect its users against these attacks, such as, in 2010 when the company settled a case with the US Federal Trade Commission (FTC). Twitter agreed to "establish and maintain a comprehensive security information security program" throughout the platform, and to conduct an independently assessed bi-annual information security audit for 10 yers [86, 87, 88]. Twitter also agreed that, for the next 20 years, it must not mislead customers about "the extent to which it protects the security, privacy, and confidentiality of nonpublic consumer information, including the measures it takes to prevent unauthorized access to nonpublic information and honor the privacy choices made by consumers".

2.14 Phishing & Malware Defence

All of today's popular web browsers (see 1.9.3: *Web Browser Usage Statistics*) come with built-in phishing and malware defence systems to protect users against such attacks. Alongside blacklists to prevent users from visiting known attack websites, web browsers also feature heuristic technologies to detect dangerous websites – therefore potentially detecting un-blacklisted threats. The effectiveness of web browsers' ability to detect phishing websites is explored further in Chapter 7: *Web Browser Phishing Detection*.

There are also additional plug-ins which can be downloaded and installed on web browsers for additional protection. Many anti-virus software also includes anti-phishing technology which works independently of internet browsers (although some will install web browser plug-ins automatically). Finally, many popular email clients have built-in phishing

and malware detection, such as: Google's Gmail, Microsoft's Hotmail and Outlook, and Mozilla's Thunderbird.

2.15 APIs

Application program interfaces (APIs) essentially allow different systems to interact over a common interface. Therefore allowing two (or more) applications to talk to each other. APIs provide a set of routines, protocols, and tools for building software applications. This thesis uses APIs provided by Google, Twitter, and Bitly. By leveraging these APIs, our infrastructure – detailed in Chapter 4: *Design & Implementation* – is able to interact with these services to access their data and services.

APIs provide an effective way for researchers to access data provided by third party organisations. It is much easier to access data via an API due to the use of the APIs routines, protocols, functions, etc. These can directly interact with researchers' software and hardware in order to carry out accurate experiments.

2.16 Effectiveness

When investigating the effectiveness of Twitter's phishing and malware defence system, it is important to define what we mean by *effectiveness*. We define *effectiveness* as: *the degree to which Twitter successfully protects its users from phishing and malware attacks*. We measure effectiveness by *evaluating a predetermined set of metrics; assessed against a framework*. First, we shall explore existing literatures' definitions of *effectiveness* to decide which metrics constitute our effectiveness evaluation. We shall then expand on these existing definitions to define our own set of metrics, along with a framework for measuring effectiveness. We will use this definition and framework to assess the effectiveness of Twitter's phishing and malware defence system in our measurement studies.

2.16.1 Definition

Table 2.2 highlights key metrics that existing measurement studies use as effectiveness evaluators. We visit existing literature in more detail in Chapter 3: *Related Literature*. Key metrics in existing studies include, Grier *et al.* (2010) [110] use the effectiveness evaluation metrics: blacklist delays, number of blacklisted URLs posted to Twitter, and user clicks. Peng *et al.* (2019) [223] state that comprehensiveness is an important metric when assessing blacklists, stating that: *"even the best blacklists miss 30%"*. Guo

Study	Metric(s)	Scope of Study
Grier *et al.* (2010) [110]	Blacklist delays, number of blacklisted URLs posted to Twitter, and user clicks	Twitter spam measurement study
Oliver *et al.* (2014) [214]	Number of malicious tweets and clicks	Twitter malware measurement study
Ludl *et al.* (2007) [169]	Blacklist delay periods	Blacklist analysis study
Sheng *et al.* (2009) [251]	Blacklist delay periods	Blacklist analysis study
Zhang *et al.* (2013) [310]	Speciality and coverage	Blacklist analysis study
Metcalf *et al.* (2015) [183]	Speciality and coverage	Blacklist analysis study
Peng *et al.* (2019) [223]	Comprehensiveness	Blacklist analysis study
Guo *et al.* (2019) [162]	Volume, intersection, exclusive contribution, latency, coverage, and accuracy	Blacklist analysis study

Table 2.2: Key effectiveness evaluation metrics of existing measurement studies.

Metric	Description
Volume	Total number of indicators in a source
Intersection	Number of indicators in one source that are not in another
Exclusive contribution	Proportion of indicators in one feed but no others
Latency	When an indicator appears in multiple feeds: time taken to appear in additional feeds after appearing in the first
Coverage	Proportion of intended indicators in a source
Accuracy	Number of correct indicators in a source (i.e. precision)

Table 2.3: Guo *et al.* (2019) [162] effectiveness metrics.

et al. (2019) [162] define 6 metrics for assessing blacklist performance, shown in Table 2.3.

We will now define our set of effectiveness evaluation metrics. Our set consists of both existing (from previous studies) and new metrics. We

update many of the existing metrics to address soundness and limitations of existing work (which we explore in more details in Section 3.10: *Literature Review*). To the best of our knowledge, no existing studies have applied the same set of metrics to evaluate the effectiveness of Twitter's phishing and malware defence system. We believe this set of metrics, with the correct methodology, will improve the accuracy of the resulting evaluation.

To the best of our knowledge, no existing studies have examined the metrics "number of blacklisted phishing / malware URLs blocked by Twitter at time of click" or "number of known and unknown blacklisted phishing / malware URLs blocked by web browser". Therefore, we introduce these novel metrics to the set. In Section 3.10: *Literature Review* we highlight limitations of existing studies – including methodological limitations when measuring some of these metrics.

Our set of effectiveness evaluation metrics is:

- Number of blacklisted phishing / malware URLs posted to Twitter
- Number of blacklisted phishing / malware URLs blocked by Twitter at time of click
- Detection delay time (from time of tweet to blacklist membership)
- URL duration in blacklist
- User exposure to attacks in terms of:
  - Views of phishing/malware tweets
  - Clicks to phishing/malware URLs
- Web browser detection rate for:
  - Known (i.e. blacklisted) websites
  - Unknown (i.e. non-blacklisted) websites
- Threat intelligence in terms of:
  - Ground truth
  - Speciality in attack
  - Size of blacklist
  - Comprehensiveness of cover
  - Intersection of blacklists
  - Frequency of update
- Comparison to existing measurement studies

Specific contributing factors will determine *how* the aforementioned metrics influence the assessment of the effectiveness of Twitter's phishing

and malware defence system. For example, a high number of blacklisted phishing / malware URLs posted to Twitter shows that Twitter's phishing / malware defence system is not effective. We define these contributing factors in a framework.

2.16.2 Effectiveness Evaluation Framework

We define a framework for measuring the effectiveness of Twitter's phishing and malware defence system. We begin this subsection with a description of our framework, then present the framework itself towards the end of this subsection. Our framework provides context, contributing factors, and impact for the metrics we defined in the Section 2.16.1 (above). Our framework consists of:

Category (of metric)
Organised into:

- Tweet URL volume
- Time / delay
- User exposure
- Web browser
- Threat intelligence
- Benchmarking

Metric
The "item" to be measured. For example: the number of blacklisted phishing/malware URLs posted to Twitter, number of clicks, accuracy of ground truth, etc.

Effectiveness Contributor
What outcome of the **metric** (defined above) contributes to *effectiveness*. For example, consider the metric: "number of blacklisted phishing/malware URLs posted to Twitter". The lower this number, the more *effective* Twitter's phishing and malware defence system is. Therefore its *effectiveness contributor* is **lower**.

For the metric: "number of blacklisted phishing / malware URLs blocked by Twitter at time of click", a higher number is better (more effective). Therefore its *effectiveness contributor* is **higher**.

Effectiveness Impact
Some metrics will have a greater impact on the overall effectiveness. For example, the metric: "delay from time of tweet to blacklist membership", with effectiveness contributor: **lower**. The *effectiveness contributor* (*lower*) has a **very high** impact on the overall effectiveness – since it is directly related to protecting Twitter users from phishing and malware attacks.

In contract, consider the metric: "intersection of blacklists", with effectiveness contributor: **lower**. Whilst we saw that the intersection of blacklists *can* plan a role in blacklist effectiveness, its impact is generally considered *low*. This is because there are other factors that have a greater impact on effectiveness, such as the size of the blacklist, its accuracy, etc.

The effectiveness impact can be assigned one of the following values:

- VH: very high
- H: high
- M: medium
- L: low
- VL: very low

Table 2.4, on the following page, presents our framework for measuring the effectiveness of Twitter's phishing and malware defence system.

Category	Metric	Effectiveness contributor	Effectiveness impact
Tweet URL volume	Number of blacklisted phishing / malware URLs posted to Twitter	Lower	VH
	Number of blacklisted phishing / malware URLs blocked by Twitter at time of click	Higher	H
Temporal	Delay between tweet and blacklist membership	Lower	VH
	Duration in blacklist	Duration in blacklist $\geq$ duration of attack	VH
User exposure	Number of potential user views	Lower	H
	Number of clicks (measured via Bitly)	Lower	VH
Web browser	Number of known blacklisted phishing / malware URLs blocked by web browser	Higher (relates to blacklist update frequency)	L
	Number of unknown blacklisted phishing / malware URLs blocked by web browser	Higher	M
Threat intelligence	Accuracy of ground truth (e.g. num false positives)	High accuracy	VH
	Speciality in attack (e.g dedicated phishing blacklist)	High speciality	VH
	Size of blacklist (e.g. number of URLs in blacklist)	Higher	M
	Comprehensiveness of cover	Higher comprehensiveness	M
	Intersection of blacklists	Lower	L
	Frequency of blacklist updates	Higher	H
Benchmarking	Compare results to existing studies	Improvement on previous studies	VL

Table 2.4: Effectiveness evaluation framework to measure Twitter's phishing and malware defence system.

Web Browser Warning Effectiveness Framework

We also present a framework for measuring the effectiveness of web browser warnings, shown in Table 2.5. We use this framework in Section 7.4.7. The framework consists of 10 metrics which have been used in multiple, different, existing studies [300, 76, 6, 82]. The Egelman *et al.* (2008) [76] study features the concept of *passive* and *active* warnings. In their experiments, *passive* warnings are defined as warning that appear on the screen but do not obstruct the user's activity. An *active* warning, on the other hand, would interfere with the user's browsing experience and force the user to take note of the warning. The results from the 2008 study showed that 79% of people chose to heed active warnings and not visit phishing websites compared to just 13% of users that were presented with passive warnings.

Study	Metric
Wu *et al.* (2006) [300]	**Neutral information:** display information to user but no action taken
	System Decision: take action on behalf of user
Egelman *et al.* (2008) [76]	**Passive warnings:** display information but no action taken
	Active warnings: require user action to continue
Akhawe *et al.* (2013) [6]	**Click count:** to bypass warning
	Time spent on warning: before user aborts or continues to website
Felt *et al.* (2015)[82]	**Technical jargon:** use simple, non-technical language
	Reading level: should be 6th grade reading level – SMOG
	Brevity: be as brief as possible
	Specific risk description: describe the risks explicitly and unambiguously
	Illustration: images can reduce time to comprehend information and can make warnings more attention-grabbing and memorable
	Risk level: warnings must stand out from the surrounding environment; red warnings are more effective that black or green warnings

Table 2.5: Framework for measuring the effectiveness of web browser warnings.

CHAPTER SUMMARY

In this chapter we saw how measurement studies provide a crucial role in establishing facts in a specified environment. We explored soundness and how measurement study findings should be interpreted with context, understanding of background concepts, and existing literature. When correctly interpreted, measurement studies can contribute to researchers, policymakers, technology designers, etc.

We explored phishing's humble roots in the *warez* and hacking communities, and the first recorded mention of the term "phishing" in the AOHell hacking tool. We explored different types of phishing attacks; specifically website phishing – which this thesis focuses on.

We introduced the 3 blacklists featured in our research: Google Safe Browsing (GSB), PhishTank (PT), and OpenPhish (OP). We defined key concepts, such as URL shorteners, and explored defence against phishing and malware attacks.

Finally, we defined *effectiveness* as: *the degree to which Twitter successfully protects its users from phishing and malware attacks*. We measure effectiveness by *evaluating a predetermined set of metrics; assessed against a framework*. We concluded with our effectiveness evaluation framework.

3
Related Literature

OUTLINE
This chapter is split into 2 parts:

1. **Summarise existing literature (Sections 3.1 to 3.9):** to understand relevant background knowledge, explore various methodologies, justify our research methodology, and provide context to our research findings.
2. **Review existing studies (Section 3.10):** to identify and address methodological limitations. *We present this at the end of the chapter where we filter down to key related studies.*

Existing studies provide motivation, inspiration and justification for our own methodology and set-up. An important consideration when interpreting results from measurement studies involves understanding relevant context. Various aspects, such as phishers modus operandi, how malicious URLs propagate on a social network, and potential detection techniques *might* provide such context.

We begin this chapter by categorising related literature into topics to explore various methodologies and provide potential context to our research. Exploring existing literature also helps us to understand what the current state of phishing and malware attacks is, how such attacks work, and if people still fall victim to these attacks.

The following paragraphs provide an overview of the subsequent sections in this chapter:

Section 3.1: Phishing & Malware Attacks
We begin this chapter with an overview of phishing and malware attacks. We explore why the attacks work and how people fall victim to such attacks. This provides context to our results by understanding how people may reveal sensitive information as a result of such attacks.

Section 3.2: Data Feeds
We then look at the role of data feeds in research. We explore what data sources are available, the accuracy and reliability of data, measurement bias, and sampling techniques. These related studies help to justify the use of certain data sources in our measurement studies and understand potential issues around reliability and bias.

Section 3.3: Threat Intelligence & Blacklists
Building on from Section 3.2, we explore how threat intelligence and blacklists can provide a source of ground truth for cyber threats. But we must understand the limitations of these sources and how these limitations impact the accuracy of ground truth. This selection of studies is particularly relevant to the ground truth of our measurement studies and justification of blacklists we use.

Section 3.4: Phishing & Malware Defence
Although outside the scope of our research, the studies in this section provide important context to our research. We explore research into phishing and malware defence to help understand the significance of our measurement study results. For example: if we determine that twitter does not defend its users from attacks, are other technologies available that *can* protect Twitter's users? These studies are important because their "contents" may be outside the metrics of our measurement – therefore potentially impacting our results.

Section 3.5: Web Browser Phishing Detection
Studies in this section relate to our web browser phishing detection study in Chapter 7. We focus on 2 key studies [311, 23] that examine the effectiveness of web browsers' phishing defence. We highlight specific limitations of these 2 studies in our Literature Review in Section 3.10.

Section 3.6: Heuristic Phishing Detection
Studies in this section also relate to our web browser phishing detection study in Chapter 7. We look at studies that have explored how web browsers might detect phishing websites without requiring blacklists.

Section 3.7: Warning Effectiveness

Studies in this section also relate to our web browser phishing detection study in Chapter 7. We look at studies that have explored the effectiveness of web browser warnings. These studies, and their findings, contribute towards our web browser warning effectiveness framework (Section 2.16.2) that we design and then use to evaluate web browser warnings we observe in our study.

Section 3.8: Measurement Studies
We now begin our exploration of existing measurement studies. This section is split into subsections of related studies, categorised by topic. Each topic relates to our thesis by providing motivation and inspiration for our methodology and also context to our results:

- **Section 3.8.1: URL Shorteners**
- **Section 3.8.2: Information Credibility**
- **Section 3.8.3: Networks**

Section 3.8.4: Phishing & Malware
In this part of the measurement studies section we focus on existing phishing and malware measurement studies. We split these studies into 2 categories: studies that measure phishing and malware on *the web in general* and those that specifically measure *on Twitter*. This categorisation helps to differentiate between methodologies for general internet measurements, and methodologies specifically applied to Twitter. We begin the section with phishing and malware measurement studies on the web in general, before focusing on Twitter.

Section 3.9: Ethics
The subject of measuring cybercrime on a public social network can be a controversial one. Therefore, building on from our introduction to ethics in Section 1.7, we now explore how existing studies approach ethics and we review their discussions.

Section 3.10: Literature Review
Our final section in this chapter examines key existing studies; presenting our literature review. This is where we highlight specific limitations of existing studies' methodologies and begin to address how such limitations can be addressed.

3.1 Phishing & Malware Attacks

This section explores existing research into phishing and malware attacks at a high level. Exploring the human aspects and psychology behind why phishing attacks work along with usability studies that investigate how people interact with phishing attacks and also mitigation techniques.

Dhamija *et al.* (2006)

Dhamija *et al.* (2006) [72] investigated the human aspects of why phishing attacks work; exploring which malicious strategies are successful at deceiving users. By exploring 200 phishing attacks in the APWG [17] archive (collected in 2005), the authors developed a set of hypotheses – organised along 3 dimensions: lack of knowledge, visual deception, and lack of attention – about why these attacks worked. The authors then carried out a usability study in which 22 participants were shown 20 web sites (7 legitimate; 9 representative phishing websites; 3 phishing websites constructed using additional phishing techniques; and 1 website requiring users to accept a self-signed SSL certificate) and asked to determine which ones were fraudulent. The study observed that 23% of participants did not look at browser-based cues such as the address bar, status bar, and general security indicators, which lead to incorrect choices 40% of the time. In their study, the most convincing phishing website was able to fool more than 90% of participants.

Jagatic *et al.* (2007)

Jagatic *et al.* (2007) [136] examined the baseline success rate for phishing attacks by carrying out a usability study on 1,731 participants. The authors of the study "acquired" these participants by crawling online public profiles, of students studying at the same university, in an attempt to quantify how reliable social content would increase the success rate of a phishing attack. It is important to note that the participants did not know they were part of a research study.

Participants were split into 2 groups: *social* and *control*. Those in the *social* group received a spoofed email that appeared to come from a known friend, containing a link a phishing website. Those in the *control* group received the same email but from an unknown sender. 921 of these participants received phishing attacks and 810 had their email address spoofed.

Of the 921 participants that received phishing attacks: 487 were in the *social* group and 94 in the *control* group. 349 (72%) of the *social* group and 15 (16%) of the *control* group fell for the attacks. The authors define "falling for the attack" as: the recipient clicked the link in the phishing

email *and* authenticated with their valid university credentials. The study also explored the psychological emotions experienced by victims in their usability study of phishing attacks.

Alsharnouby *et al.* (2015)

Alsharnouby *et al.* (2015) [13] explored "why phishing still works" and defined a set of user strategies to reduce phishing attacks. Their study determined that only 53% of users were able to successfully detect phishing websites – despite participants being primed to detect phishing websites. The study used eye tracking technology to measure where users focus their attention and to determine how effective web browser security indicators are. The study also concludes that a user's technical proficiency does not correlate with improved detection scores.

Stajano *et al.* (2009) & Ferreira *et al.* (2015)

Other existing studies have also explored the human aspects of phishing attacks, including: Stajano *et al.* (2009) [256] investigated scam victims to define 7 principles for systems security: distraction, social compliance, herd, dishonesty, kindness, need and greed, and time. Ferreira *et al.* (2015) explored the psychology behind why people fall for phishing attacks by analysing the persuasion techniques used in social engineering and their use in phishing

Current State of Malware Studies

Existing research, by Provos *et al.* 2007 [233], has also provided a contemporary state of malware on the web, along with its evolution – over a 12 month period, starting in March 2006. Their study analysed web-based malware; characterising threats into 4 main categories: web server security, user contributed content, advertising, and third-party widgets. Existing studies have also surveyed the current state of malware, malware detection systems, and malware evasion techniques [152, 89, 175, 243, 125]. Research has also explored the impact of phishing attacks on healthcare organisation system along with mitigation approaches [230].

SECTION SUMMARY

The studies in this section have demonstrated how easily people can be fooled by phishing attacks. Key aspects such as lack of knowledge and attention can lead victims into being deceived by such attacks. We also see somewhat questionable ethics in the Jagatic *et al.* study whereby participants were not initially aware that they were involved in a research

study. One could argue that the study would only work if participants did not know they were involved. Although, as seen in the Alsharnouby *et al.* study, participants can still be fooled by phishing attacks – even when they know they're involved in a research study.

3.2 Data Feeds

This section explores the role of data feeds in research; exploring themes such as data sources, the accuracy and reliability of data, measurement bias, sampling techniques, etc. It is important to understand these themes before planning and carrying out a measurement research study in order to produce the most accurate and reliable results.

Tall *et al.* (2009)

A specific example of a data source of phishing emails includes the Phisherman project: Tall *et al.* (2009) [265] – a data repository collected by numerous different organisations. The project aims to *"create a single information resource available that provides ready access to ongoing and historical phishing attacks for first responders, brand owners, researchers, and law enforcement"*. Stating that challenges include *"rapid automated validation to cope with the increasing scale of the attacks, and privacy protection, particularly when attacks are targeted at specific individuals"*.

Pitsillidis *et al.* (2012)

Pitsillidis *et al.* (2012) [228] present a comparative analysis of spam email feeds; providing examples of data feeds that are available to researchers. Their study characterises 10 distinct spam feeds by comparing the contents of these data sources – including assessing false positives in the dataset, timing of emails collected and temporal uncertainty, and scope.

The study's main findings suggest significant variations in the spam feed contents based on how the data is collected – e.g. spam collected via botnet, MX honeypot, seeded honey accounts, human identification, and domain blacklists. The study also shows how research conducted using these data sources can produce differences in findings. The study suggests that *"high-quality blacklist feeds offer very good coverage and first appearance information. They also offer the best purity since they are usually commercially maintained, and have low false positives as their primary goal"*. However, the study does note that *"[blacklists] are less useful for studies that rely on last appearance or duration information"*.

Vis *et al.* (2013) & Lomborg *et al.* (2014)

Vis *et al.* (2013) [286] discussed how data is "made", by whom and how; focusing on the use of APIs – specifically Twitter's – for collecting data in research studies and the tools used to process that data. The study explores aspects such as how most academics use Twitter's free API – a random sample of all global tweets – and do not have access to the firehose dataset – which includes 100% of all public tweets – "due to the cost involved in purchasing and processing firehose data"; how Twitter samples its firehose dataset; how collected data should be inspected for duplicates, spam, etc; and how features change on Twitter (i.e. data collected in 2009 will be different in 2013 due to features specific to Twitter's platform being added, updated, retired). The study also notes how many researchers do not include enough details about how they collect data, such as: search and filter settings, timeframe and dates when data was collected, which APIs or sources were used, etc.

Similarly, Lomborg *et al.* (2014) [167] discussed how social media companies that share their data via APIs can help the research community carry out empirical analysis – as opposed to researchers deploying web crawling to collect data. The study focuses on Facebook and Twitter as social media companies that make their data available for researchers via APIs. The study highlights methodological considerations (such as sampling and constraints on generalisation; validity and data quality – observing "what" social media users are doing does not necessarily explain"why" they are doing it; reliability, in terms of software and API design architecture), along with legal and ethical implications (such as informed consent and anonymisation) of empirical research that makes use of APIs for data collection.

Gonzalez *et al.* (2014)

Gonzalez *et al.* (2014) [99] assess the bias in samples of large online networks by comparing 3 samples of Twitter data collected through the search and streaming APIs. The study analyses communications between Twitter users relating to political protests taking place during a 1 month period in May 2012; reconstructing the network of mentions and re-tweets according to the search and streaming APIs. Key findings show that smaller samples do not present an accurate picture of peripheral activity and that bias is greater for the network of mentions.

Whilst the Gonzalez *et al.* (2014) study highlights important considerations for sampling Twitter data, the specific use case of the study's data collection involves analysing crowds and large groups of communities with the aim to monitor and interpret communications *in context*. Whereas an analysis of non-interconnected activities on the social network – such as how many users mention a certain hashtag, or tweet a specific URL –

would not fall prey to the same bias, since the interactions are not so tightly related. For example: a random sample of 1% of Twitter users that mention a specific hashtag, compared to a 10% or 100% random sample, should produce similar results – since the only relation is the shared mention of a hashtag.

Morstatter *et al.*

In a similar study, Morstatter *et al.* (2013) [196] compare data from Twitter's streaming API service with data collected via Twitter's "full, albeit costly," Firehose stream service. The study addresses researchers' concerns around Twitter's lack of documentation regarding what and how much data is shared through their streaming API, and whether sampled data is a valid representation of the overall activity on Twitter. Using both statistical metrics and metrics that facilitate the comparison of topics, networks, and location of tweets, the study concludes that the "results of using the Streaming API depend strongly on the coverage and the type of analysis that the researcher wishes to perform".

Similarly to the previous study, some of the strategies used by Morstatter *et al.* to determine bias involve assessing strongly interconnected aspects, such as most frequent hashtags and topic analysis. Additionally, certain methodologies and techniques can be deployed to compensate for these biases. For example: Twitter provides a "trending hashtags" API which allows researchers to quickly determine current trending (most frequently included in tweets) hashtags in specific geographical regions – which would alleviate this specific bias raised by Morstatter *et al.*

Morstatter *et al.* produced a follow-up study in 2014 [195] which explained how Twitter's Sample API can be used to detect bias in the Streaming API – without having to use Twitter's Firehose data steam. To detect bias in the trend of a hashtag, the study demonstrates bootstrapping[74] the Sample API to obtain a confidence interval for the relative activity for the hashtag.

Ghosh *et al.* (2013)

Ghosh *et al.* (2013) [96] (a study that we discuss in more detail in Section 3.8.2: *Information Credibility*), investigated the most effective way to sample data generated by users in social networks. They compared data produced by "random" versus "expert" (i.e. knowledgeable) user sampling. The study opens its problem statement with: *"though some research studies [68, 164] have used the firehose, very few organizations in the world have access to the firehose. So most analytics companies and researchers rely on sub-sampled data rather than the entire dataset. Against this background, this paper investigates the following key question: What is the most effective*

way to sample the data generated by the users in social networks?". Part of the study's conclusion states that *"...random (1%) sampling preserves certain important statistical properties of the entire data set, which expert sampling does not. For example, expert sampling does not capture conversational tweets that might be deemed as less important by experts."*. Although this paper is making a comparison between the 2 sampling methods – random versus expert users – their research demonstrates that Twitter's sub-sampled API is still a reliable source of data.

Clayton *et al.* (2015)

In a 2015 study entitled *"Concentrating Correctly on Cybercrime Concentration"* [60], Clayton *et al.* discussed a specific "concentration bias" that may arise as a result of hidden characteristics of how data is collected. For example: PhishTank might report 40% of phishing URLs, but 100% of PayPal phish – so it appears that criminals overwhelmingly target PayPal. Concentration bias may also arise when only a handful of criminals operate a certain scam. For example: In 2007, the RockPhish gang accounted for 2/3 of phishing spam – so their techniques suggested concentrations for most phishing criminals.

Social Media Data Feeds

With regard to leveraging social media data feeds: Osborne *et al.* (2014) [217] investigated Facebook, Twitter, and Google plus to determine which is best for breaking news. Key results from the study show that all 3 sources carried the the same major events, but that Twitter consistently leads Facebook and Google Plus – with Facebook and Google Plus mostly reposting newswire stories from multiple sources. This study shows that Twitter is a useful source for researchers that want to tap into a data source that is the most popular platform for users.

Burnap *et al* (2015). [50] produced COSMOS (the collaborative online social media observatory): a system to analyse large social media datasets and manage workflows between tools. Longley *et al.* (2015) [168] explored the geotemporal demographics of Twitter usage.

Martin *et al.* (2015) [176] investigated potential sampling bias that can arise when using subsets of data – focusing on App Store Mining and the App Sampling Problem of user reviews. The study notes that specific metrics (e.g. price, rating, download rank) can vary significantly between subsets – and suggest the use of correlation analysis to find trends across partitions to mitigate sampling bias.

Finally, Fungherr *et al.* (2016) [141] carried out a systematic literature review of Twitter use in election campaigns, analysing 127 papers. The study notes that Twitter's Streaming API is a popular method used by

researchers for collecting data from the social media platform, with popular selection criteria being: users, hashtags, and keywords. However, the study also notes that a large number of papers in the literature review do not specify their mode of data collection. This makes it difficult to verify research findings, replicate studies, and compare data.

Later on in this thesis, in Section 3.9: *Ethics*, we will review existing studies that have explored ethical issues surrounding the use of data feeds in cyber security research [266, 10] and concerns about anonymising data [5].

SECTION SUMMARY

As we have seen in this section, there are numerous data feeds available to researchers. There are also numerous ways to leverage those data feeds. As noted by Pitsillidis *et al.*, the way in which data providers collect their data can have a significant impact on the quality and usefulness of that data. Multiple data sources may produce different results depending on the suitability of each data source for a particular study.

We also see from these existing studies that Twitter's free Stream API is a popular data source amongst researchers. The cost of Twitter's Decahose and Firehose data streams simply make them inaccessible for the majority of academics. However, leveraging Twitter's free Stream API as a data source may introduce bias – depending on how the data is used and the research questions posed. Research that relies on interconnected tweets (i.e. analysing crowds in context) may incur bias, but techniques are available to alleviate bias, such as bootstrapping and Twitter's Trending Hashtags API.

3.3 Threat Intelligence & Blacklists

Threat intelligence, or cyber threat intelligence, provides information about current, known active threats. For organisations wanting to protect themselves from cyber attacks, threat intelligence plays a key role in their defence mechenisms. For researchers wanting to explore current cyber attacks, threat intelligence can provide a ground truth to establish what is malicious and what is benign in the online world. However, is is important to understand the accuracy and reliability of threat intelligence – which includes blacklists – to ensure this ground truth is correct.

Jung *et al.* (2004)

Jung *et al.* (2004) [139] examined spam emails and the use of DNS blacklists as a filtering technique. Measured between December 2000 and February 2004, the study observed that the volume of email increased by 866%

between 200 and 2004; blacklist lookups accounted for 14% of all DNS requests they observed in 2004 – compared to just 0.4% in 2000. The study also analysed 7 popular blacklists (abuseat, dsbl, opm, rfc-ignorant, sorbs, spamcop, spamhaus) and observed that 80% of spam sources are identified in some DNS blacklists; some DNS blacklists appear to be well-correlated with others.

The measurement study was carried out on the authors' university mail servers (at MIT's Computer Science and Artificial Intelligence Laboratory – CSAIL); collecting bidirectional TCP SYN/FIN/RST packet traces [140]. Although the Jung *et al.* study focused on spam emails, their results show that domain blacklisting can be an effective means of detecting spam emails – and therefore a potentially reliable ground truth.

Ludl *et al.* & (2007) & Sheng *et al.* (2009)

Two key studies: Ludl *et al.* (2007) [169] and Sheng *et al.* (2009) [251], focused on phishing blacklists and how effective they are at protecting users from phishing email attacks, paying particular attention to the delay from an email containing a phishing URL being received to that URL appearing in a blacklist.

Ludl *et al.* (2007) automatically tested the effectiveness of the blacklists maintained by Google and Microsoft with 10,000 phishing URLs. Their findings showed that the Google phishing blacklist initially detected 64.14% of the 10,000 phishing URLs and that Microsoft's blacklist detected 50.48%. These figures rose to 65.22% and 55.88%, respectively, by the end of the experiment due to delayed additions to the blacklist. After removing phishing websites that were offline, and therefore not harmful to users, Google's blacklist initially detected 87.89% of the now 3,592 remaining URLs and Microsoft's detected 59.55%. These figures rose to 90.23% and 67.18% respectively, after delayed blacklist additions. Ludl *et al.* conclude that phishing blacklists are "quite successful in protecting users" against phishing attacks, especially considering that Google's blacklist detected over 90% of the phishing attempts.

Sheng *et al.* (2009) also examined the effectiveness of phishing blacklists. Their study attempted to show the limitations of the Ludl *et al.* study. The Sheng *et al.* study's main argument for the limitations of the Ludl *et al.* study was that the phishing websites used were not fresh enough. Sheng *et al.* showed, in their study, that blacklists take time to update their databases and therefore fresh phishing websites take time to be detected. Their main conclusion was that phishing blacklists are ineffective on fresh phishing websites – defined as being less than a few hours old. The blacklists tested by Sheng *et al.* were: Microsoft Internet Explorer version 7, version 8, Firefox 2, Mozilla Firefox 3, Google Chrome (0.2.149.30), Netcraft toolbar

(1.8.0), McAfee Site-advisor (2.8.255 free version), and Symantec Norton 360 (13.3.5). Two main tests were carried out in October and November of 2008. Results show that the majority of phishing blacklists do not reach their full effectiveness until at least 24 to 48 hours have passed since the URL was discovered. This means that, when using just a phishing blacklist as a means of protection, users are vulnerable to fresh phishing websites that have not been detected yet. Users are particularly at risk at time zero of phishing websites being published.

These 2 studies, by Ludl *et al.* and Sheng *et al.*, illustrate that blacklists can be an effective ground truth – but that the temporal aspects of blacklists should be considered.

Zhang *Et al.* (2013) & Metcalf *et al.* (2015)

Zhang *Et al.* (2013) [310] analysed 9 blacklists from 3 different categories: spam (CBL, BRBL, SpamCop, WPBL, and UCEPROTECT), phishing/malware (SURBL, PhishTank, hpHosts), and active attack/probing behaviour (Dshield) over a 7-day measurement period. Key results show a significant overlap within the same category of blacklists: BRBL and CBL (the 2 largest spam blacklists) cover about 90% of other spam-related lists; hpHosts, PhishTank, and SURBL (phishing/malware blacklists) also significantly overlap. Whereas overlaps between categories are trivial.

Similarly, Metcalf *et al.* (2015) [183] compared the contents of 86 internet blacklists to analyse the ecosystem of blocking network touch points and blacklists. The study selected blacklists to cover a variety of target behaviours and geographic areas, such as botnet command and control, spam email senders, phishing senders, identifiers within email message bodies, scanning, and malicious download locations. Key results show that domain-name-based indicators are unique to one list 86% to 97% of the time and that IP-address-based indicators are unique to one list 82% to 95% of the time; *i.e.* there is little overlap between the blacklists they analysed.

The study concludes that each blacklist contains a distinct set of malicious activity; combining all of these blacklists will not produce a global ground truth. The study also notes that "academics comparing their results to one or a few blacklists to test accuracy are advised to reconsider this validation technique". Experiments carried out for our thesis include 3 popular phishing blacklists (PT, OP, GSB). Whilst every effort was made to include a "complete coverage" of phishing blacklists, a number of considerations, such as financial, contractual, etc, impacted our ability to include such a complete set of phishing blacklists. We believe that the core blacklists we have used in our experiments are significant in their coverage and speciality, therefore should provide accurate results.

Kuhrer *et al.*

Kuhrer *et al.* (2012) [148] presented the design and evaluation of a blacklist analysis system which the authors used to analyse 49 distinct blacklists. They collected more than 2.2 million unique blacklist entries and more than 410,000 distinct URLs during an 80-day measurement period. In 2014 [149], Kuhrer *et al.* leveraged their 2012 framework to analyse 15 public malware blacklists and 4 blacklists operated by antivirus vendors. Key results show that less than 20% of malicious domains are detected between all 15 analysed blacklists, and that most antivirus blacklists failed to protect against malware that utilise domain generation algorithms.

Peng *et al.* (2019)

Peng *et al.* (2019) [223] investigated VirusTotal and its 68 third-party vendors to examine their labelling process on phishing URLs. The study's core methodology involved creating their own phishing URLs (mimicking PayPal and IRS) then submitting said URLs for scanning by VirusTotal – allowing them to analyse incoming network traffic and dynamic label changes at VirusTotal.

Key results show that even the "best" vendors missed 30% of their phishing URLs and that scanning results are not immediately updated to VirusTotal after scanning; along with inconsistencies between VirusTotal scans and some vendors' scans. Although the VirusTotal dataset is not used in this thesis, results from Peng *et al.* illustrate how blacklist coverage is not perfect.

Guo *et al.* (2019)

Guo *et al.* (2019) [162] measured the characterisation of threat intelligence to understand how effective these methods are as defence mechanisms – between January 1 2016 and 31 August 2016 and again between 1 December 2017 and 20 July 2018. The study defined a set of 6 threat intelligence metrics:

1. Volume: total number of indicators in a source
2. Intersection: number of indicators in one source that are not in another
3. Exclusive contribution: proportion of indicators in one feed but no others
4. Latency: when an indicator appears in multiple feeds: time taken to appear in additional feeds after appearing in the first
5. Coverage: proportion of intended indicators in a source

6. Accuracy: number of correct indicators in a source (*i.e.* precision)

The study then analysed 47 distinct IP address threat intelligence sources covering 6 categories of threats (scan, brute-force, malware, exploit, botnet, spam) and 8 distinct malware file hash threat intelligence sources. Key threat intelligence sources analysed in the study include: Facebook ThreadExchange [79], Paid Feed Aggregator, Paid IP Reputation Service, Public Blacklists and Reputation Feeds (AlientVault, Badips, Abuse.ch, Packetmail, SpamhausSBL, CBL, SORBS).

Key results show that threat intelligence sources vary widely based on their collection methods and specialist areas; few sources explain their data collection methodology; labels can be ambiguous; larger feeds do not necessarily contain "better" data; data collection does not necessarily imply the feeds' attributes (*i.e.* crowd-sourced feeds are not always slower than self-collecting feeds); most threat intelligence sources are singletons (low intersection) with higher-correlating data sources often seeing intersections of about 10%.

The study notes that the relevance of its 6 defined threat intelligence metrics (volume, intersection, unique contribution, latency, coverage, and accuracy) will vary depending on a number of factors, such as the use case for a specific threat intelligence source, cost of false positives or false negatives, cost of latency, budget, etc. The study acknowledged that a "complete" coverage of threat intelligence sources is difficult to achieve – especially for academic researchers – for a number of reasons, such as "prohibitively expensive or publication-restricted data sources". The study also acknowledged limitations in ground truth, stating that "It is simply very difficult to obtain the full picture of a certain category of threat, making it very challenging to precisely determine accuracy and coverage of feeds".

Ramachandran *et al.* (2007)

Ramachandran *et al.* (2007) [237] observed that 35% of spam emails were missed by the Spamhaus and SpamCop blacklists because the spam sender's IP addresses did not reside in the blacklists. In addition to this, 20% of these IP addresses did not appear in these blacklists after 1 month – most IP addresses evaded the blacklists for about 2 weeks before they were added; some evaded the blacklists for almost 2 months. To address this issue, the author's created a spam filtering system, called *SpamTracker*, that analyses the patterns of spam senders in order to detect spam – early tests showed that their proposed *behavioural blacklisting* technique can "complement existing blacklists" – detecting many spammers before they appear in blacklists.

Sinha *et al.* (2008)

Sinha *et al.* (2008) [254] investigated reputation-based blacklists used to block spam to determine how effective these techniques are at detecting spam. The study's ground truth consists of 4 blacklists (NJABL, SORBS, SpamHaus, and SpamCop) which were used to analyse more than a million spam emails received by a university of 7,000 hosts during a measurement period of 10 days in June 2008.

Key results show that the NJABL blacklist contained 90% false negatives, SORBS contained 65%, SpamCop contained 35%, and SpamHaus contained roughly 36% false negatives. False positive rates were generally low across all blacklists, except SORBS which saw an overall false positive rate of 10%. The study also observed that 90% of undetected spam sources sent very little spam and appeared on their network for just 1 second. These results, similarly to Ramachandran *et al.* show that popular blacklists can have very high false positives – therefore allowing large volumes of spam to go undetected and reach users' inboxes.

Thomas *et al.* (2016)

Thomas *et al.* (2016) [267] investigated threat intelligence "exchanges"; exploring the potential advantages and disadvantages of global reputation tracking. Their data includes 45 million IP addresses which abused (spam, bulk account creation, fake engagement, malware distribution) 6 Google services (Gmail, YouTube, ReCaptcha, comment spam, bulk account creation, and Safe browsing) during a 14-day period between 7 and 21 April 2015.

Key results showed that 66% of abusive IP addresses remained active for only 1 day – after which dynamic IP address reallocation caused the host to change. Additionally, NATs polluted IP reputation – due to the large user populations they represent – therefore only 38% of abusive IP addresses exclusively relayed harmful traffic; the rest shared some overlap with benign devices. This lack of intersection and correlation among the different sources makes sharing threat intelligence data challenging – although these studies [237, 254, 267] focused on blacklists that contain IP addresses as threat sources.

Moore *et al.* (2008)

Moore *et al.* (2008) [191] explored the effectiveness of the crowd-reporting service PhishTank; examining reports from 176,366 phishing URLs submitted between February and September 2007. The study observed that PhishTank is dominated by its most active users and that participation follows a power-law [208] distribution – which *could* allow the service

to be manipulated. The study analysed 3,798 PhishTank users, casting a total of 881,511 votes – of which, the top two submitters (anti-phishing organisations themselves), contributed 93,588 and 31,910 phishing records respectively. Additionally, the top verifiers of PhishTank voted over 100,000 times – whereas most users vote only a few times.

The study also noted that the PhishTank voting community is made up of 25 moderators; collectively casting 652,625 votes (75% of all votes) – showing that moderators contribute towards the majority of votes – but that a significant contribution is still made by regular users. The study compared PhishTank's blacklist to the curated phishing blacklist of an unnamed "company" – finding PhishTank to be slightly less complete and significantly slower to confirm phishing URLs than the blacklist of "company". The study also explores cases where incorrect information has propagated into PhishTank.

Overall, the study observed that PhishTank's phishing URLs were mostly accurate; identifying only a few incorrect URLs – all of which were later corrected. The study concludes that they "do not advocate against leveraging user participation in the design of all security mechanisms".

Chen *et al.* (2015)

As we have seen in this section, existing research studies typically rely on one or more data feeds or blacklists as their source of ground truth. Despite its promising title, *"6 million spam tweets: A large ground truth for timely Twitter spam detection"*, the 2015 study by Chen *et al.* levered access to Trend Micro's Web Reputation Technology as its ground truth. Nevertheless, the study states that: *"currently researchers are using two ways to generate ground truth, manual inspection and blacklists filtering. While manual inspection can label a small amount of training data, it is very time–and resource-consuming. A large group of people is needed during the process. Although HIT (human intelligence task) websites can help to label the tweets, it is also costly, and sometimes the results are doubtful* [53]. *Others apply existing blacklisting service, such as Google SafeBrowsing, to label spam tweets. Nevertheless, these services' API limits make it impossible to label a large amount of tweets. We used Trend Micro's Web Reputation Service* [214] *to identify which URLs were deemed malicious tweets... We define the tweets which contain malicious URLs as Twitter spam. In our dataset of 600 million tweets, we identified 6.5 million malicious tweets, which accounted for approximately 1 % of all tweets"*. The study does not provide further information about the accuracy of Trend Micro's Web Reputation Technology – their ground truth. However, the study does provide a potentially useful benchmark for the percentage of spam on Twitter.

SECTION SUMMARY

In this section we have explored existing studies that investigate threat intelligence and blacklists. We see that blacklists can provide a reliable defence from, and source of ground truth for, cyber threats. However, the timings of blacklists needs to be considered since there may be delays from when a threat becomes active online to when a threat appears in a blacklist. We also see that blacklists of the same category can contain a significant overlap which helps to achieve a wide coverage. However, compiling a complete and comprehensive ground truth is a challenging and difficult task. Therefore most organisations will make trade-offs to achieve a good enough ground truth that provides effective protection for their needs.

In this thesis our ground truth is provided by 3 popular phishing and malware blacklists (GSB, OP, and PT). These blacklists specialise in phishing attacks – which is a key threat that we are researching – and provide effective coverage for our requirements. We see from existing studies that IP blacklisting can be ineffective because IP addresses are often reused and change frequently – therefore making it difficult to pinpoint threats. However, URL blacklisting is also not perfect, since it can have high false positive rates – *i*.e. they do not detect all threats. Finally, we also see that crowd-reporting, as provided by the PhishTank blacklist, is a reliable threat intelligence source. Despite some potential flaws in a crowd-reporting system, as raised by Moore *e*t al., the technique remains reliable and the PhishTank blacklist was deemed accurate.

3.4 Phishing & Malware Defence

Although somewhat outside of the scope of this thesis, it is sill within context and therefore important to understand what defence techniques are available to protect users against phishing and malware attacks. Some of the research carried out for this thesis gives examples of where Twitter might not protect its users against phishing and malware attacks. However, any number of these users may still have been protected from such attacks by other defence mechanisms, such those listed in this section.

A high level overview, provided by Gupta (2017) *et al.* [114] explored the state of the art and future challenges in the fight against phishing attacks. Along with a literature review of contemporary phishing detection and defence solutions, the study discusses the history of phishing attacks, motivation for attackers, along with a taxonomy of various types of phishing attacks. Similarly, Purkait *et al.* (2012) [234] reviewed available phishing literature and phishing countermeasures to determine trends in the literat-

ure itself along with said attacks. Almomani *et al.* (2013) [12] present a comparative study and evaluation of phishing email filtering techniques to provide an understanding of the problem, its current solution space, and potential future research directions.

Two core principles for protecting users again phishing and malware attacks include prevention and detection. In terms of prevention, various techniques have been proposed, including Dynamic Security Skins [71] and Trusted Devices [220] along with educational aspects of phishing training including PhishGuru [150] and the game Anti-Phishing Phil [250]; the effectiveness of these two educational approaches were analysed [151]. The effectiveness of toolbars in protecting users has been investigated [300, 311], along with the effectiveness of web browser warnings [76, 206, 6]. Work has explored demographic analysis of phishing susceptibility and effectiveness of interventions [249]. Research has also explored if smaller screens on mobile devices can contribute towards users being more susceptible to phishing attacks [213], along with using using eye movement tracking to access how "alert" users are during phishing attacks [204], and explored mouse behaviour as an index of phishing awareness [308].

In terms of detection, there are a number of key focus areas here: phishing websites themselves, URLs, and for the social network: Twitter specific features. We see that machine learning plays an important role here due to its ability to process and detect patterns in large amounts of data – although it's important to understand the drawbacks of any machine learning approach and the importance of training data [9]. A number of literature reviews exist that explore contemporary state of the art in machine learning techniques for phishing detection [2, 145, 7].

An existing study proposed, and examined the effectiveness of, proactively blacklisting malicious and phishing domains [81, 229]. A number of techniques to detect phishing websites have bene proposed [1, 25, 44, 83, 91, 142, 143, 163, 172], such as: CANTINA [312] (and CANTINA+ [301]) to examine website content – along with examining domain top-page similarity features [244], and large-scale automatic classification [298]. Basnet *et al.* have explored a number of automated detection techniques [27, 30, 31], including mining the web to detect phishing URLs [29] and the use of confidence-weighted linear classifiers to detect phishing emails [28]. Anomaly based phishing website detection has been explored [219], along with exploitable redirects on the web [253], and a hybrid phish detection approach by identity discovery and keywords retrieval [302].

In their 2011 paper entitled *"Evaluating a semisupervised approach to phishing url identification in a realistic scenario"*, Gyawali *et al* [115] discuss the problem of working with an inbalanced data set (phishing and spam URLs with 1:654 ratio) and training on data where only 10% is labelled. In their proposed solution, *"labelling only 10% of the URLs manually and*

using a semisupervised learning algorithm... Evaluation results show that our proposal is competitive if it is applied in combination with appropriate feature selection and undersampling techniques". Additionally, a number of existing studies have developed techniques to detect phishing and malware attacks using URL-based heuristics [146, 209, 210, 170, 36] – reducing the overhead required to process web page content.

In their 2011 study, Thomas *et al.* [268] proposed a real-time spam URL filtering system called *Monarch*. They conclude their study by stating: *"we show that Monarch can provide accurate, real-time protection, but that the underlying characteristics of spam do not generalize across web services. In particular, we find that spam targeting email qualitatively differs in significant ways from spam campaigns targeting Twitter"*. With that in mind, various studies have focussed their machine learning based detection efforts on social networks [262, 41, 49, 90, 212, 304, 306, 309, 179, 307, 59, 4], for example: by deploying social honeypots and machine learning [156] and mapping the spread of astroturf in microblog streams [238]. Existing studies have also focused their machine learning on Twitter specific features such as looking at redirection chains to detect suspicious URLs [158], analysing suspended accounts [269], detecting automated accounts [58], using social graph models [288], and a system called PhishAri to detect phishing in real-time on Twitter [4].

Section Summary

We see that existing studies into phishing and malware defence can typically be organised into two categories: prevention and detection. Machine learning plays a key role in detection techniques, due to its ability to analyse large data sets and detect patterns.

To defend against phishing and malware attacks, many machine learning techniques include the unique features of social networks, such as Twitter, to improve their accuracy.

The wealth of existing research demonstrates that numerous defence mechanisms already exist to protect people on the internet from phishing and malware attacks. However, the effectiveness of such techniques will depend on their application and the specific scenario in which a user encounters an attack. Therefore these existing defence techniques cannot solely be relied upon for complete protection from cyber attacks.

3.5 Web Browser Phishing Detection

A notable study was carried out by Zhang *et al.* in (2006) [311] in which various browser extensions were tested to see how effective they were at detecting phishing URLs. Two sources of phishing URLs were used: PhishTank and APWG. The tools tested in the study were: CallingID, Cloudmark, EarthLink, eBay, IE7, Firefox, Firefox/Google, Netcraft, Netscape, SpoofGuard, TrustWatch.

One of the main results of this study was that SpoofGuard achieved a detection rate of just over 90% and that Firefox and Google Chrome had a detection rate of 70% when the URL was first extracted, which rose to over 80% after 24 hours. Since Google Chrome was using the Google Safe Browsing blacklist in the experiment, the delayed detection rate is likely to illustrate the time taken for phishing URLs to appear in the blacklist. Since this study, most of the tools that were analysed are no longer available or have not been updated for many years. The main tools to have survived are the web browsers themselves.

A 2012 study by AV Comparatives [23] carried out a series of phishing website detection tests on 5 popular web browsers: Apple Safari 5, Google Chrome 23, Microsoft Internet Explorer 9, Mozilla Firefox 18 and Opera 12. The exepriments were carried out between the 11th and 19th December 2012. A total of 294 phishing websites were used in the tests. All tests were carried out on Windows 7 SP1 64-bit. The study's key findings were that Opera detected the greatest number of phishing websites (94.2% detected), followed by Internet Explorer (82%), Chrome (72.4%), Safari (65.6%), and finally Firefox (54.8%).

3.6 Heuristic Phishing Detection

Due to the previously noted limitations of blacklists, an alternative approach to detecting phishing websites is to automatically assess various features of a website. This assessment produces a probability score of how likely a given website may be a phishing attack. This approach does not require a blacklist to detect phishing websites. Numerous studies have been carried out into automated phishing detection and some of the most relevant studies are explored in this section.

A notable study carried out in 2004 by Chou *et al.* [56] extracted various features of a given website by creating a web browser plugin called SpoofGuard. Their tool assesses two key areas of information about a given website: stateless and stateful page evaluation. Stateless page evaluation looks at the URL (for various signs such as the @ character in a URL), images (to see if logos of another organisation have been used), links (all

	Classified as legitimate	Classified as phishing
Legitimate page	4,131	18
Phishing page	115	565

Table 3.1: Ludl *et. al* (2007): Confusion matrix for page classifier.

links on the page are examined against the URL check) and passwords (does the page require the user to enter a password into a form, if so is it secure?). The stateful page evaluation looks at the domain (does the domain look like a previously visited website?), referring page (has the user come from an email client such as Hotmail, Gmail etc) and image-domain associations (a database of images, such as corporate logos, and their association with domain names). Finally, SpoofGuard evaluates submitted HTTP POST data to determine if the user is sharing a password with a spoof site. All of this data is fed into a scorer which combines all of these tests to produce a phishing probability score. If this score is high enough, the user is presented with a phishing warning.

We have already seen the results of SpoofGuard's detection rate in Zhang et. al's 2006 study [311]. Without the use of a blacklist, the tool was able to detect over 90% of phishing websites; outperforming all other tools in the test. The SpoofGuard plugin was released in 2006 for Internet Explorer although, its use was more of a proof-of-concept. The tool has not been updated since the study and no further versions of the plugin have been released.

The Ludl *et al.* (2007) study [169] (which was previously mentioned in the Phishing Blacklists Effectiveness section) also looked at various on page properties of phishing websites in an attempt to aid phishing detection. The main properties they look at are: forms, input fields, links, whitelist references, script tags, suspicious URL and use of SSL. Their study then continues to build a classification model which uses the J48 algorithm (an implementation of the decision tree algorithm C4.5 [236] from the data mining tool Weka [299]) to classify pages as phishing or safe. Table 3.1 shows the classification quality (confusion matrix) of their model whereby over 80% of phishing websites were correctly identified.

Another study in 2007, by Zhang *et al.* [312], produced a tool called CANTINA that uses TF-IDF, an information retrieval and text mining algorithm, and Robust Hyperlinks to detect phishing websites based on their contents. Their approach essentially uses a lexical signature technique to uniquely identify web pages. The basic idea is that when producing phishing websites, phishers will simply copy the web page they are attempting to masquerade as. By making use of these unique lexical signatures (or

Robust Hyperlinks), CANTINA can then identify when a website is a replica of a legitimate website based on its signatures' similarity to the legitimate website. Their results show that CANTINA was able to detect around 90% of phishing websites, with 1% false positives.

In 2007, Garera *et al.* [91] looked specifically at the structure of URLs to detect phishing attacks. Their approach argues that most phishing URLs use similar techniques to deceive users. Their study defined four phishing URL types:

1. **Obfuscating the host with an IP address:** hostname is replaced with an IP address and the organisation being masqueraded as normally placed in URL e.g.:
   - *http://210.80.154.30/test3/.signin.ebay.com/ebayisapidllsignin.htm*
   - *http:// 0xd3.0xe9.0x27.0x91:8080/.www.paypal.com/uk/login.html*
2. **Obfuscating host with another domain:** valid domain with organisation being masqueraded as in URL. Often relies on redirect e.g.:
   - *http://21photo.cn*
   - *https://cgi3.ca.ebay.com/eBayISAPI.dllSignIn.ph*
   - *http://2-mad.com/hsbc.co.uk/index.html*
3. **Obfuscating with large host names:** contains masqueraded hostname in URL along with other words and the actual domain e.g.:
   - *http://www.volksbank.de.custsupportref1007.dllconf.info/r1/vm/*
   - *http://sparkasse.de.redirector. webservices.aktuell.lasord.info*
4. **Domain unknown or misspelt:** makes use of misspelt domains that look similar to masqueraded domain or domains that are unknown e.g.:
   - *http://www.wamuweb.com/IdentityManagement/*
   - *http://mujweb.cz/Cestovani/iom3/SignIn.html?r=7785*

The study acknowledged that URL detection based on these four types alone was not particularly robust as it could be easily evaded by an attacker. Therefore, their study also looked at various other website features which were grouped into: Page Based (including Google's Page Rank score – in theory: the higher the Page Rank score, the more reputable the website is), Domain Based (is the domain in a previously generated White List of trusted organisations), Type Based (using previously defined obfuscation types as features) and Word Based (e.g. words such as "login" and "signin" often appear in phishing URLs). The resulting classifier produced an accuracy of 93.4% although an extensive analysis of this accuracy is not provided in

Type	Total Number of URLs	Percentage
1	3,110	33.32%
2	1,616	17.30%
3	4,337	46.46%
4	273	2.9%

Table 3.2: Garera *et. al* (2007): Distribution of Obfuscation Types.

the research paper. The classifier was used on the Google Safe Browsing toolbar to detect how many phishing URLs fell into each of the URL types defined above. Their findings are shown in Table 3.2.

In 2005, Liu *et al.* [297] proposed a solution that looks at the visual similarity between websites to determine if a site is phishing. The similarity assessment used in this technique is carried out by measuring three metrics:

1. **Block level similarity:** A website is essentially made up of blocks (text or images); the contents of these blocks can be compared. Text blocks compare features such as colours, border style and alignment, etc., and image blocks compare features such as alternative text, dominant colour, and image size, etc.

2. **Layout similarity:** Having measured the block level similarity, what is the similarity of the layout of those blocks on the web page?

3. **Overall style similarity:** This assesses the visual style of the website using features such as font family, background colour, text alignment, and line spacing.

Their study tested this detection technique against 8 known phishing websites and 320 legitimate websites (to test for false positives). Two tests were carried out with a similarity threshold of 0.9 and 0.7 respectively. The first produced no false alarms but missed one and the second produced 4 false alarms but none missing.

The Liu *et al.* study was further improved in 2008, when Medvet *et al.* [181] provided a similar solution. The approach used by Medvet *et al.* was different in that it was inspired by two previous anti-phishing solutions: AntiPhish [147] (a tool that warns user's when they share the same login credentials across multiple websites) and DOMAntiPhish [241] (a tool that builds on AntiPhish by comparing the Document Object Model of websites to reduce warnings when user's enter the same login credentials on multiple trusted websites). Basically, the approach by Medvet *et al.* takes the three metrics from the Liu *et al.* solution to produce a signature for a given website. A legitimate website's signature could then be compared to its

phishing version's signature to determine if the phishing website was a copy of the original. These signatures could then be used in a solution such as AntiPhish to warn when a user is about to share sensitive information with a website that has copied a trustworthy website. The results from this signature based detection technique were that 100% of 140 tested phishing websites pairs were detected (when compared to their legitimate copied websites). 27 false phishing websites were also tested for which only 2 were falsely detected as phishing.

3.7 Warning Effectiveness

As mentioned in the Introduction section of this chapter, phishing detection can be rendered useless if it is not communicated and presented to the user in an effective manner. Alerts and warnings that confuse the user may be ignored, the phishing detection techniques wasted, and the user may continue onto a phishing website to share their sensitive data. This section aims to explore existing research into the effectiveness of such warnings.

In 2006, Wu *et al.* [300] assessed five web browser toolbars to determine how effective their warnings were at preventing users from visiting phishing websites. In their study, the five web browser toolbars were grouped into three simulated toolbars based on the types of information they display:

1. **Neutral Information:** information about the website being visited is shown such as: domain name, hostname, registration date, and hosting country. This information then allows the user to make their own decision on whether the site is safe

2. **SSL Verification:** information simply highlights if the website uses SSL encryption or not

3. **System Decision:** this toolbar makes a decision on behalf of the user and displays a red warning if it believes a website is a phishing attack

Subjects were then given the task of protecting login credentials for a fictitious user and told to act as a personal assistant to this user. One of the main findings in this study was that a typical user's primary goal is not security; they are usually attempting to carry out some other activity. Therefore, having a warning message appear in a toolbar was not enough to encourage users to take evasive action. Most users simply ignored the warnings or did not understand what they meant – and many users simply did not understand how sophisticated some phishing attacks can be so were not aware of what to look out for.

A further study in 2008, by Egelman *et al.* [76], took the Wu *et al.* study one step further by comparing passive and active warnings. In their

experiments, passive warnings are defined as warning that appear on the screen but do not obstruct the user's activity. An active warning, on the other hand, would interfere with the user's browsing experience and force the user to take note of the warning. The results from this study showed that 79% of people chose to heed active warnings and not visit phishing websites compared to just 13% of users that were presented with passive warnings.

Akhawe *et al.* (2013) [6] conducted a large-scale field study into the effectiveness of web browser warnings. They define the metrics: passive and active warning messages. In their experiments, *passive* warnings are defined as warning that appear on the screen but do not obstruct the user's activity. An *active* warning, on the other hand, would interfere with the user's browsing experience and force the user to take note of the warning. The results from the study showed that 79% of people chose to heed active warnings and not visit phishing websites compared to just 13% of users that were presented with passive warnings.

Felt *et al.* (2015)[82] explored the concepts of comprehension and adherence in their study to improve SSL warnings. The study looks at a number of features that are designed to communicate the warning effectively to users. The features are: technical jargon (text should user simple, non-technical language that are aimed at a broad user base); reading level (reading level should target all users – 6th grade reading level e.g. SMOG formula [178]); brevity (large quantities of text look like they will take effort to read – warnings should be brief); specific risk description (describe the risk explicitly and unambiguously); illustration (images can reduce time to comprehend information and can make warnings more attention-grabbing and memorable); and risk level (warnings must stand out from the surrounding environment; red warnings are more effective that black or green warnings).

3.8 Measurement Studies

In this section we will explore key existing measurement studies that investigate topics related to this thesis. These measurement studies explore the topics of: URL shorteners, information credibility, networks, and phishing and malware attacks. Some of these studies carry out their measurements on Twitter, whilst other studies implement their measurements on the web in general.

By exploring these topics of measurement studies, we aim to assess how existing studies carry out their experiments and to explore what methodologies they use. This provides inspiration for the measurement experiments that are designed and deployed for this thesis, which we will explore in

Chapter 4: *Design & Implementation*. These existing measurement studies can also provide context for the results we see from the research carried out for this thesis, and also help to understand the landscape we are measuring.

3.8.1 URL Shorteners

URL shorteners play an important role on Twitter. Due to the 280 character limit per tweet (raised from 140 in 2017 [275]), URL shorteners keep text count to a minimum. Although Twitter introduced its own URL shortener in 2011 [279], users can still shorten their submitted URLs via any number of shorteners – forming a redirection chain for each shortener. This section explores key papers related to the use of URL shorteners to understand what URL shorteners are popular both on the internet in general and on Twitter, how phishers use URL shorteners, how URLs propagate on the web and Twitter, and more.

Antoniades *et al.* (2011)

Antoniades *et al.* (2011) [16] carried out a large-scale crawl of the URL shortening services *bit.ly* and *ow.ly* and examined their use on Twitter. The authors collected more than 20 million URLs from Twitter – of which 87% were shortened. The study was carried out before Twitter introduced its own URL shortening service whereby **all** tweeted URLs are now shortened via *t.co*.

The study's results show that 50% of tweeted URLs, analysed during their study, were shortened via *bit.ly*, 4% via *tl.gd*, 3.5% via *tinyURL*, and 1.5% via *ow.ly*. Results show that 60% of users arrived at tweeted *bit.ly* URLs from non-web applications (email, IM, apps, phone, direct) and 23% via twitter.com. This browsing model suggests a "word of mouth" propagation of short URLs posted to Twitter.

The study's methodology involves using Twitter's Search API to collect URL-containing tweets and brute-forcing Bitly and Owly's available key-space to collect short URLs. Due to the Twitter API's rate limit, the authors query the API once every 5 minutes for tweets that contain HTTP URLs; each request retrieves up to 1,500 tweets and goes back no more than 7 days in time. This method collected more than 20 million tweets containing HTTP URLs. The bit.ly brute-force collection process involved searching the entire key-space [0-9, a-z, A-Z] for hashes up to 3 characters in length, since bit.ly uses random hashes. Ow.ly does not use random hashes, but serially iterates over the available short URL space. Therefore the authors collected a full set of ow.ly short URLs created for a 9-day period. Ow.ly does not provide an API for URL data, therefore results are only shown for bit.ly.

Chhabra *et al.* (2011)

Chhabra *et al.* (2011) [55] investigated phishers that use URL shortening services to masquerade phishing URLs on Twitter. Their study sampled 6,474 blacklisted phishing URLs from the blacklist PhishTank that had been shortened via *bit.ly*. They showed that the amount of space saved by using a URL shortener was, on average, 39% and that 50% of phishing URLs saw a space gain of 37% or less – for context: Antoniades *et al.* showed, for generic URLs, a 91% space gain [16]. This implies that URLs shortened by phishers are not as long as generic URLs; favouring the hypothesis that URL shorteners are not only used to make communication terse but to obfuscate the URL to escape any scrutiny which is based on only URL text.

The study's methodology consists of 3 key phases:

1. Fetch the PhishTank database for the year 2010 and extract URLs voted as confirmed phishing – resulting in 1,18,119 URLs
2. Query Bitly API for the global shortened URL for any matching URLs from step 1 dataset, also query the domain name for every phishing URL in step 1 dataset – this resulted in 6,474 Bitly URLs for "phishing", with 3,692 exact matches and the rest domain matches
3. For every Bitly URL from step 2 dataset, query Bitly API for clicks, clicks-by-day, countries, and referrers – for referrals from Twitter they use Twitter's API to fetch additional information

Wang *et al.* (2013)

Wang *et al.* (2013) [289] measured short URL spam on Twitter by analysing click traffic of over 600,000 tweeted Bitly URLs. Tweeted Bitly URL click traffic analysis shows that the majority of tweeted Bitly clicks are from direct sources (i.e. email messages, instant messages, applications, etc) and that spammers utilise popular websites, such as Twitter, Facebook, YouTube, etc, to attract more attention by cross-posting links.

Their methodology is as follows: query Twitter API every minute to collect tweets, and associated Twitter users, containing trending topics – about 600 topics between November 2009 and February 2010. Users or tweets that were marked as suspended or removed due to violating Twitters terms of service were marked as spam in the dataset – this is their ground truth. They also check the short URLs collected in their dataset (including all URLs in the redirection chain) against public blacklists: Google Safe Browsing, McAfee SiteAdvisor, URIBL, SURBL, and Spamhaus – labelling matches as spam, to validate the ground truth.

The study notes that the lag effect of blacklist validation is not a problem due to their own delay between generating and labelling their dataset. Wang *et al.* do not state the delay time between their data collection and labelling

for spam URLs – therefore it is possible that there could indeed be a lag in blacklist validation which would mean their labelled dataset may not be accurate. All short URLs were extracted from their dataset; Bitly made up about 57%. The study then queried Bitly's API to collect URL traffic data on URLs in the dataset of suspended or removed Twitter accounts.

Nikiforakis *et al.* (2014)

Nikiforakis *et al.* (2014) [211] investigated the ecosystem of ad-based URL shortening services from a security and privacy perspective. Their study explored ad-based URL shorteners on the whole web and was not specific to Twitter. They discovered that ad-based URL shorteners are susceptible to a number of vulnerabilities, such as tabnabbing and link hijacking which can lead to drive-by-downloads, browser-exploits, scams, phishing attacks; and privacy concerns through the use of sequential (i.e. predictable) URLs and URL leaking through HTTP Referrer headers.

The authors discovered 892 malicious web pages that used an ad-based URL shortener – 81% of these came from **adf.ly** links. In their dataset of 4,300 users that had clicked on an ad-based URL shortener, 50% of users were running outdated software – 25% of which were susceptible to at least one identified exploit. The study also analysed 29,709 ad-based short URLs and discovered that the majority of these URLs were published on low-reputation websites; 26% of these falling under the category of "Blogs / Web Communications" and a further 26% of these short URLs redirected back within the ad-based short URL ecosystem; bringing users into an endless loop of ad pages.

After analysing the landing pages of ad-based short URLs, the authors discovered that the majority of URLs point to high-reputation websites such as file-hosting services, YouTube videos, Facebook apps. Although less than 10% of the landing pages with file hosting services were available. These landing page results suggest that many ad-based URL shortening services are used as "wrappers" to promote popular content that is already free and widely available to the public – and that the creators of these ad-based URLs lure their "victims" with appealing content to drive traffic (and thus ad revenue for profit) without providing the promised content to link-following users.

The key methodology used in the Nikiforakis *et al.* study involves downloading historical data from the blacklist Wepawet [63] between 1st January and 31st July 2013 and searching for ad-based URLs within this dataset. To research consumers of ad-based URLS the authors paid for adverts on 2 of the popular ad-based URL shortening services: $5 for 1,000 impressions from the US and $5 for 5,000 impressions worldwide on **adf.ly**; and $6.60 for 2,000 UK impressions on **linkbucks.com**. They then used

partial fingerprint matching (IP address, browser's User agent, and list of plugins) and cookies to investigate how lucrative ad-based would be for an attacker.

Out of the 8,000 impressions paid for, the authors only collected 4,300 fingerprints – mainly due to crawlers, scrapers, and bots lacking proper Javascript support. They then compared users' browser set-ups to a list of known browser exploits [62]. In order to research the producers and landing pages of ad-based URL shorteners the authors queried Bing's Search API for URLs that contained a set of 10 ad-based URL shorteners, then used the PantomJS [225] scraper. They used WOT Reputation API [292] and Trendmicro's Site Safety Center service [273] to categorise the websites.

Maggi *et al.* (2013)

Maggi *et al.* (2013) [171] examined how effective short URL countermeasures are at preventing users from shortening malicious URLs. Again, this study was not specific to Twitter, as it measured general URL shorteners across the web. They found malicious URL countermeasures to be ineffective and trivial to bypass. The study also carried out a large-scale collection of HTTP interactions that originated when internet users accessed websites that contained short URLs. They monitored 622 distinct URL shortening services between March 2010 and April 2012, collecting 24,953,881 unique URLs. The study's key findings show that relatively small numbers of internet users encountered malicious short URLs. During their second year of measurement, the authors noticed in increase in the number of drive-by-download short URLs.

Subsection Summary

The studies explored in this subsection shed light on how URL shorteners are used. *Bitly* tends to dominate the url shortening market as the most frequently used shortening service. We see how how phishers can take advantage of URL shortening services to increase their coverage and gain more victims. We also see URL shortening as a popular technique used in an attempt to hide phishing URLs. Exploring different types of URL shortening services shows that ad-based URL shorteners may provide hidden motives. Not only are URLs, that are shortened with ad-based URL shortening services, ripe with phishing URLs, people that share such URLs may be more motivated by profits than their users' best interest.

When exploring the methodologies employed by these existing studies, we see the popularity of APIs and blacklists as data sources and ground truths. Many of the studies lookup large datasets of URLs, obtained from sources such as Twitter's Stream API, in one or more blacklists. This allows the

studies to determine how many of their sampled URLs fall into certain categories (e.g. phishing). We also see effective use of the *Bitly* API to retrieve and analyse statistics for certain URLs – such as clicks and referrers – which can help to determine a URL's popularity and therefore its impact during a cyber attack campaign.

3.8.2 Information Credibility

This section aims to provide a summary of existing research that explores information credibility on Twitter. Although not directly related to this thesis' methodology, the topic of information credibility links into cases where phishing and/or malware URLs are spread via "trending topics". This relates to the previous section (3.8.1: *URL Shorteners*) where we saw how "word of mouth" propagation of shortened spam URLs ties into information credibility.

We will explore the theory that Twitter users whom fall victim to so-called "fake news" on Twitter may be at risk of phishing and malware URL attacks. Although most of the studies we examine in this section involve creating novel machine learning classifiers, they all require training data. This training data can provide measurements of information credibility on Twitter – which will be our main focus of these studies. The measurement data in these existing studies illustrates and highlights the prevalence of misinformation on Twitter.

Twitter is an open platform whereby anyone can tweet a message publicly. The means it can be difficult for Twitter users to determine what information is accurate and reliable on the social network; the spread of misinformation and so-called "fake news" is a common problem in contemporary culture [67, 155, 8] (*e.g.* misleading Coronavirus map (February 2020) [33], misleading Australia wildfire maps (January 2020) [32], out-of-context photos (February 2020) [165]). In April 2019, the UK government launched a sub-committee to tackle disinformation [123] – although the problem of so called "fake news" in society is not new [260].

Castillo *et al.* (2011)

Castillo *et al.* (2011) [53] investigated the information credibility of news propagated through Twitter by using machine learning techniques to automatically classify tweets related to "trending" topics as either credible or not credible. The study used human assessment, via Mechanical Turk [14], to evaluate their machine learning classifier. Their results show that there are measurable differences in the way messages propagate on Twitter, that can be used to classify messages automatically as credible or not credible,

with precision and recall of their machine learning classifier in the range of 70% to 80%.

The study's methodology for gathering training data is as follows: the authors use Twitter events detected by Twitter Monitor [177] (an online tool that detects bursts in frequency of sets of keywords and returns a keyword-based query for each event) during a 2-month period. For each event detected by Twitter Monitor, the authors collected all tweets during a 2-day window centred on the peak of each burst. The authors do not state if they used Twitter's API to search for this information. Each set of tweets, produced from each Twitter event, corresponds to a "topic". They collected over 2,500 topics. The authors then randomly selected 383 topics from the Twitter Monitor collection to be categorised, by humans via Mechanical Turk, as either "news" (statements about a fact or actual event) or "chat" (messages based purely on personal/subjective opinions) manually. Topics were grouped into sets of 3 per task and 7 different human evaluators on Mechanical Turk were required to evaluate each task, during a 10-day period. Each topic was assigned a label if 5 out of 7 evaluators agreed on the label – otherwise the topic was labelled as "unsure".

Results of this labelling process show that: 29.5% (113 cases) of topics were labelled as "news" and 34.9% (134 cases) as "chat" while 35.6% (136 cases) were labeled as "unsure". The authors then ran an event supervised classifier on 2,524 cases detected by Twitter Monitor, whereby 747 cases were labelled as "news". These labels were manually verified by humans, via Mechanical Turk, whereby evaluators assessed each case from a sample of 10 tweets per case – again, 7 evaluators were assigned per task and 5 out of 7 evaluators must agree per label, otherwise it was labelled as "ambiguous". Results showed that 41% (306 cases) were labelled as "almost certainly true", 31.8% (237 cases) as "likely to be false", 8.6% (65 cases) as "almost certainly false" and 18.6% (139 cases) as "ambiguous".

Although the Castillo *et al.* study approaches the problem of information credibility on Twitter from a machine learning perspective (i.e. creating a classifier to automatically detect credible news), their methodology for collecting ground truth data for training the classifier itself produces an insightful measurement. In their training data: 35.6% of trending Twitter topics were classified as "chat" versus 34.9% as "news" – a fairly balanced ratio. However, when testing their classifier: 747 out of 2,524 (30%) trending topics on Twitter were automatically classified as "news", with 237 (32%) of these topics being manually labelled by humans as "almost certainly true". This means that only 237 (9%) of the original 2,524 Twitter topics were deemed newsworthy (i.e. statements about a fact or actual event) – a low proportion of topics on Twitter. This provides evidence to suggest that over 90% of trending topics, discovered by the authors of the study, were either "chat" (messages based purely on personal/subjective

opinions) or it was not clear if they were newsworthy. In summary, the Castillo *et al.* study data shows that a low proportion (9%) of trending topics on Twitter are regarded as credible news.

Gupta *et al.* (2012)

Gupta *et al.* (2012) [112] analysed the credibility of information in tweets corresponding to 14 high impact news events of 2011 around the globe. They analysed 5,578 tweets and found, on average, 30% of tweets posted about an event contained situational information about the event, but only 17% of these tweets contained credible information. 14% of tweets were spam (i.e. tweets contained words belonging to trending topics but were not related to the event). 50% of tweets relating to an event express personal opinion or reactions and therefore provide no useful information. The study also used regression analysis to identify important content and sourced based features, which they use to train a supervised machine learning and relevance feedback approach, to rank tweets according to their credibility score – therefore predicting the credibility of information in a tweet.

The study's methodology involved querying Twitter's Trends API every 3 hours to return the top 10 trending topics on Twitter. They then used Twitter's Streaming API to collect tweets corresponding to these trends as query search words. They collected tweets relating to a topic until it was no longer trending. This methodology is similar to Castillo *et al.* (2011) [53], except the Twitter Trends API is used as a source for trending topics. Gupta *et al.* (2012) [112] collected over 35 millions tweets (of which over 22 million were singleton and over 13 million were re-tweets/replies) from more than 6 million Twitter users. This dataset contained over 4 million tweets with URLs and 3,586 unique trending topics. Data was gathered from the time period 12th July 2011 to 30th August 2011.

The authors then shortlisted their dataset of 3,586 trending topics down to 14 major events that occurred globally during their data collection period. Each event had 1 or more trending topics associated with it. For each event, the authors considered tweets containing the words in trending topics to be the set of tweets for that event. Criteria for event selection included: news events like political, financial, natural hazards, terror strikes and entertainment news. Each event had at least 25,000 tweets, and topics were trending for at least 24 hours. Human annotators were then used to evaluate the credibility of 500 tweets per 14 topics (7,000 tweets in total) – each tweet was annotated by 3 different humans. To check the reliability of their human annotation results, they used the Cronbach Alpha score to calculate the inter-annotation agreement for all 7,000 tweets, producing a value of 0.748 ($>$0.7 implies high agreement between annotators [153]). The authors labelled each tweet when at least 2 annotators agreed on the

same label and discarded all tweets when all three annotators gave different labels. This resulted in a dataset of 5,578 tweets.

Again, the Gupta *et al.* (2012) [112] study focuses on tackling information credibility from the perspective of building a machine learning classifier. However, again, training data is required which provides an insightful measurement of information credibility on the social network. In their results we see that 17% of tweets relating to a news event contain credible information. 14% of tweets contained trending topics that were unrelated to the news event (spam) and 50% of tweets expressed personal opinions or reactions. This study shows that a low proportion (17%) of tweets relating to trending topic are credible. Although the authors verify that the human annotators agree on the evaluations, the authors do not verify if the annotations are correct – therefore the ground truth of their data may not be accurate. The authors do acknowledge the limitations of using human annotation to establish ground truth.

Ghosh *et al.* (2013)

Ghosh *et al.* (2013) [96] investigated how to sample the data generated from Twitter users by analysing the differences between sampling tweets from random versus "expert" users (i.e., "Twitter users whose followers consider them to be knowledgeable on some topic"). This is based on existing research which suggested that "topical experts are often the primary drivers of interesting discussions on Twitter" [20].

During a 1-month period, Ghosh *et al.* sampled over 63 million tweets from around 500,000 "expert" users and a subsampled set of over 21 million random users. From a random sample of 1 million tweeted URLs: 1,520 URLs (tweeted by 1,447 users; 0.140%) from the random sample were malicious versus 129 URLs (tweeted by 46 users; 0.022%) in the "expert" sample. Of these 1,447 users, 501 (34.6%) were suspended by Twitter. Out of those 501 users, 468 (93%) joined Twitter during the same month the measurement was carried out. A "significant fraction" of the remaining accounts had posted over 100 malicious URLs in their lifetime. In addition to these measurements, the authors also compared the datasets for diversity, timeliness, and trustworthiness of the information contained within them.

Overall, their results show that "experts'" tweets are "significantly richer in information content (whereas close to 90% of the random sample is devoid of topical information), cover more diverse topics, and more popular content. Experts' tweets are also more trustworthy (contain considerably fewer malicious URLs and spam) and they often capture breaking news stories marginally earlier than random sampling".

The methodology used by Ghosh *et al.* for sampling random and expert Twitter users is as follows:

1. Twitter's Stream API is used to to collect a 1% random sample of all tweets during December 2012
2. To generate a set of expert Twitter users: crawl the first 50 million Twitter user accounts (in terms of creation date), using Twitter's API. To filter these users down to expert users: use the List-rank [94, 54] metric whereby users are ranked according to how many Twitter Lists they reside in – the authors require a user to appear in at least 10 Lists for inclusion in their dataset

Their resulting dataset, based on this List-rank criteria, consists of 584,759 "expert" Twitter users. The study then collects all tweets posted by this set of expert Twitter users during December 2012. The authors acknowledge that this set of "expert" users is biased towards those that joined Twitter early and therefore countries where Twitter first became popular (e.g. USA). However, they state that their aim is to test their Twitter sampling methodology and "*not* to identify all experts or obtain an unbiased sample of experts in Twitter".

During the month of December 2012, their study collected 63,497,081 tweets from 427,674 "expert" Twitter users, and 124,253,878 tweets from 30,046,582 random Twitter users – which was subsampled down to 63,497,081 tweets from 21,941,041 Twitter users so that the number of tweets was comparable to the "expert" dataset. To check for malicious URLs: the authors randomly selected 1 million URLs from each of the two datasets and checked how many were blacklisted. They did this by looking up each URL via Google's Safe Browsing API and also checked for malware "warning pages" where URL shortening services such as bit.ly and tinurl were used.

Ghosh *et al.* (2013) provide evidence that "expert" Twitter users tweet considerably fewer malicious URLs compared to random users. However, their dataset of "expert" users is based on those that appear in 10 or more Twitter Lists and that were one of the first 50 million to register on Twitter. This does bias their definition of "expert" users towards early adopters of Twitter and, typically, American citizens. The primary focus of this research paper is not a measurement study, but an introduction to a novel sampling method for Twitter data and a proof-of-concept of this sampling methodology. However, the results from this study do support the idea that users whom supply misinformation, or so-called "fake news", on Twitter are also more likely to include malicious URLs within their tweets. Therefore, Twitter users that fall for misinformation may be at an increased risk of phishing and malware attacks.

Gupta *et al.* (2013)

Gupta *et al.* (2013) [113] investigated the role of Twitter, during Hurricane Sandy (2012), to spread fake images about the disaster. They identified

10,350 unique tweets containing fake images that were circulated on Twitter, during Hurricane Sandy. Their study discovered that 86% of tweets spreading the fake images were retweets. They showed that the top 30 of 10,215 users (0.3%) resulted in 90% of the retweets of fake images. Also, network links such as follower relationships of Twitter, contributed very little (only 11%) to the spread of these fake photos' URLs. With this data, the study goes on to produce a machine learning classifier to detect fake images of Hurricane Sandy, achieving an accuracy of 97%. They show that tweet based features (as opposed to user based features) were more effective at distinguishing real images from fake images of Hurricane Sandy.

The methodology employed by the Gupta *et al.* study is similar that of Ghosh *et al.* (2013), in that they use Twitter's Trends API every hour to get current trending topics, then use these topics as query search words for Twitter's Streaming API. The Gupta *et al.* measurement duration was 20 months. Tweets containing the words "hurricane" and "sandy", between 20th October and 1st November 2012, were extracted from their sampling setup. This dataset contained 1,782,526 tweets from 1,174,266 million unique users – of which 622,860 tweets contained URLs. The study's ground truth for fake images was a set of "articles, tweets, and blogs" [199, 15, 295]; a prominent data source used was a Guardian website listing fake Hurricane Sandy images [111]. The study discovered 10,350 tweets with fake images, 10,215 users with fake images, 5,767 tweets with real images, and 5,678 users with real images.

The Gupta *et al.* (2013) study shows that, in their dataset, misinformation on Twitter – fake images of Hurricane Sandy – was created by a small group of users but then spread by retweets – mostly by users that did not follow the creator. This case study illustrates how misinformation propagates on the social network and how fake news is spread by large numbers of users. Although the Ghosh *et al.* study focuses on fake images of Hurricane Sandy as their "malicious" content, the study demonstrates how malicious content in general (e.g. phishing and malware URLs) might spread on Twitter.

Giglietto *et al.* (2014)

Giglietto *et al.* (2014) [97] investigated Twitter conversations that occur whilst people watch a television programme on another device. They sampled 2,489,669 tweets that contained hashtags relating to 11 popular Italian political talk shows. The result of categorising tweets in their dataset was: 59% personal opinions, 19% attention-seeking, 15% pure information, 5% emotion.

The study's key methodology uses Twitter's firehose – which contains 100% of tweets – via DiscoverText GNIP importer [138]. They sampled

tweets between 30th August 2013 and 30th June 2013 that contained hashtags relating to one of 11 political talk shows on the Italian free-to-air broadcasters. This dataset spans the complete series (1,076 episodes) of the 11 talk shows. They then analysed their dataset based on activity per minute and peaks detection to focus on content at a deeper level.

The study shows that, in their dataset, only 15% of tweets contained pure information that was not tainted by personal opinion – although they do not verify the accuracy of these information tweets. Also, it is important to note that their dataset consisted of tweets relating to political talk shows. This sort of TV show is perhaps more likely to generate comments that are opinion based and therefore may possibly bias the results somewhat. Their results are similar to those in the Gupta *et al.* (2012) study, whereby 17% of tweets were deemed "credible", and in the Castillo *et al.* study, whereby 9% of tweets were "newsworthy".

SUBSECTION SUMMARY

This subsection has explored key existing studies relating to information credibility, misinformation, and the spread of so called "fake news" on Twitter. We saw how trending hashtags, and high impact events, are commonly hijacked by Twitter users aiming to spread misinformation – with only 9% to 17% of tweets relating to trending topics being credible. Ghosh *et al.* provided empirical evidence that "expert" Twitter users tweet considerably fewer malicious URLs compared to "random" users – supporting the notion that Twitter users whom fall for misinformation may be more susceptible to phishing and malware attacks. We also saw how quickly misinformation can spread on a social media platform – such as in the case study of fake images of Hurricane Sandy.

Overall, we see that, due its open platform nature whereby anybody can broadcast a message, Twitter will always attract users wishing to spread misinformation. It can be difficult for general Twitter users to distinguish "fake news" from credible information on the platform, and this misinformation can be ripe with malicious content. Therefore it is vital that Twitter encourages and highlights credible information on its platform and protects users from phishing and malware attacks.

3.8.3 Networks

The spread of information on Twitter, whether that be, for example, misinformation or phishing URLs, can be traced through various links of connected users. By investigating these links of users, or networks, we can understand how such information spreads on the social network in the first

place. This section explores networks on Twitter to explore topics such as how URLs are spread on the social network, analysing spammers' social networks, investigating link farming, and how follower markets operate.

Rodrigues *et al.* (2011)

Rodrigues *et al.* (2011) [240] investigated the concept of "word-of-mouth" as a way for Twitter users to share and discover URLs. Their key findings are:

- URLs spread via word-of-mouth to a large proportion of Twitter users – in some cases millions of users
- Popular URLs are usually started by multiple "initiators" (tree root; users that independently share a URL)
- Unpopular domains can gain exposure via word-of-mouth
- URL propagation trees on Twitter are wider than they are deep (in contrast to URL propagation trees on email)
- Users who are geographically close together are more likely to share the same URL
- Nearly 90% of URLs the study's dataset are introduced by a single Twitter user, without any retweets
- Whereas, URLs with retweets typically gained a 3.5 times larger audience than those without

In terms of their methodology: the study uses the same dataset from Cha *et al.* (2010) [54] which was gathered when Twitter agreed to whitelist 58 of the study's servers in August 2009. The authors crawled through all Twitter user IDs ranging from 0 to 80 million; no Twitter users with an ID greater than 80 million existed at that time. The resulting dataset consisted of 54,981,152 Twitter users; 1,963,263,821 directed follower links; and all 1,755,925,520 public tweets posted by these Twitter users between March 2006 and September 2009 (excluding 8% of Twitter users set to private). Nearly 75% of these URLs were link shorteners – although their methodology to determine if a URL is a shortener was simply if the final URL is different from the tweeted URL.

The study then focuses on 2 datasets; subsets from their large dataset, of 1-week periods (to mitigate link rot in URL shorteners). Dataset 1 (D1), Jan 1-7 2009, consists of 1,239,445 distinct URLs; 6,028,030 tweets; 295,665 retweets; and 995,311 users. Dataset 2 (D2), Apr 1-7 2009, consists of 4,628,095 distinct URLs; 17,381,969 tweets; 1,178,244 retweets; 2,040,932 users. Top domains were:

- 8.5%: Twitpic.com
- 3%: Blip.fm
- 2.1%: YouTube.com
- 2.1%: Plurk.com
- 1.4%: Tumblr.com

The study observed 30 URL shortening services in total. The top URL shortening services were:

- 68% (D1), 81% (D2): Tinyurl.com
- 11% (D1), 24%24 (D2): Bit.ly

Interestingly, between January and April 2009 **bit.ly** doubled its presence. In their dataset, since September 2009, more than 30% of tweets contain URLs – equivalent to 1.3 million distinct URLs per day.

After categorising URLs, they consisted of:

- 10%: Photos
- 4%: Music
- 3%: Videos
- 1%: news
- 0.3%: applications

The study also discovered that "word-of-mouth gives all URLs and content a chance to become popular, independent of popularity of the domain it comes from". URLs from popular domains reached, on average, 49,053 users while unpopular domains reached 1,107 users. Therefore offering URLs from domains that are unheard of a fair chance to gain popularity via word-of-mouth sharing on Twitter.

Yang *et al.* (2012)

Yang *et al.* (2012) [305] carried out an empirical analysis of the cyber criminal ecosystem on Twitter. By exploring inner social relationships of 2,060 criminal accounts, they discovered a community whereby criminal accounts are socially connected – with a hub of accounts in the centre of the graph that are more inclined to follow criminal accounts. The study also explores outer social relationships between criminal accounts and their friends outside the criminal community, categorised into:

- Social butterflies: accounts with very large followers and followings, whereby 48% of accounts follow back, compared to 1.5% on normal accounts

- Social promoters: large following-follower ratios, large following numbers, and high URL ratios
- Dummies: have few tweets but many followers

In terms of methodology, their dataset is the same used in their previous study [304]: they used Twitter's Streaming API to crawl Twitter accounts in multiple rounds to mitigate sampling bias. In each round, their crawler randomly selects 20 seed Twitter accounts and collects all those seed accounts' followers and followings, this process is then repeated. In each round the 40 most recent Tweets and URLs are collected. Using this sampling method to gather data from April to July 2010, the study collected: 485,721 Twitter accounts with 791,648,649 followings; 855,772,191 followers; 14,401,157 tweets; and 5,805,351 URLs – recording only the final landing page for any redirections.

The authors discovered 10,004 accounts that posted malicious URLs, as detected with GSB and a high-interaction client honeypot implemented using Capture-HTC [124]. Of this dataset, the authors only included accounts with a spam ratio (number of spam tweets to total number of tweets) greater than 10%, resulting in 2,933 accounts. They then further manually verified these spam accounts, resulting in 2,060 spam accounts, which are then used in their dataset for the social analysis. The study also developed a criminal account inference algorithm.

The Yang *et al.* study illustrates how criminal users on Twitter are well connected to other criminal users, therefore forming criminal networks. It shows how legitimate Twitter users may be exposed to content and URLs that are promoted by these criminal networks.

Ghosh *et al.* (2012)

Ghosh *et al.* (2012) [95] investigate link farming on Twitter, whereby users acquire large numbers of followers to increase their audience, perceived influence, and ranking. They analysed over 41,352 spammer Twitter accounts and discovered that link farming is wide spread on the social network – 27% of all 54,981,152 Twitter accounts in their study had been targeted by these spammer Twitter accounts. They also discovered that the majority of spammers' links are farmed from a small fraction of Twitter users ("social capitalists") who follow back anyone that follows them. The authors propose a method to rank spammers which penalises users for following a large number of spam accounts on the social network.

In terms of methodology, the authors use the same dataset from the Cha *et al.* (2010) [54] study (same as the Rodrigues *et al.* study), consisting of 54,981,152 Twitter user accounts. To detect spammers, the authors rely on Twitter's policy of suspending accounts that have breached its rules [278]. In

February 2009 their crawler scanned all Twitter accounts in the dataset and selected those whose profile page redirects to *http://twitter.com/suspended*; collecting 379,340 such accounts that were suspended between August 2009 and February 2011.

Their method of spam verification for these accounts is if *bit.ly* or *tunyurl* short URLs have been posted by these users that lead to warning pages by these URL shortening services. This method resulted in the verification of 41,352 suspended accounts that had posted at least one shortened, blacklisted URL – this dataset is used for their study.

It is important to note that 76% of the top 100,00 link farmers in their study were legitimate Twitter users. They discovered that these "social capitalists" are highly likely to follow back any user that follows them, because this is a good social strategy to gain an audience and connect with more people. These users include many successful bloggers and domain experts. Although legitimate link farmers exhibit very different network connectivity than spammers, many of the legitimate link farmers unknowingly connect with spammers – which allows the spammers to increase their reputation and exposure. The study does proposes a ranking system to discourage "social capitalists" from colluding with spammers.

The Ghosh *et al.* study shows how easy it is for spammers on Twitter to gain influence and a large audience. By exploiting the reciprocality of legitimate "social capitalists" Twitter users, spammers can expand their networks with relatively low effort. This then gives the spammers freedom to share any content they so choose, such as malicious or phishing URLs – depending on their motives.

Stringhini *et al.* (2013)

Stringhini *et al.* (2013) [263] investigated Twitter follower markets – a multi-million dollar industry [224] – whereby users of the social network can purchase additional followers for a fee. The study focuses on pyramid markets, whereby free users of the market allow their Twitter accounts to be controlled by the pyramid merchant – these free accounts are referred to in the study as "victims". Paying customers of the pyramid markets then gain followers when the merchant instructs the victim (free) accounts to follow the paying customers. The study discovered the market rate for followers to be between $40 and $26 for 10,000 followers.

The study analysed the characteristics of both victim accounts and market customers. They detect that customers initially see a dramatic increase in followers to their Twitter accounts, followed by a steady decrease as victim accounts unfollow the customers due to the content not being interesting – about 70% of customers lose more than 100 followers. By analysing the characteristics of follower dynamics, the study also developed

a system to detect customers of follower markets.

For their methodology and data gathering, the study obtained a ground-truth consisting of 69,222 victims and 2,909 customers of follower markets. To locate popular follower markets, the study used search engines and a Support Vector Machine (SVM) classifier to distinguish genuine real Twitter follower market websites from benign ones. To collect Twitter accounts of victims, the authors purchased followers from 5 top markets (BigFollow: 16,185, Bigfolo: 1,404, JustFollowers: 20,897, NewFollow: 10,307, and InterTwitter: 20,429).

To detect and collect customers, the authors registered 180 new Twitter accounts, set them up as free (victim) accounts in follower markets, then developed an algorithm – based on analysing the followers of these newly-created Twitter accounts – to identify the market customers (BigFollow: 2,781, Bigfolo: 37, and JustFollowers: 91). The study also collected 4 million legitimate Twitter users consisting of 2 million randomly-sampled users and 2 million randomly-sampled legitimate users comparable with market customers (i.e. with >100 followers).

To establish the market size, the authors searched for tweets advertising the follower markets in their dataset, using Twitter's 10% Steam API feed sample from 16th January to 7th May 2013. They collected over 3.3 billion tweets and discovered that 5,473,041 tweets were advertising follower markets along with 740,506 victims. The study notes that, to hide their involvement in purchasing followers, premium customers gain followers slowly (e.g. 3,000 over 1 month).

The Stringhini *et al.* study illustrates how anyone on Twitter can increase their number of followers for a fee. In the Ghosh *et al.* study, we saw how spammers can exploit the reciprocality of legitimate "social capitalist" Twitter users to expand their followers with relatively low effort. With the Stringhini *et al.* study, we see a different – paid – method for spammers to achieve a similar outcome. These studies show how easy it could be for phishing and malware URLs to reach large audiences on Twitter – and how these dangerous URLs can appear to come from influential Twitter users with large followers. Victims of these attacks on Twitter may believe that the attacks originated from trustworthy Twitter users.

SUBSECTION SUMMARY

The existing studies we have explored in this subsection show how URLs can propagate on Twitter, such as via "word-of-mouth". We also see how Twitter users with nefarious motives can exploit the social network to increase their reputation and, therefore, influence. We see here how a malicious individual (or criminal group) can carry out a Sybil attack (defined in Section 2.5: *Sybil Attack*) by controlling numerous fake Twitter accounts in order to

manipulate legitimate Twitter users. A Sybil attack could be carried out either by leveraging the criminal community and link farming, as explored by Yang *et al.* and Ghosh *et al.*; by paying for followers, as explored by Stringhini *et al.*; or a combination of both paid and free techniques.

3.8.4 Phishing & Malware

This section begins with an exploration of existing phishing and malware measurement studies whose scopes examine the web in general. This section then narrows down into Twitter specific phishing and malware measurement studies.

Phishing & Malware Measurement Studies On The Web In General

Moshchuk *et al.* (2006)

Moshchuk *et al.* (2006) [197] investigated the prevalence of spyware on the web by using a web crawler to carry out a large-scale, longitudinal study. The study carried out 2 key measurement periods: one in May 2005 (over 18 million URLs crawled) and one in October 2005 (nearly 22 million URLs crawled). In both crawls, the authors found executable files in approximately 19% of the crawled websites, and spyware-infected executables in about 4% of the crawled websites. In May 2005, after crawling 18 million URLs, the authors discovered spyware in 13.4% of 21,200 executables they identified. They also discovered that 5.9% of websites they processed contained scripted "drive-by-download" attacks.

The authors observed a noticeable decline in the presence of "drive-by-download" attacks in October compared to May 2005 (2.4% compared to 1.6%, respectively). The key methodology for their study involved using the Heritrix public domain Web crawler [126] to crawl over 2,500 websites for executable downloads, installing these executables in a virtual machine, then running the Lavasoft AdAware anti-spyware tool [154] as their ground truth to detect spyware. The study crawled websites from 8 categories: adult entertainment, celebrity-oriented, games-oriented, kids, music, online news, pirate/warez, and screensaver or "wallpaper". They also included CNet's *download.com* shareware site.

Boshmaf *et al.* (2013)

Boshmaf *et al.* (2013) [46] investigated how vulnerable online social networks (OSNs) are to a large-scale infiltration campaign run by an army of "socialbots": bots that control OSN accounts and mimic actions of real

users. “Socialbots”, as opposed to other bots, are designed to pass as real humans. The bots achieve this by either mimicking the actions of real users or by simulating a user by using artificial intelligence. As a result, this “socialbot” is able to gain influence on its target OSN – and can therefore manipulate users on the platform (e.g. to spread misinformation, alter political opinion, expose malicious URLs, etc).

The study used Facebook as its data source and ran their “socialbot” for 8 weeks on the platform. The study’s key results show that, by exploiting known social behaviours of users, Facebook can be infiltrated with a success rate of up to 80%. Depending on a user’s privacy settings, even more private user data can be exposed. The study notes that running such a large-scale infiltration campaign may be profitable but is not effective in terms of sustainability or being an independent business.

Gonzalez *et al.* (2011)

Gonzalez *et al.* (2011) [98] investigated the prevalence of phishing by form – a technique whereby phishers use forms on well-know websites (e.g. Google Forms) to elicit information from their victims. This type of phishing attack is successful for criminals because the forms are hosted on trusted websites (such as *Google.com*) therefore convincing victims of the attack that it is safe to submit their private information to the phishing form.

The study leveraged 2 key data sources: PhishingCorpus [203] which consisted of 2,240 emails, containing 82 phishing emails, collected between 7 August 2007 and 26 July 2011; and the PhishTank archive, containing approximately 151,500 confirmed phishing URLs collected between July 2010 and August 2011.

The authors observed that, in their dataset, the first occurrence of a phishing email that employed the phishing by form technique occurred on December 14 2009. This email message told the receiver that they had exceeded their inbox quota and must therefore “re-validate by authenticating or risk loosing email access”; the email included a URL to a form hosted on *eformit.com*. The study observed the first occurrence of a phishing by form attack that adopted a *Google Sheets* form in March 2011.

In terms of how popular the phishing by form technique is: 0.13% of the PhishTank URLs and 3.7% of the PhishingCorpus messages leveraged this technique. The average lifetime of a phishing by form site is 15 days 22 hours. In comparison, the average lifetime of regular phishing sites in the PhishTank dataset ranged from less than 1 hour to over 500 days, with an average of 13 days 11 hours.

Motoyama *et al.* (2011)

Motoyama *et al.* (2011) [198] investigated the prevalence of web abuse jobs posted to online labour markets. The study analysed 7 years' of data collected from the freelance job sharing platform *Freelancer.com*. The study identified web abuse jobs, in the dataset, and categorised them under account creation, social network link generation, and search engine optimisation support, along with characterising the evolution of price and demand.

For data collection, the study used Freelancer's API to crawl the entire project IDs' space – which ranged from 1 to 1,015,634 – between December 16 2006 and April 6 2011. This dataset contained 815,709 active users and 842,199 jobs that were posted between February 5 2004 and April 6 2011. The study noted that approximately 46% of posted jobs report a worker selected – although this number represents a lower bound on the number of job transactions since some buyers and sellers will conform jobs through private messaging, away from the Freelancer website.

To categorise jobs: the study first manually identified the categories of jobs on the site, then used a combination of keyword matching and supervised learning to categorise all jobs in their dataset – focusing on "dirty" jobs that relate to account creation, "CAPTCHA" solving, verified accounts, search engine optimisation (e.g. link building) tasks, etc.

Key results of this study identify various different job types among a number of classes that fall under "dirty" jobs – along with the price for each job. This illustrates that malicious users can easily purchase labour to assist in their web abuse activities – again, increasing the number of techniques available to reach more victims.

Thomas *et al.* (2017)

Thomas *et al.* (2017) [270] carried out a longitudinal measurement study to investigate the underground ecosystem fuelling credential theft. The study collected data between March 2016 and March 2017 and identified 788,000 potential victims of off-the-shelf keyloggers; 12.4 million potential victims of phishing kits; and 1.9 billion usernames and password exposed via data breaches and traded on black market forums. Using this data, the study explored how easy it is for criminals to gain access to a victim's email account – due to exposed passwords matching the email account password (*i.e.* when people use the same password for multiple websites). In their dataset, 7-25% of exposed passwords matched the victim's Google account.

The study also examined the miscreants' toolbox to discover easy to acquire key-loggers, phishing kits, etc., that help to drive this lucrative underground ecosystem. Data collection for the study involved regularly crawling *paste* sites and black hat forums for data breach credential leaks,

then parsing the crawled data for verification. With ethics in cyber security research being a hotly debated topic [75], the study does state that they do not purchase or trade credential leaks, any discovered plain text passwords were hashed in their dataset, and that all exposed Google accounts were re-secured via a forced password reset – as per Google policies [264]. The study also attempts to address the vulnerability of exposed user credentials by suggesting a number of mitigation techniques such as blocking login attempts where the user's history or device(s) does not match.

Wang *et al.* (2012)

Wang *et al.* (2012) [291] measured malicious crowd-sourcing systems to investigate *crowdturfing* – a portmanteau of "crowd-sourcing" and "astroturfing", whereby customers initiate "campaigns" and users typically receive financial compensation for performing "tasks" that go against accepted user policies. For its data source, the study used 2 popular crowdturfing platforms in China: Zhubajie (ZBJ, zhubajie.com, active since Nov 2006) and Sandaha (SDH, sandaha.com, active since March 2010). The authors crawled these 2 sites in September 2011.

Key results show that $4 million dollars was spent on crowdturfing campaigns on these sites during the study's measurement period. The study observed a total of 76,000 crowdturfing campaigns posted to ZBJ (92% of all campaigned posted during the timeframe) and 3,000 posted to SDH (88% of all campaigns posted to the site). For ZBJ, these campaigns consisted of 169,000 workers, 17.4 million total tasks, 6.3 million (36%) submissions, and 3.5 million (56%) were accepted. This generated $3 million: $2.4 million for workers and $595,000 for ZBJ (20% commission). For SDH: the 3,000 posted campaigns consisted of 11,000 workers, 1.1 million tasks, 1.4 million submissions (130%), and 751,000 (55%) were accepted. This generated $161,000: $129,000 for workers and $32,000 (20% commission) for SDH.

The most popular types of campaigns posted to these sites include: account registration, forum posting, blog posting, microblog (e.g. Twitter), Q&A (posting and answering questions on Q&A sites like Quora.com), etc. The study also posted its own benign crowdturfing campaigns to these websites (ZJB and SDH), including a link for "more info" within each campaign's description, which is used to measure the total number of clicks each campaign received – along with allowing workers to carry out "tasks" that the study can measure. This experimental setup allowed the authors of the study to evaluate the effectiveness of the crowdturfing campaigns that they advertised on the sites.

Key results show career crowdturfers that control thousands of accounts on OSNs – which are carefully managed by hand. These workers can gen-

erate significant information cascades which are not detected by security systems designed to catch automated spam – therefore allowing the spread of potentially dangerous information or malicious URLs into OSNs. The authors note that the spam generated from these campaigns is highly effective, driving hundreds of clicks from normal users.

Schryen *et al.* (2007) & Prince *et al.* (2005)

Schryen *et al.* (2007) [247] investigated the impact that placing email addresses on the Internet has on the receipt of spam. Key results show that web placement attracts 70% of all honeypot spam emails, followed by newsgroup placements (28.6%) and newsletter subscriptions (1.4%). More than 43% of email addresses on the web have been abused, compared to about 27% on newsgroups and 4% for newsletter subscription. Only about 1.5% of spam emails' topics were related to the location where the email addresses were sourced from – i.e. spammers send emails in a "context insensitive" manner. Their honeypot received 57,273 emails, of which, 26,882 (47%) were spam.

In a similar study, Prince *et al.* (2005) [231] investigated spammers' email harvesting techniques by analysing data gathered from the Project Honey Pot [283] – a distributed honey pot network of over 5,000 members to track email harvesters and spammers. The study observed that approximately 6.5% of traffic visiting their honey pots subsequently turned out to be spam harvesters. The average time from a spam trap email address being harvested to receiving its first email is just over 11 days – with the fastest turnaround being less than 1 second and the longest being over 223 days.

These studies highlight the risks associated with leaving contact details publicly visible on the internet and how mass crawling is a common technique used by spammers.

Irani *et al.* (2008)

Irani *et al.* (2008) [132] investigated the evolution of phishing emails – by exploring over 300,000 messages collected between August 2006 and December 2007. The study observed 2 key techniques used by phishers for sending messages: flash attacks, where a large volume of messages are sent over a short period of time; and non-flash attacks, where the same phishing message is sent over a relatively longer period of time. They also classiffied phishing features into 2 groups: transitory, whereby features are present in a few attacks and have a relatively short life spam; and pervasive, whereby features are present in most of the attacks and have a long life span.

The authors conclude that transitory features are generally a strong indicator of phishing, and pervasive are weak selectors of phishing. The

paper does not specifically cite the source of the dataset, other than stating it was provided by "a large anti-phishing organisation". The corpus contained over 1.8 million spam and phishing messages spanning 15 months from August 2006 to December 2007 – with the study's authors using a number of conservative techniques to filter the dataset down to just phishing email.

McGrath *et al.* (2008)

McGrath *et al.* (2008) [180] investigated various aspects of phisher modi operandi, examining the anatomy of phishing URLs and domains, registration of phishing domains and time to activate, along with the servers used to host phishing content. The core aim of their study is to provide heuristics for filtering systems. In doing this, the paper provides an interesting measurement study with information that is relevant to this thesis' topic.

The study's data sources are PhishTank (consisting of 44,320 URLs, 17,105 domains, 144 TLDs) and MarkMonitor [174] (consisting of 25,304 URLs, 7,394 unique domains, 116 TLDs) – both were collected over a duration of 71 days from November 30 2007. The study also collected data from DMOZ (the Open Directory Project [205]), used to observe differences between phishing URLs and good URLs, which contains over 9 million unique URLs and 2.7 million unique domain names. 14% of phishing URLs contained in PhishTank were also in MarkMonitor. Similarly, 12% of domains in PhishTank were also in MarkMonitor.

Han *et al.* (2016)

Han *et al.* (2016) [117] examined phishing kits to analyse phishing attacks, their mechanisms, the behaviour of criminals, their victims, and the security community involved in the process. The study collected data over a period of 5 months by running phishing kits in sandboxes – which protected the identify of victims. This allowed the authors to conduct a lifecycle analysis, from the time in which the attacker first installs and tests the phishing pages on a compromised host, until the last interaction with victims and security researchers.

For data collection, the study utilised infrastructure from their previous research [52], and leveraged the popularity of Amazon EC2 as a target for attackers looking for machines to compromise [116], to install their honeypot on. The honeypot contained 18 vulnerable PHP pages that were known to be exploited by hackers– and allowed attackers to upload their phishing kits to the researchers' machine. The paper contains details of their infrastructure and implementation, along with how they designed the system to convince attackers that their phishing kits were working, whilst protecting the victims.

Key results from the study: the authors collected 643 unique phishing kits between September 2015 and January 2016. Of this dataset, 474 kits (74%) were correctly installed by 471 distinct attackers. The remaining kits were likely automatically uploaded by bots but never unpacked or configured. The successfully installed phishing kits targeted 36 unique organisation: mostly banks, social networks, and e-commerce portals. The 5 most frequently targeted organisations were: Paypal (375), Apple (26), Google (10), Facebook (9) and the French online tax payment system (6).

The study includes detailed analysis of how phishing kits were setup and operated by attackers, the behaviour of victims of the kits, and the lifecycle of the kits. Key results show an average lifetime of 8 days for phishing kits uploaded to their research infrastructure. Most attackers discovered the honeypot via Google (28%) and Yahoo (0.8%). Over 40% of the attacks came with Facebook referrers – suggesting that attackers leverage the social network to share data. Once installed, 70% of attackers visited the phishing pages and 58% submitted fake credentials to verify the installation succeeded. Their honeypot received 2,468 victims, connected to 127 distinct phishing kits. 215 users (9%) submitted their credentials to the phishing page.

An important finding in the study was the observation of a spike in phishing kit visitors just after a phishing page appears in a public phishing blacklist. This crowd phenomenon may lead other studies, that measure phishing web page traffic, to unknowingly overestimate the real number of victims. The study also measured the detection times for the blacklists GSB and PhishTank to detect the phishing kits installed on their research infrastructure. Whilst 98% of phishing pages were correctly identified by the 2 blacklists, the study observed an average delay time of 12 days. The study splits this figure into 2 separate categories: phishing kits which received victims were blacklisted in an average of 20 days after installation, while phishing kits with no victims were blacklisted in an average of 10 days after installation.

The study notes that the delay in detection time varies depending on the evasion techniques used by the attackers. 62% of phishing kits were blacklisted only after 75% of victims had already connected. However, in another category, 27% phishing kits were blacklisted before 25% of victims had connected to the URLs. The study also noticed that GSB initially blacklisted individual phishing kits. But after many URLs were reported for the same domain name, GSB started to blacklist entire directories in those domains – a blacklisting technique that may proactively blacklist other kits that were installed within these same directories. The study states that their "...results show that GSB and PhishTank are not fast enough to blacklist new phishing kits, which leaves victims on their own to identify and protect against phishing attacks".

Cui *et al.* (2017)

Cui *et al.* (2017) [64] monitored 19,066 phishing attacks over a period of 10 months and observed that the attacks were replicas or variations of other attacks in their dataset. The study suggests that small groups of attackers could be behind a large part of current attacks – therefore taking down that group could have a significant impact on the observed phishing attacks. The study proposed a number of techniques to improve phishing website detection and prevention.

The study utilised PhishTank as its dataset, collecting 21,303 "verified" URLs between January 1 and October 5 2016. The study then used a crawler to fetch the DOM of each URL along with the server's IP address. 2,237 of the total URLs they collected from PhishTank were unreachable: 350 returned 400-level HTTPS errors [128], 59 were reported to have been taken down by the hosting provider, and 1,828 were empty or failed to load. Therefore the study were left with 19,066 URLs that they could process.

Moore *et al.*

Moore *et al.* have produced numerous studies that explored phishing blacklists, their overlap, and effectiveness of takedown efforts by defenders [188, 189, 190, 191, 194, 192, 193]: In 2007 they analysed the current state of phishing attacks and defence [188], they also examined the impact of website take-down on phishing [189]; in 2008 they followed-up on their 2007 studies with an investigation into the consequence of non-cooperation in the fight against phishing [190]; In 2009 they explored the temporal correlations between spam and phishing websites [194], they also examined website compromise and re-compromise [192]; and in 2010 they explored the difficulties involved in measuring phishing on the internet [193].

Subsection Summary

As we have seen in this subsection, there is a wealth of existing literature that has measured phishing and malware on the web in general. From exploring spyware and phishing by form, to analysing socialbots and crowdturfing.

The longitudinal spyware measurement study, executed by Moshchuk *et al.*, involved 2 large-scale web crawl measurements which were carried out 5 months apart. The study's ground truth relies on the Lavasoft AdAware anti-spyware tool to detect spyware, once discovered executables were installed on the author's virtual machine. In its write-up, the study details which versions of the AdAware spyware detection database was used along with a description of their hardware setup. This greatly improves the verifiability

of the study and allows for other researchers to repeat and replicate the study to compare results.

Many of the existing studies we have explored in this section relate to other measurement topics we have explored in previous sections, such as information credibility and networks. We see a number of studies that analyse the labour market behind malicious activities on the web. The Motoyama *et al.* and Wang *et al.* studies provide evidence to illustrate how easy it can be to purchase labour for driving malicious campaigns (such as a Sybil attack, posting phishing URLs, etc). In the Thomas *et al.* study we see empirical evidence of how easy it can be for criminals to gain unauthorised access to individuals online accounts.

The methodologies involved in studies that examine underground ecosystems first require access to key data. This is often provided as an API to a job market – such as the Freelancer API in the Motoyama *et al.* study – or crawling the web, as seen in the Thomas *et al.* and Moshchuk *et al.* studies. Correctly labelling datasets to determine underground jobs often leverages machine learning techniques, as seen in the Motoyama *et al.* study.

Having explored measurement studies that examine phishing and malware on the web in general, we shall now explore existing studies that have focussed their measurements onto the social networking platform: Twitter.

Phishing & Malware Measurement Studies On Twitter

Grier *et al.* (2010)

Grier *et al.* (2010) [110] characterised spam on Twitter: finding that 2 million (8%) of 25 million URLs posted to the social network point to either phishing, malware, or scam websites (using the umbrella term "spam" for these 3 categories). The study used blacklists as their ground truth. The study identified 2 types of spamming accounts: those created primarily for sending spam, and legitimate accounts that have been compromised. The study analysed clickthrough rates for spam URLs to determine that 0.13% of Twitter users visited these blacklisted spam websites. The study also measured the performance of blacklists, as a filter for protecting Twitter users against spam attacks, to discover that blacklists are too slow – allowing more than 90% of Twitter users to view a potentially harmful website before it becomes blacklisted. The 3 blacklists used in this study are: GSB [102], URIBL [285], and Joewein [296].

Whilst many scams present in email carry over to Twitter, the study identified Twitter-specific features that spammers use to gain exposure:

- Call outs: whereby a specific Twitter user is addressed (e.g. *win an iPhone @victim*)
- Retweets: approximately 1.8-11.4% of spam tweets are retweets of blacklisted URLs, with sources including purchased retweets and spam accounts retweeting other spam accounts
- Tweet hijacking: whereby spammers prepend spam URLs in retweets – a technique that is not against Twitter's rules
- Trend setting: seen when a large enough volume of tweets contain a specific matching hashtag(s), at which point the specific hashtag(s) becomes trending – something spammers can exploit
- Trend hijacking: whereby spammer append currently trending hashtags to their spam tweets

The study uses Twitter's Stream API as a data source to collect over 200 million tweets over a 1 month duration, between January and February 2010. They use two data taps: a random sample of all tweets to generate statistics about the amount of URLs in tweets and general Twitter trends, and a URL specific stream to carry out their core measurements. Of the detected 2 million blacklisted URLs in the study, 5% were malware and phishing while the remaining 95% were scams. 245,000 tweeted URLs in their dataset were shortened with *bit.ly*, therefore the study was able to leverage Bitly's API to analyse click traffic; discovering a combined total of over 1.6 million clicks. Of these Bitly URLs that received clicks, 50% receive fewer than 10 clicks, whilst the upper 10% of URLs receive 85% of the 1.6 million clicks.

The techniques identified by Grier *et al.*, that spammers use to gain exposure, are relatively easy for spammers to deploy. This adds to the techniques we have already seen in previous studies, illustrating that there is a healthy amount of such techniques available for phishing and malware creators to use as part of their arsenal to gain victims via Twitter.

Stringhini *et al.* (2010)

Stringhini *et al.* (2010) [262] investigated to what extent spam has entered social networks; specifically focusing on spammers' modus operandi. The study consists of a measurement study followed by the development of a machine learning classifier to automatically detect spammers in social networks.

Their measurement study involved logging profile traffic (e.g friend requests, messages, invitations, etc) by deploying 300 "honey-profiles" to

each of the social networks: Facebook, MySpace, and Twitter (900 profiles in total). These honey profiles acted passively and did not send any friend requests – but did accept all friendship request they received. The author's system collected traffic logs of their profiles for 12 months on Facebook (6 June 2009 to 6 June 2010) and for 11 months on MySpace and Twitter (24 June 2009 to 6 June 2010). Profiles were checked slowly (approximately 1 account visited every 2 minutes) to avoid being detected as a bot by the social networking site – which would have resulted in the profile's account being deleted.

Overall, their profiles received 4,250 friend requests and 85,569 messages: the Facebook profile received 3,831 friend request (of which 173 were from spammers) and 72,431 messages (of which 3,882 were spammers); MySpace received 22 friend requests (of which 8 were from spammers) and 25 messages (0 spam); and Twitter received 397 friend requests (of which 361 were from spammers) and 13,113 messages (of which 11,338 were spammers). In order to label profiles as spam, the authors first manually checked all profiles (that contacted their honey profiles) to detect common traits. Spammers were typically either braggers (bots that post messages to their own feed) or posters (bots that send a direct message to other users).

The authors then trained a spam identification classifier which used features such as FF ratio (number of friend requests versus actual friends), URL ratio (messages received that contain URLs versus total number of messages), message similarity (between messages received from different users), etc. Although this study analysed 3 social network, and focused on spam as a general term, it does illustrate the volume of spam measured on the social networks along with highlighting the spammers' modus operandi.

Lee *et al.* (2011)

Whilst the previous study, Stringhini *et al.*, examined 3 social networks (which included Twitter), Lee *et al.* (2011) [157] focused their investigation on the modus operandi of "content polluters" on *just* the Twitter social network. The paper begins with a measurement study, followed by an evaluation of features to create a machine learning classifier to automatically detect content polluters.

In their measurement study, the authors deployed 60 honeypots to the Twitter social network, during a 7 month measurement period, harvesting 36,000 candidate content polluters. Their measurements ran from 30 December 2009 to 2 August 2010, gaining 36,043 followers, of which 7,773 (24%) Twitter users followed more than one of the authors' honeypot accounts. After removing users that followed more than 1 account, the study was left with 23,869 users.

The approach, used by the authors to attract content polluters on Twitter, differs from that in [262]. Lee *et al.* actively send out spam-looking tweets – in order to discourage regular Twitter users from following their honeypot accounts – whereas the honeypot accounts used in [262] are passive. Lee *et al.* deduce that any Twitter user that follows their honeypot accounts must be a bot – since no real human Twitter user would want to follow them. To verify this, the authors investigated their 23,869 newly acquired followers and discovered that 5,562 (23%) had been suspended by Twitter – with an average time of 18 days for the accounts to be banned after following the honeypots.

The authors then used the Expectation-Maximization (EM)[69] cluster analysis algorithm to find groups of users with similar behaviours. This allowed the authors to explore common themes amongst the content polluters to produce a set of features (follower ratio, tweet frequency, etc.). The results of the measurement aspect of this study illustrate the number of active content polluters – 36,000 – that were active during a 7-month timeframe.

Thomas *et al.* (2013)

Thomas *et al.* (2013) [271] investigated the market for fraudulent Twitter accounts to monitor prices, availability, and fraud – perpetrated by 27 merchants over the course of a 10-month period. The study also analysed 10-20% of the fraudulent Twitter accounts, produced by said merchants, to discover that they had generated $127,000 to $459,000 in revenue. The study also produced a machine learning classifier to automatically detect fraudulent accounts sold via the marketplace.

As part of their measurement, the study, with permission from Twitter, purchased 121,027 fraudulent Twitter accounts on a bi-weekly basis from June 2012 to April 2013. Prices ranged from $0.01 to $0.20 per account – with a median cost of $0.04. Similarly to [291, 198], the Thomas *et al.* study leverages the "underground market" to discover merchants' web abuse services. The authors' discovered these underground markets on blackhat forums and freelance labour markets (including *Freelancer.com* – which we saw in [198]).

In terms of ethical considerations related to buying fraudulent services from underground markets, the authors consulted with Twitter to produce a set of guidelines for interacting with merchants; which is featured in the appendix of their paper. The Thomas *et al.* study provides further information as to the prevalence of underground markets – and demonstrates how readily available these services are to purchase.

Oliver *et al.* (2014)

Oliver *et al.* (2014) [214] examined 573.5 million URL-containing tweets, over a 2-week period, from 25 September to 9 October 2013 to understand how Twitter is abused. They discovered 33.3 million (5.8%) malicious tweets. In their study, malware tweets receive an average of 0.03065 clicks per tweet whilst "traditional phishing" tweets received 0.00959 clicks per tweet. The study leverages 2 methods to identify malicious tweets: the blacklist Trend Micro Web Reputation Technology and a clustering algorithm which is described in the paper. The paper also details specific campaigns that they observed during their measurements.

Liu *et al.* (2014)

Liu *et al.* (2014) [166] investigated the evolution of Twitter users and behaviour in a 7-year longitudinal study; analysing over 37 billion tweets between 2006 and 2013. The study utilises 2 key datasets: an almost-complete collection of all tweets between 21 March 2006 and 14 August 2009, collected by previous work [54]; and data collected via Twitter's "gardenhose" (a random sample of approximately 10% of all public tweets) between 15 August 2009 and 31 December 2013.

Key results of the study observe an increase in the number of suspended Twitter users, with over 6% of the entire Twitter population suspended by late 2013 – which is in line with [271]. When examining the average ratio of friends-to-followers, the ratio increases from 1.50 to a high of 1.77 in January 2012 before returning to its previous value. This increase corresponds well with the rise of Twitter follower spam in 2010 and 2011 (Stringhini *et al.*(2012) [261]).

Subsection Summary

In this subsection we have explored existing studies that measure phishing and malware on Twitter. We see studies leveraging methodologies that involve "infiltrating" the social network in order to measure malicious users. This is seen in both the Stringhini *et al.* and Lee *et al.* study whereby social honeypot accounts are deployed to the social network to gain spammer followers. Whilst the Stringhini *et al.* study takes a passive measurement process towards this, the Lee *et al.* proactively sent out spam-looking Tweets in order to discourage regular followers. This increases the robustness of their methodology with the theory that only bots would follower their honeypot accounts – which they also verified by checking which accounts had been suspended by Twitter. Another interesting methodology is leveraged by Thomas *et al.*, whereby fraudulent Twitter accounts were purchased from the underground marketplace in order to study them. This empirical

approach to collecting data for their study provides a unique insight into the underground economy of purchasing fraudulent accounts.

Establishing ground truth is crucial in many of these existing measurement studies. We see many studies relying on blacklists for their ground truth, such as GSB, URIBL, and Joewin in the Grier *et al.* study; and Trend Micro, along with a clustering algorithm, in the Oliver *et al.* study. We also see methodologies which do not require blacklists to determine ground truth, as seen in the Liu *et al.* study, which examined accounts which have been suspended by Twitter. The main disadvantage of their methodology is that they cannot specifically categorise suspended accounts to determine why an account was suspended by Twitter. They could potentially cross-reference these suspended accounts with one or more blacklists to elaborate on their findings.

3.9 Ethics

Building on from the introduction to Ethics in Section 1.7, this section explores how ethics has been approached in existing literature and examines key ethical considerations surrounding cyber security research. Drawing on existing studies and literature reviews, this section aims to discuss the important ethical considerations in key areas such as using publicly available data – some of which is generated by humans, dealing with compromised accounts, and how to research phishing vicariously; without actually becoming a phisher or breaking the law. We will also explore the ethical considerations of sharing measurement data, since measurement datasets can be of great benefit to the research community – but must be shared responsibly.

Thomas *et al.* (2017)

Thomas *et al.* (2017) [266] explored "ethical issues in research using datasets of illicit origin"; examining the ethical principles from more than 20 peer reviewed papers that deal with illicitly obtained data sets. The study draws on case studies that acquire their data via: malware and exploitation (Carna scan, 2012; AT&T iPad users database brute force, 2012; malware source code – e.g. Zeus 2011, Mirai botnet 2016), password dumps, leaked databases (Booter database leak; Patreon crowd-funding, 2015; underground forums), financial data leaks (Panama/Mossack Fonseca papers leak, 2015), and classified materials (Manning's WikiLeaks dump of 700,000 documents, 2010; Snowden's NSA and GCHQ data leak, 2013).

Key ethical topics raised by Thomas *et al.* include: informed consent, human rights, releasing and using shared data, hacking and intervening, analysis techniques, ethical review, and REBs. The study addresses concerns around anonymising datasets: "Both Allman & Paxson, and Partridge warn against relying on the anonymisation of data since deanonymisation techniques are often surprisingly powerful. Robust anonymisation of data is difficult, particularly when it has high dimensionality, as the anonymisation is likely to lead to an unacceptable level of data loss" [5].

Thomas *et al.* state the following set of ethical issues that require consideration when conducting research with data of illicit origin: identification of stakeholders, informed consent, identify harms, safeguards, justice, and public interest. Thomas *et al.* state the following list of legal issues surrounding research with data collected illegally – although they state that the laws can be complex and that researchers should seek their own legal advice: computer misuse, copyright, data privacy, GDPR, terrorism (e.g. failing to report terrorist activities), indecent images (e.g. if scraping certain data dumps), national security, and contracts.

Overall, Thomas *et al.* conclude that, of the papers they reviewed, ethical issues are tackled "inconsistently, and sometimes not at all" – and that safeguards and legitimate ethical justifications are often lacking. Many of the papers reviewed in the study did not clearly address the positive benefits of their research or state Research Ethics Board (REB) approval. Thomas *et al.* note that most research studies that do not involve humans do not require REB approval – however, the authors suggest that any research that involves human data, even if that data is acquired non-directly from humans, should still seek approval from REB.

Although Thomas *et al.* focus on data that has been obtained via "illicit origin", whereas data acquired for this thesis does not come from such sources, the ethical and legal issues raised in their study are still relevant to any study carried out in the cyber security area. A similar paper, by Egelman *et al.* (2012) [75] discusses the ethics of performing research using public data of illicit origin.

A number of other studies carried out by Thomas *et al.*, such as in 2013 [271] and 2017 [270] also encountered specific ethical issues. In their 2013 study, the authors worked closely with Twitter to produce a set of guidelines to purchase fraudulent Twitter accounts. Interacting with these merchants in an ethical manner allowed the authors to study the underground marketplace and measure the prevalence of Twitter fraud on the social network. Additionally, in their 2017 study, the authors complied with Google's policy of resetting breached accounts. This allowed the study to analyse compromised accounts within their dataset. As a result, the study was able to measure the prevalence of compromised accounts "in the wild" and to reverse engineer how such accounts became victims in the first

place. This also enabled the study to produce a set of guidelines to help increase account security.

Han *et al.*

As we have seen, it is often necessary to infiltrate an underground economy, or network, in order to effectively measure it. Since this is sometimes the only way to acquire empirical date for analysis. This might even involve carrying out the activities of – and therefore behaving exactly like – a criminal, as seen in the Han *et al.* study [117]. This requires careful planning and thorough ethical policies.

In their study, the authors deploy their phishing kits in sandboxed environments to ensure no data is leaked. They also disable the phishing websites from actually submitting any data in the first place – which alleviates risk from data leaks. Although such risk cannot be entirely eliminated from the study, since a "victim" to the researchers' phishing page will still enter their credentials into an online form. Therefore it is possible that this data could still be leaked from the victim's computer – although, since this data is not entering the researchers' servers, the study would be less responsible in the event of such a data leak.

Alllman *et al.* (2007)

Alllman *et al.* (2007) [10] discuss various issues and ethical considerations relating to shared measurement data. They propose a "framework" for how researchers and users can treat particular datasets. Their paper focuses on data acquired through network measurements, however, many of the points raised in their paper also apply to other types of measurement data.

The paper discusses how difficult it can be for researchers to acquire measurement data, due to requiring the right opportunity in terms of administrative and legal permissions, operation support and resources, along with time consuming debugging and monitoring the measurement process for faults, etc. With these challenges in mind, the authors discuss how important it can be for the research community for researchers to share their measurement datasets. However, the authors also express their concerns about how datasets can sometimes be shared, particularly in terms of attitudes and assumptions from some researchers about how to provide and share measurement data. Therefore the authors discuss some high-level issues that should be considered when sharing such data.

In terms of data release considerations: the authors advise that data sharers understand the threat model and apply a suitable anonymisation policy – however, despite anonymisation techniques, data sharers are likely to release more information than they realise. Data sharers should also explicitly define an Acceptable Use policy for the data to ensure such data

is used appropriately – and that researchers whom breach regulations run the risk of censure when attempting to publish work that utilises the data. Data sharers should also explicitly define what metadata has been collected (and anything that is not shared) to help researchers answer any questions about the data. Finally, data sharers should explicitly state any notification requirements if their data is used in any publications and what acknowledgements should be used.

The authors also discuss purpose-provided data, in situations where data is released publicly, or when sharing data amongst colleagues or students. In these situations the data sharer may be more lax on anonymising the data because they trust the researcher(s). The authors offer guidelines which include: limiting re-distribution of data and keeping the data secure, when sharing non-pubic data: state whom has access to the data, and finally data should only be accessible to researchers for the the duration of a specified project – with permission sought for additional research projects. The authors discuss various issues relating to de-anonymising data along with numerous points to consider. Finally, the authors discuss interactions between data sharers and researchers that use the data. The authors advise researchers to ask about any "blind spots" in shared data, that may arise as a result of anonymisation techniques, and not to make assumptions about the data.

Williams *et al.*

Williams *et al.* (2017) [293] report on an ethics consultation to identify best practice procedures for the publication of Twitter data in research findings – predominantly focusing on the UK context. The study explores ethical considerations around informed consent, anonymisation, and the minimisation of harm; how conflicts can arise between commercial, regulatory, and academic practices; and how sometimes good ethical practice might compromise academic integrity.

The study acknowledges that it was not able to produce a firm consensus on best practices of publishing tweets and the handling of Twitter data. A key area of disagreement arose over whether or not publicly available data is affected by standard ethical obligations relating to informed consent, anonymisation, and the minimisation of harm. Some argue that, in every case, if consent cannot be obtained, then research data cannot be used. However, others argue that the situation should be approached on a case-by-case basis. The study highlights that, whilst a full consensus on the ethical use of Twitter data is unlikely, there is an urgent need for discussion on the topic and to find a shared pathway that researchers can follow.

The primary focus of the Williams *et al.* study is on the ethical considerations of publicising Twitter data; specifically sharing the contents of

individual tweets. Although we do not feature individual tweets in this thesis, the ethical considerations raised in the study are still important to consider.

SECTION SUMMARY
As we have seen in these existing studies, there are many ethical considerations that need to be carefully addressed when researching cyber security topics. This is especially true when such research involves the acquisition and analysis of empirical data. We shall continue to explore the ethics of cyber security research when we plan and design our experiments in Chapter 4: *Design & Implementation*.

3.10 Literature Review

In this section we analyse key methodologies of existing measurement studies. Table 3.3 shows the studies we will focus on, along with the metrics and methodologies that their measurements use to evaluate effectiveness.

The Grier *et al.* (2010) [110] study aims to characterise phishing, malware and scam URLs posted to Twitter. As part of their broad study, they analysed blacklist delays and performance. This part of the study focuses on 3 metrics: blacklist delays, number of blacklisted URLs posted to Twitter, and user clicks. One of the first limitations of this study is that it only touches on blacklist performance as part of an overall, broad analysis of spam on Twitter. The study does not provide a fine-grained nor in-depth assessment into how effective Twitter's use of blacklists are at protecting its users from phishing and malware attacks. Another limitation of the study is that details of their methodology are not provided. This limits the replicability and reproducibility of the study. This links to our first and second research questions.

For the first metric, blacklist delays, the study's methodology measures the delay from when a blacklisted URL is tweeted to when it appears in a blacklist. The methodology treats multiple tweets of the same URLs as being unique, independent events. One of the main problems with this methodology is that a URL may be tweeted at a certain point in time, then tweeted again on multiple occasions at much later dates, closer to the point at which that URL becomes blacklisted. This then skews the results because the average delay time for that URL to become blacklisted, when calculated using all tweet times containing that URL, will appear to be smaller than the time of first tweet to blacklist delay. This will tend to underestimate the exposure of users. This links to our fifth research question.

Study	Metric	Methodology
Grier *et al.* (2010) [110]	Blacklist delays	Delay = from time of tweet to time of blacklist; multiple URLs treated independently
	Number of blacklisted URLs posted to Twitter	Check tweeted URLs for membership in blacklist: GSB, Joewein, and URIBL
	Blacklistd URL user clicks	Get Bitly click data (via Bitly's API) for blacklisted, tweeted URLs in dataset that have been shorted with Bitly
Zhang *et al.* (2006) [311]	Phishing detection rates of web browser	Compare detection rates of 9 web browser extensions
AV Comparatives (2012) [23]	Phishing detection rates of web browser	Compare detection rates of 5 web browsers
Ludl *et al.* (2007) [169]	Phishing blacklist detection rates	Lookup known phishing URLs (sent via email) in blacklists GSB and Microsoft
Sheng *et al.* (2009) [251]	Phishing blacklist detection rates	Visit known phishing URLs (sent via email) in web browsers

Table 3.3: Summary of key existing studies' metrics and methodologies for evaluating effectiveness.

The study's second metric is: number of blacklisted URLs posted to Twitter. The methodology involves checking each tweeted URL in their dataset for membership in 3 blacklists: GSB, Joewein, and URIBL. Since this study aims to characterise "spam" on Twitter, they include the spam blacklists Joewein, and URIBL. Although the study's scope of blacklists is fine for their broad characterisation of spam, it does limit the preciseness of the study. More specialist blacklists – such as OP and PT – would be needed to provide suitable coverage for a fine-grained, in-dept analysis of phishing on Twitter. Another limitation of the study's methodology is that it does not address URL redirection chains. A single URL may contain multiple hops before reaching its final landing page. All of the URLs contained in this redirection chain should be checked for blacklist membership to ensure more comprehensive coverage. This links to both our first and second research questions.

The study's third metric is: blacklisted URL user clicks. The methodology

involves retrieving Bitly click data, via Bitly's API, for blacklisted and tweeted URLs in their dataset that have been shorted with a Bitly URL. There are a number of limitations to this approach of gathering URL click data. One, is that only checking Bitly URLs for click data may skew the results. Bitly was the biggest URL shortener when the measurement study was conducted; this may carry an inherent bias of trust resulting in disproportionately more people clicking on Bitly links compared to other URLs or URL shorteners. Another limitation is that click data from Bitly includes the entire history of a link. This means that clicks that were recorded before the URL was tweeted, or became blacklisted, may be included in the results – therefore potentially producing inaccurate results. This can be rectified by defining a start date for click data in Bitly's API. Additionally, the study does not state if click data from Bitly has been verified as coming from Twitter (see the aforementioned limitation of the replicability of their study) – therefore clicks may not have come from Twitter. This may also potentially affect the accuracy of the results. This can be rectified by filtering Bitly's click data by referring domain (in this case Twitter). The measurement study was carried out in 2010. A year later, Twitter launched its own URL shortening service: t.co. Since then, all URLs posted to Twitter are shortened via this service. This gives Twitter full control over and monitoring of URLs posted to, and clicks leaving, its platform. Some of the limitations of the study's methodology could be addressed by making use of Twitter's URL shortener. This would allow the metric to determine if a given URL was blocked by Twitter at time of click. These limitations link to our sixth, seventh, and eighth research questions.

Zhang *et al.* (2006) [311] used the metric of analysing phishing detection rates of web browsers as their measure of effectiveness. Their methodology involved visiting known phishing URLs from APWG and GSB in web browsers with phishing detection extensions installed. Their study included 9 such web browser extensions. One of the main limitations of this approach is that the detection rates of the web browsers themselves are not being tested. Only the browser extensions. This links to our first research question.

The limitation of the previous study is addressed by an AV Comparatives (2012) [23] study. The study considers the same metric: phishing detection rates of web browsers. It improves on the previous study by visiting known phishing URLS within the web browsers – without any additional extensions installed. One of the key limitations of this study is that the web browsers are only tested on one operating system Microsoft Windows. Whilst results *should* be consistent across operating systems, it is important to verify. Another limitation of the study is that it is not clear how many of the phishing websites had already been detected by blacklists when they were tested in the web browsers. The effectiveness of web browser's phishing

detection rates could be improved by splitting the results into URLs that were blacklisted and non-blacklisted at time of test. This links to our first research question.

Both Ludl *et al.* (2007) [169] and Sheng *et al.* (2009) [251] feature the metric of phishing blacklist detection rates as their measures of effectiveness. Both studies use a similar methodology that involves visiting phishing URLs – sourced via phishing email campaigns – in web browsers to determine blacklist detection rates. This technique works because web browsers use the blacklists to check if a visited website is "safe". One of the main limitations of the 2007 study is that the phish used to determine detection rates were not "fresh" enough. Blacklists take time to update, therefore a freshly emailed phishing URL is unlikely to have had time to appear in a blacklist. The 2009 study addressed this limitation by using phishing URLs that were more fresh. A limitation of both of these studies is that blacklist effectiveness is being assessed via the web browser's detection rate. However, there can be a delay in web browsers updating their blacklists – which may affect the accuracy of the studies' results. Additionally, web browsers may feature heuristics that can detect phishing websites without a blacklist. This may cause the study to conclude that blacklists have a higher detection rate that they actually do. Another limitation of the study is that it focuses on phishing emails as a source of URLs. Since the aim of our research is to assess the detection rates of phishing URLs on Twitter, the same methodology – using phishing email as a source – would produce inaccurate results in our studies. These limitations link to our first and second research questions.

No previous studies – that have analysed either GSB, OP, or PT – have assessed the blacklists against the framework we defined in Section 2.16.2. In particular, metrics such as size, comprehensiveness, overlap, have not previously been studied. In addition to this, no previous studies have assessed the effectiveness of Twitter's URL shortener (t.co) in their work. These gaps in existing literature link to our third, fourth, sixth, and seventh research questions.

Section Summary

Our critical analysis in this section has highlighted key limitations of existing studies, such as the methodologies used to measure metrics in order to evaluate effectiveness. The main limitations we identify in existing literature, linked to our research question(s), are summarised in Table 3.4.

Study	Limitation	RQ
[110]	Broad study of spam on Twitter that is not specific to phishing & malware nor fine-grained or in-depth.	1, 2
	Details of methodology, implementation, etc not provided. Limits reproducibility.	1, 2
	Multiple tweets of same URL treated independently; underestimates results.	5
	Does not use specialist blacklists.	1, 2
	Does not address URL redirection chain (or implementation details).	1, 2
	Only checking Bitly URLs may skew results.	8
	Bitly click data includes entire history; may produce inaccurate results.	8
	Does not state if Bitly clicks come from Twitter.	8
	Does not include Twitter's URL shortener (t.co).	6, 7
	Out of date since Twitter has evolved; user base increased from 30 million in 2010 to over 330 million users in 2017, daily tweets increased from 35 million in 2010 to over 500 million in 2017, character limit increased from 140 to 280, dynamics changed, etc.	1, 2, 5, 8
[311]	Does not test "vanilla" web browser phishing detection rates.	1
[23]	Only tests web browsers on Microsoft Windows operating system.	1
	Not clear how many tested URLs were blacklisted at time of test.	1
[169]	Tested phishing URLs are not "fresh" enough.	1, 2
[169, 251]	Blacklist effectiveness tested via web browsers; may introduce bias and inaccurate results.	1, 2
	Does not assess if web browsers feature phishing detection heuristics; may bias results.	1, 2
	Focuses on phishing emails as a source – not tweeted URLs.	1, 2
n/a	No previous studies have analysed or characterised blacklists GSB, PT, and OP for URL uptake, dropout, typical lifetimes, and overlap.	3, 4
	No previous studies have analysed the effectiveness of Twitter's URL shortener (t.co) at protecting users from phishing and malware attacks. Most studies were conducted before Twitter introduced its URL shortener (in 2011).	6, 7

Table 3.4: Key limitations and omissions of existing studies, linked to our Research Question (RQ) numbers.

Chapter Summary

In this chapter we summarised existing literature to explore various methodologies, justify our research methodology, and provide context to our research findings. We also reviewed existing studies to identify and address methodological limitations.

We explored human aspects and psychology behind phishing and malware attacks; exploring why phishing attacks work, usability studies to investigate how people interact with phishing attacks, and mitigation techniques.

We saw that Twitter's free Streaming API is a popular data source amongst researchers and that the cost of the Twitter Decahose and Firehose streams are too expensive for most academics.

Blacklists typically constitute a reliable foundation for ground truth in existing studies – however, delays must be considered. Complete ground truth coverage is often unrealistic. Therefore, blacklists that specialise in specific attacks (e.g. phishing) usually provide effective cover.

Phishing and malware defence – typically organised into the categories: prevention and detection – often involve educating users as a prevention strategy and machine learning as a detection technique.

We explored existing research into web browser phishing detection, heuristic phishing detection, and warning effectiveness – all relating to Chapter 7: *Web Browser Phishing Detection*.

Existing studies can provide additional context to our results. We explored topics such as information credibility and propagation to understand how victims may encounter threats on Twitter. This can help paint the "bigger picture"; understanding where our results fit within the wider environment.

We explored how existing studies approach various ethical considerations of cyber security research. Existing studies identify best practices for ethics and encourage a consistent approach to ethical considerations. We also saw advice for responsibly sharing measurement data.

Finally, we evaluated existing studies to highlight key limitations, such as methodologies used to measure metrics in order to evaluate effectiveness. The main limitations we identify in existing literature, linked to our research question(s), are summarised in Table 3.4.

4
Design & Implementation

OUTLINE
This chapter details the design, methodology, and implementation of our empirical measurement studies. We present our measurement system: ***Phish****alytics*; including the design architecture, core systems, and shared components. We discuss the chronological order of our studies, key design decisions, ethical considerations, and our test-driven development process.

Towards the end of this chapter we describe the methodology of our 4 main studies (Chapters 5 to 8). Finally, we present our infrastructure and technical implementation details.

Having explored existing research, we see the following omissions in current knowledge:

- Research into 3 phishing blacklist we use (GSB, OP, and PT): uptake, dropout, typical lifetimes, and overlap of URLs in these blacklists. Existing studies have explored some aspects of some of these blacklists, but not all 3 for our research questions.
- We identify existing literature that investigates Twitter's use of blacklists and delay times as part of a broader research study on spam [110]. We want to repeat part of their study, to determine if the results have changed, and also further explore how effectively Twitter uses blacklists. We also want to carry out a novel, fine-grained and in-depth study of phishing and malware on Twitter.
- Research that investigates Twitter's URL shortener (*t.co*), how effective it is at protecting Twitters uses against phishing and malware attacks, and how it compares to other blacklists.

The methodology and experiments we plan, and the infrastructure we build, are designed to answer our core research questions and help to fill these ommisions in existing literature. We will also explain some of our key design decisions.

Some of the key challenges that can arise while working with APIs during a longitudinal measurement study include the ever-changing landscape that is being measured, along with changes to the various APIs themselves. One of our studies (Chapter 6: *Time-of-Post Twitter Study*) leverages a version of GSB which did not enforce a rate limit on lookups at the time when we carried out that specific study. This meant we could lookup all URLs in our study to determine how many resided in GSB. However, after we completed this study, GSB implemented a rate limit of 10,000 lookups per day. Therefore, our studies that took place after this new rate limited had been introduced use an altered methodology to work around this rate limit (Chapter 8: *Time-of-Click Twitter Study* and Chapter 5: *Blacklist Analysis Study*).

4.1 Phishalytics Design Overview

Phishalytics will collect data from various sources, store that data in a manner which allows efficient lookups, then perform analyses on that data. We also need to be aware of ethical and privacy considerations, professional standards, and hardware / software limitations.

Phishalytics consists of 4 core systems: the Blacklist Analysis System (BAS), the Phishing & Malware Tweet Detection System (PMTDS), the Web Browser Testing Suite (WBTS), and the Twitter URL Shortener Investigation System (TURLSIS). These 4 core systems are described in the subsequent sections (4.1.1 to 4.1.4). The overall architecture of **Phish**alytics can be seen in Figure 4.1. Table 4.1 shows which **Phish**alytics systems are used in which of our studies and thesis chapters.

In addition to the 4 core systems already mentioned, **Phish**alytics also contains a number of shared components. These include the Tweet History Search System (THSS), the URL Click Data Lookup System (CDLS), the Blacklist Update and Lookup System (BULS), the URL Redirection Chain Extractor System (RCES), and Tweet Collection System (TCS). These shared components are described in Section 4.1.5.

Key data sources that we use for our experiments include: Twitter's API, Google's Safe Browsing (GSB) API, and the PhishTank (PT), and OpenPhish (OP) data feeds. For our measurement studies we build an infrastructure that is designed to work around some limitations of the data sources we use. These limitations include: Twitter's data feed is a "small sample" of all global tweets, the GSB API blacklist, in some of our studies, is limited to

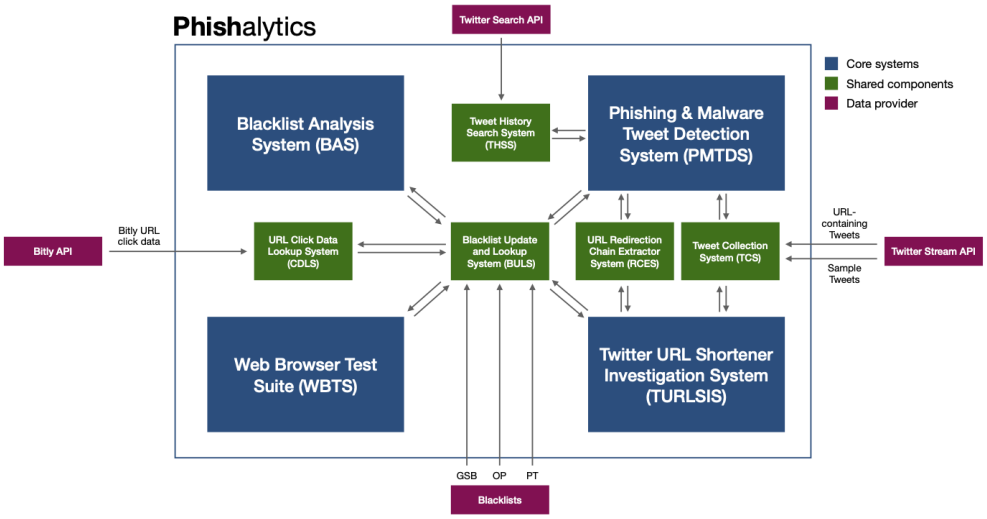


Figure 4.1: **Phish**alytics design architecture.

System	Study	Chapter
BAS	Blacklist Analysis Study	5
PMTDS	Time-of-Post Twitter Study	6
WBTS	Web Browser Phishing Detection	7
TURLSIS	Time-of-Click Twitter Study	8

Table 4.1: **Phish**alytics core systems linked to our relevant studies and thesis chapters.

10,000 daily lookup per day, and our Twitter URL Shortener Investigation System (TURLSIS) cannot send too many HTTP requests per second because this would flood Twitter's servers. The design decisions and methodology described in this section ensure that, despite these limitations of the data sources we use, our measurement framework, and experiments we run on this framework, produce accurate results to answer our research questions. The implementation details of the methodology described in this section are explained in more technical details in Section 4.8: *Infrastructure & Implementation*.

4.1.1 Blacklist Analysis System (BAS)

In Chapter 5: *Blacklist Analysis Study* we use BAS to analyse 3 key blacklists: GSB, OP, and PT. We investigate URL uptake, dropout, typical lifetimes, and overlap. These characteristics of the blacklists help us to evaluate the

effectiveness of the blacklists. BAS interacts with the BULS component to retrieve information about each of the 3 blacklists. BAS then analyses the blacklists' information through a number of different experiments to determine the characteristics of the blacklists. The methodology we use in BAS is described in Section 4.7.1.

4.1.2 Phishing & Malware Tweet Detection System (PMTDS)

In Chapter 6: *Time-of-Post Twitter Study* we use PMTDS to investigate how effective Twitter's use of blacklists is at protecting its users from phishing and malware attacks. Our study focuses on the delay period between an attack URL first being tweeted to appearing in one of the 3 blacklists. PMTDS interacts with the shared components: TCS, RCES, and BULS. PMTDS does this by checking all publicly tweeted URLs in the TCS component for blacklist membership in the BULS component. The redirection chains for all URLs in the TCS component are extracted via the RCES component. PMTDS then carries out a number of measurements and experiments to investigate how effective Twitter's use of blacklists is in protecting its users from phishing and malware attacks. The methodology we use in PMTDS is described in Section 4.7.2.

4.1.3 Web Browser Testing Suite (WBTS)

In Chapter 7: *Web Browser Phishing Detection* we use WBTS to test the detection rates of popular web browsers across different operating systems. This section gives an overview of the design for WBTS. The methodology for WBTS is described in Section 4.7.3 and the infrastructure and implementation details are described in Section 4.8.9.

WBTS comprises 4 core components: the Master Controller (MC), Test Machines (TMs), a Monitoring System (MS), and the Test Suite Software (TSS). The architecture design for WBTS can be seen in Figure 4.2. The 4 core components, along with their roles, are described in Table 4.2.

One of the main problems encountered during the tests was that, occasionally, the system for testing a web browser would fail. This could have been for any number of reasons such as the browser failing to load, keyboard shortcut not working, application unavailable for focusing etc. Detecting these failures was problematic since it was not always feasible to watch the TMs constantly. Therefore the Monitoring System (MS) provided an effective solution to alert the author to any problems.

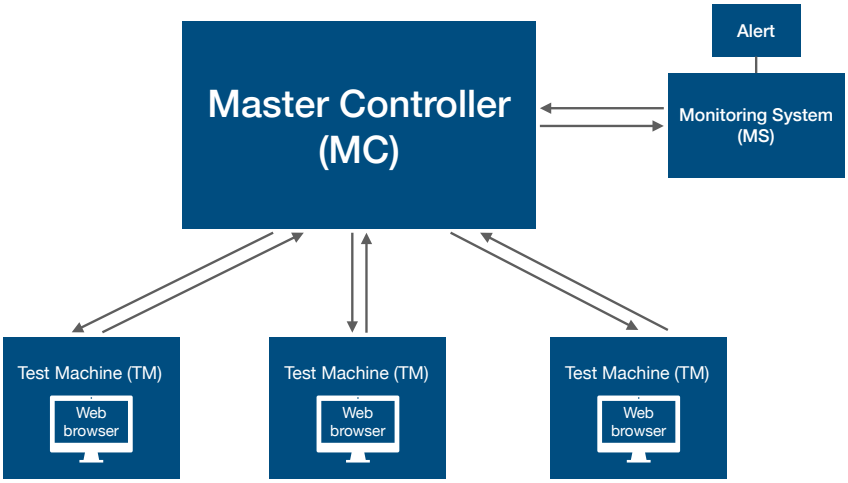


Figure 4.2: Web Browser Testing Suite (WBTS) design architecture.

Component	Role
Master Controller (MC)	Extract phishing URLs from data source, orchestrate tests (on TMs), collate data, analyse data, produce statistics and reports
Testing Machines (TMs)	Run various operating systems and web browsers in a safe environment; execute TSS to determine web browser detection rates (when instructed by MC)
Monitoring System (MS)	Provide a monitoring and alert system to detect errors and improve recovery time from failed tests
Test Suite Software (TSS)	Custom-built software tailored to each specific operating system and component; runs on MC, TMs, and MS

Table 4.2: Overview of the Web Browser Testing Suite (WBTS) system core components.

4.1.4 Twitter URL Shortener Investigation System (TURLSIS)

In Chapter 8: *Time-of-Click Twitter Study* we use TURLSIS to investigate how effective Twitter's URL shortening service (*t.co*) is at protecting Twitter users from phishing and malware attacks. TURLSIS interacts with the same shared components as PMTDS, that is: BULS, RCES, and TCS. TURLSIS also interacts with PMTDS to provide additional functionality. TURLSIS

does this by checking all tweets that contain phishing URLs (from PMTDS) and checking which URLs have been blocked by Twitter at time of click. The methodology used in TURLSIS is described in Section 4.7.4 and the technical implementation details are described in Section 4.8.8.

4.1.5 Shared Components

Tweet History Search System (THSS)

The THSS component establishes historical context for tweets to determine when a given tweet first appeared on Twitter's platform. THSS receives its data from Twitter's search API – which is external to **Phish**alytics. The THSS component is used by the core system: PMTDS. The technical implementation details for the THSS component are described in Section 4.8.7.

URL Click Data Lookup System (CDLS)

The CDLS component provides click data for blacklisted URLs that have been shortened with Bitly. CDLS receives its data from Bitly's API – which is external to **Phish**alytics. The CDLS component is used by the shared component BULS and the core system PMTDS. The technical implementation details for the CDLS component are described in Section 4.8.5.

Blacklist Update and Look-up System (BULS)

The BULS component stores, updates and performs lookups against popular blacklists. It receives its data from 3 blacklists: GSB, OP, and PT. These blacklists are external to **Phish**alytics. The BULS component is used by all 4 of the **Phish**alytics core systems. The technical implementation details for BULS are described in Section 4.8.6.

URL Redirection Chain Extractor System (RCES)

Across all our studies, for any tweeted URL, there could be a number of hops or redirections that are made before arriving at the final landing page. For this reason a redirection chain extractor is used to check each URL contained within in a redirection chain against each of the blacklists. The technicalities of this redirection chain extractor system are explained in more detail in Section 4.8.4. The RCES component expands all URLs that are contained within a URL chain. The RCES component allows core **Phish**alytics systems to check URL redirection chains for blacklist membership. The RCES component is used by the core systems: PMTDS and TUSIS in Chapters 6 and 8 when we investigate phishing on Twitter.

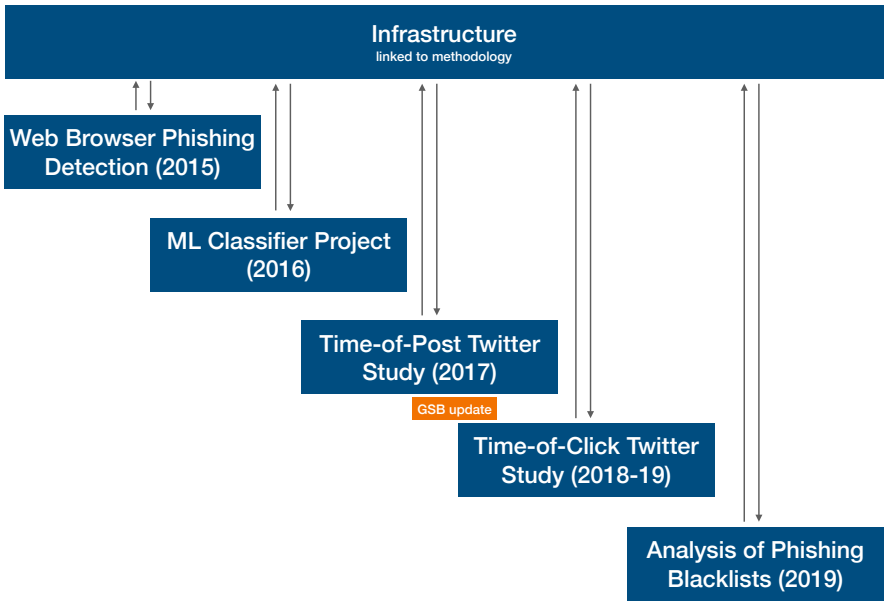


Figure 4.3: Timeline of our research projects.

Tweet Collection System (TCS)

The TCS component receives its data – live tweets – from Twitter's stream API. Twitter's stream API is external to **Phish**alytics. These tweets are saved into a database. The TCS component receives 2 sources of live tweets: sample and URL-containing. The implementation details for the TCS component are described in Section 4.8.3. The TCS component is used by the core systems PMTDS and TUSIS in Chapters 6 and 8 when we investigate phishing on Twitter

4.2 Timeline of Research Projects

To improve the overall narrative and comprehensibility of this thesis, our studies are not presented chronologically. Figure 4.3 shows the chronological order of our projects. Chronologically, we conducted our analysis of phishing blacklists (2019; Chapter 5: *Blacklist Analysis Study*) last. However, to present a clearer understanding of blacklists, and correlate certain findings with later chapters, we present our analysis of phishing blacklists (2019) first. Following this, we present our time-of-post Twitter study (2017; Chapter 6: *Time-of-Post Twitter Study*) which includes a key finding: large numbers of phishing and malware URLs are posted to Twitter.

To provide context to the time-of-post Twitter study (2017), we examine the effectiveness of web browser phishing detection in our 2015 study

(Chapter 7: *Web Browser Phishing Detection*). Therefore exploring what impact the results from our 2017 study might have on user safety in terms of exposure to phishing URLs. We follow-up on our 2017 findings in our time-of-click Twitter study (2018-19; Chapter 8: *Time-of-Click Twitter Study*) by investigating Twitter's approach to protecting users from phishing and malware URLs at time of click.

Our ML classifier project (see Section 4.6.2) took considerable time to plan, research, implement, and test. The project ultimately provided a motive for our subsequent measurement studies – and did not directly result in a research study or publication. We include it in the timeline diagram to show how it impacted and influenced other projects. The ML classifier project initially connected to the existing web browser detection system to test how effective web browsers were at detecting "fresh phish". That is, all URLs classified as phishing by the ML classifier were immediately sent to the web browser testing suite for analysis. See Section 9.9 where we discussion our ML classifier and future work.

Understanding that our studies are not presented chronologically helps to explain certain discrepancies between contiguous chapters. For example: we did not factor in certain key findings from our phishing blacklist analysis study (2019) – such as OP's faster phishing URL detection compared to PT – in the time-of-post Twitter study (2017). We could have designed an experiment with this finding in mind to determine if the OP blacklist is more effective on Twitter due to its detection speed, etc.

4.3 Key Design Decisions

Many of our key design and methodology decisions were influenced by the knowledge gained from existing literature. We also replicate some experiments found in existing literature to determine how the measurement landscape has changed over time and if results are still similar.

As we saw in the previous chapter, the vast majority of existing studies, that leverage Twitter's streaming API, use Twitter's free version and ***not*** the paid Decahose (10%) or Firehose (100%) versions. We decided to also use Twitter's free stream as a data source. Our reason for using Twitter's free stream, as per other academic researchers, is cost. We contacted Twitter for a quote to access the Decahose feed and received a figure that was considerably beyond our budget. We utilise a number of Twitter's other APIs, such as their Search API and Hashtags API, in order to alleviate any bias that could occur as a result of using Twitter's smaller data feed.

We also saw in the previous chapter how important a comprehensive and reliable ground truth is. In our study, we want to determine which tweeted URLs are phishing or malware, which requires a source of ground

truth. We decided to leverage blacklists for this ground truth, since blacklists provide an efficient and reliable source. Another source of ground truth would be to leverage a honeypot. However, we decided that the main focus of our study would be on phishing websites. Phishing attacks are very difficult to detect using honeypots due to their use of psychological social engineering techniques (such as lack of knowledge, distractions, etc) – which can work well on humans, but not computers. Malicious websites are easier to detect using honeypots since their malicious activity (such as executing malicious code on the honey client machine) can be objectively monitored and detected by the machine. Therefore we decided that blacklists would provide an adequate ground truth for our study.

We decided to use the URL blacklists: GSB, PT, and OP, because they are specialised phishing blacklists (and, in the case of GSB, also contain malware URLs). As we saw in previous literature, a complete and comprehensive ground truth can be difficult to achieve and often is not required. Because we will be analysing just phishing and malware URLs in our study, leveraging these blacklists provides us with the specialist ground truth we require. Our decision to choose URL blacklists was because we receive a source of ULRs (from Twitter) – which means we can lookup tweeted URLs in the blacklists to determine membership. We did not consider IP blacklists to be suitable for our research due to, as seen in previous literature, IP addresses changing frequently which would make blacklist membership ineffective. Another reason for choosing the 3 blacklists is that they offer open access and their rate limits are compatible with our requirements of looking up large numbers of URLs – although, as mentioned, we did experience some problems with GSB's rate limit.

Some previous studies criticise researchers for not including enough details of their methodology and implementation in their write-ups. Therefore, this thesis aims to take a thorough and detailed approach to explaining our methodology, implementation, and technical setup.

4.4 Data Collection

Our measurement studies require a source of live tweets from Twitter. To achieve this we setup two sources of incoming tweet feeds using Twitter's Stream API. The first stream provides a small sample of all global tweets and the second stream provides a sample of all global tweets that contain one or more URLs. The first stream is used to provide a general picture of Twitter activity during collection – and to help alleviate any bias – and the second stream is used to carry out our measurement analysis. Our analysis involves searching for tweeted URLs in various blacklists, measuring blacklist delay times, and determining which tweeted URLs are blocked by twitter. Both

of these tweet streams are saved locally in a database.

We also require a way to store and search various blacklists. We retrieve a copy of each blacklist (GSB, PT, and OP) and store them in our database. We use GSB's API to retrieve the latest copy of the blacklist. Each URL within the GSB blacklist is encrypted as a SHA-256 hash prefix. To determine if a given URL resides in the GSB blacklist: the URL's full hash is calculated, then this URL's hash *prefix* is checked for membership in the local copy of the blacklist. If there is a match (on the hash prefix), then the full URL hash is retrieved from Google's servers to determine if there is a match on the our local, precalcualted full URL hash. If the full URL hash matches then that means the URL resides in GSB. URLs in GSB are categorised as either social engineering, malicious, unwanted software, or potentially harmful application. The blacklists PT and OP are provided as JSON files which we retrieve directly from their websites – the technical details of this can be seen in Section 4.8: *Infrastructure & Implementation*.

Another requirement for our study is the ability to check all tweeted URLs, collected from Twitter's Stream API, to determine if they have been blocked by Twitter. All URLs tweeted on Twitter's platform are shortened via Twitter's URL shortener, *t.co*. These *t.co* shortened URLs are stored in our database alongside the corresponding full URLs. We retrieve the redirection chain of each *t.co* URL in our dataset to determine which URLs have been blocked by Twitter. Full details of this are in Section 4.8.8: *Twitter URL Shortener Investigation System*.

4.5 Ethical Considerations

Building on from what we have already discussed and explored in Sections *1.7*: *Ethics* and *3.9*: *Ethics*, we now describe the main ethical considerations that we incorporate into our experiments and research.

All data sources that are used in this thesis are public. These data sources are provided by reputable organisations and include Acceptable Use policies, which we adhere to throughout this project.

Twitter is a social network that consists of people publicly broadcasting their own messages (*i.e.* tweets); it's a microblogging platform. By default, Twitter users' tweets are public, and Twitter shares these public Tweets with third parties via its API service. Upon signing-up to the service, all Twitter users agree to Twitter's Terms of Service [277], which states that:

> "By submitting, posting or displaying Content on or through the Services, you grant us a worldwide, non-exclusive, royalty-free license (with the right to sublicense) to use, copy, reproduce, process, adapt, modify, publish, transmit, display and distribute such Content in any and all media or distribution methods now

> known or later developed (for clarity, these rights include, for example, curating, transforming, and translating). This license authorizes us to make your Content available to the rest of the world and to let others do the same. You agree that this license includes the right for Twitter to provide, promote, and improve the Services and to make Content submitted to or through the Services available to other companies, organizations or individuals for the syndication, broadcast, distribution, Retweet, promotion or publication of such Content on other media and services, subject to our terms and conditions for such Content use"

As we have seen in our previous ethics sections, datasets that consist of human-produced data require ethical care and consideration. Therefore, because our dataset of tweets is generated by humans, we need to address how we shall process this data. Twitter users have agreed to share submitted content with Twitter and that Twitter may share this data with third parties. In addition to Twitter's Developer Agreement, we will process the data we receive from Twitter in an objective and scientific manner in a way that does not cause harm or distress to any Twitter users. Our aim for processing this data is to answer our research questions in order to gain scientific discoveries. Data we receive from our various sources is only used for the purpose of our research and experiments.

In terms of privacy: the only personal identifiable information we collect from Twitter is usernames – which Twitter users agree to make public when they register. It is important to note that any Twitter user can choose to set their account to "private", in which case their tweets will not be shared with Twitter's third parties and therefore we do not receive them in our dataset. We do not collect IP addresses or geographical information of Twitter users in our dataset.

It is important to note that a given Twitter user may have tweeted phishing or malware URLs because their account has been compromised. Therefore the privacy of the user in such a situation needs to be considered. In this situation, the individual that owns this Twitter account is probably unaware that their account has been hijacked. Consequently, in addition to Twitter's Acceptable Use policy, and to further maintain user privacy, this thesis does not specifically identify individual Twitter accounts that we observe to have shared blacklisted URLs. Also, since twitter usernames can easily be used to identify individuals that own the accounts, this additional ethical consideration further enhances the privacy of the people that contributed to the information in our dataset.

The PhishTank and OpenPhish blacklist datasets are public. Therefore, all URLs contained within these datasets are visible to anyone, worldwide. The GSB blacklist is publicly available, however, all URLs residing in the

blacklist are hashed – therefore all URLs in GSB are private. Our infrastructure is designed so that all data we receive from PT, OP, GSB, and Twitter is transmitted via secure SSL connection and stored securely on our system.

4.5.1 Ethically Responsible Data Sharing

Our infrastructure has been designed so that we can share our dataset(s) with other researchers in a way that Twitter usernames and URLs/domain names are hashed to maintain privacy. This ensures that each username or URL/domain is uniquely identifiable within the dataset – but without revealing the actual source. Although our data has been acquired from publicly available sources – and therefore does not require anonymising – it does mean that we can anonymise our data before sharing if the need arises (*e.g.* via risk assessment, etc).

4.6 Test-Driven Development

By employing a test-driven development process, we were able to ensure that our infrastructure could accurately collect and measure the empirical data we required for our experiments. This involved testing our infrastructure on smaller sets of data, or running the system for shorter periods of time, to carry out tests and fix any errors. This process was time consuming and laborious. However, the result allowed us to create a reliable and sturdy measurement infrastructure that could run, uninterrupted and continuously, for months at a time without fault. This was crucial, since some of our experiments could not resume if they were interrupted.

This section describes some examples of where parts of our infrastructure required a more thorough test-driven development process. Since our infrastructure incorporates and relies upon many external libraries and services, there were some situations where we needed to make suitable modifications and adjustments or ensure that certain services were reliable. There were also occasions where we needed to alter the direction of our research as a result of problems that we occurred.

4.6.1 GSB Python library

We use the Python library *gglsbl* [84] to implement our local copy of GSB. An important modification was made to the *gglsbl* library to improve lookup times for large numbers of URLs. In the library, the method *lookup_url()* is used to lookup an individual URL in the local hash prefix database. It does this by performing a search in the SQLite database for that URL's hash prefix. This lookup technique caused a bottleneck when testing the system on large volumes of URLs, therefore we modified the library to output a

Python *dictionary* (hash table) of all URL hash prefixes. Our system can then perform a lookup for each tweeted URL's hash prefix against this *dictionary*. Since the Python *dictionary* implementation uses a hash map, the typical time complexity for this lookup is constant; $O(1)$. This means lookups are considerably faster than using the off-the-shelf version of the GSB library.

4.6.2 Machine Learning Classifier

One of our early research aims was to determine blacklist delays by leveraging a source of "fresh phish". Our intention was to conduct a lifecycle analysis study. Our methodology involved creating a machine learning classifier to automatically detect tweets that contained phishing URLs. These URLs would be frequently checked for blacklist membership. Delay times could then be calculated on these blacklisted URLs.

We designed a machine learning classifier, based on PhishAri (Aggarwal *et al.* 2012 [4]), to detect tweets that contain suspected phishing URLs. Whilst the classifier we built appeared to work, it was detecting large volumes of tweeted URLs that already resided in the GSB blacklist. These early results contradicted prior knowledge ([200, 201]) that Twitter did not allow blacklisted URLs to be tweeted. Therefore we suspected that our classifier was not accurate. To test the accuracy of our classifier, we defined our first measurement study to establish the facts: how many tweets contain blacklisted URLs? In this experiment, we also measured the delay times for URLs – that did not reside in a blacklist at time of tweet – to appear in a blacklist after being tweeted.

Ultimately, our first measurement study grew, became more complex, and provided a rich set of data for answering our research questions. The results of our first measurement study also provided more areas to explore. Therefore we carried out various extensions and follow up experiments to our initial measurement study. Consequently, we did not return to the original machine classifier for detecting "fresh phish" because we were able to answer our research questions without it. We believe the data we collected from our measurement studies presents a lifecycle analysis of blacklisted phishing and malware URLs on Twitter.

Many of the features we built for the machine learning part of our research ultimately evolved into other research experiments that were used for this thesis. In particular, the original infrastructure we built for designing and testing the machine learning classifier ultimately developed into our measurement experiments. Specific parts of the system, such as the tweet collection system, along with the meta data it collects, and the blacklist data collection system, were modified for our other experiments. Since our machine learning classifier provided motivation for our measurement

Category	Feature
URL Based	Length of URL
	Number of dots in URL
	Number of redirects
	Levenshtein distance from tweeted to landing URL
	Presence of conditional redirects (e.g Safari redirected to different page to Google bot)
	Number of subdomains in URL
	Number of forward slashes in URL*
Tweet Based	Number of @ tags used in tweet
	Number of # tags used in tweet
	Presence of trending hashtags in tweet
	Length of tweet
	Number of RTs
	Position of # tags
Network Based	Number of followers
	Number of friends
	Follower-folowee ratio
	User profile description has text (true/false)
	Age of account
	Number of tweets
	Number of lists user appears in **
	Number of tweet favourites (from other users)*
	Number of tweets favourited (by user)*
	User has default profile (true/false)*

Table 4.3: Twenty-three-feature model training features for our machine learning classifier.
* New features we add in addition to those in [4].
** Feature we improve from [4].

studies, and played a key role in our research direction, we provide details of the classifier in the rest of this section.

We will now outline the details of our machine learning classifier that is designed to detect phishing tweets. Our classifier uses 1 of 2 feature sets: either the 23-feature model or the 6-feature model. The 23-feature model consists of 3 categories: URL-based (7 features), tweet-based (6 features), and network-based (10 features). Table 4.3 summarises the features of the 23-feature model; Table 4.4 the 6-feature model. 19 of our features are based on the 22-feature model in [4] with the exception of all 3 features in their WHOIS category because we were unable to access WHOIS domain information during our experiments. We also improve their feature “part of

Number of friends
Number of followers
Follower-folowee ratio
Age of account
Number of @ tags used in tweet
Number of hash tags used in tweet

Table 4.4: Six-feature model training features for our machine learning classifier.

list" to "number of lists user appears in" because we believe this increases the feature richness. We provide 4 additional features, marked with an * in Table 4.3. The reason for splitting into two models (23 and 6-feature) is that [4] produce a 7 feature model based on an evaluation of the most informative features for detecting phishing tweets in their 22-feature model. We were unable to include 1 of the features from their 7-feature model due to the aforementioned WHOIS limitation.

Our model uses the random forrest algorithm for classification. This is based on the evaluation and results from [4]. For each data point to be classified, the random forrest algorithm randomly chooses a subset of features (from 4.4 or 4.3) for classification. The algorithm selects the most relevant features of the data point, therefore improving the predictive accuracy and controlling over-fitting. We trained our classifier on tweets collected during 25, 26 and 27 March 2017 which consisted of 453 phishing tweets. These tweets were labelled by checking all tweets during this timeframe for URLs that were blacklisted. A sample of 451,000 benign tweets were then evenly selected throughout this timeframe. The ratio of benign to phishing tweets altered the weighting of our classifier, which we discuss later. We then tested the classifier on approximately 24 hours' worth of tweets whose timestamps immediately followed the training data. This training data consisted of 1 million tweets: 999,861 benign; 139 phishing.

To assess the accuracy of our classifier we use the confusion matrix seen in Table 4.5, where TP is true positive, FN is false negative, FP is false positive, and TN is true negative. We then use the confusion matrix evaluations in 4.6. Table 4.7 shows the confusion matrix for our 23-feature model after testing. As a comparison, the same test was carried out using the 6-feature model; the confusion matrix is shown in Table 4.8. The accuracy of our 23-feature model classifier to predict phishing tweets is as follows: 28% sensitivity (or true positive rate), 99% specificity (or true negative rate), 0.00016% fall-out (or false positive rate), and 72% miss-rate (or

		Predicted	
		Phishing	Benign
Actual	Phishing	TP	FN
	Benign	FP	TN

Table 4.5: Confusion matrix for classification.

Evaluation	**Equation**
Sensitivity (or recall; hit rate; true positive rate (TPR))	TP / (TP + FN)
Specificity (or selectivity; true negative rate (TNR))	TN / (FP + TN)
Fall-out (or false positive rate (FPR))	FP / (FP + TN)
Miss-rate (or false negative rate (FNR))	FN / (FN + TP)
Precision (or positive predictive value (PPV))	FN / (FN + TP)
Accuracy (ACC)	(TP + TN) / (P + N)

Table 4.6: Confusion matrix evaluators.

		Predicted	
		Phishing	Benign
Actual	Phishing	39 (28%)	100 (72%)
	Benign	16 (0.0016%)	999845 (99%)

Table 4.7: Confusion matrix for our 23-feature model classifier results on 1 million testing tweets (999,861 benign; 139 phishing).

		Predicted	
		Phishing	Benign
Actual	Phishing	14 (10%)	125 (90%)
	Benign	67 (0.0067%)	999794 (99%)

Table 4.8: Confusion matrix for our 6-feature model classifier results on 1 million testing tweets (999,861 benign; 139 phishing).

false negative rate). This gives an overall accuracy of 99%. The precision (positive predictive value) is 71% on phishing classification and 99% on benign classification.

Table 4.9 shows the confusion matrix for the machine learning classifier in [4]. When comparing the results of our 23-feature classifier to [4],

		Predicted	
		Phishing	Benign
Actual	Phishing	92.31%	7.78%
	Benign	9.60%	94.41%

Table 4.9: For comparison: confusion matrix for the classifier used in Aggarwal *et. al.* (2012) [4].

a noticeable difference is that [4] has a higher sensitivity (true positive rate) for phishing classification, 92.31% compared to our classifier's 28%. However, our classifier's specificity (true negative rate) is 99% compared to 94.41% in [4]. The reason for this is that we adjusted the weighting of our classifier to increase the accuracy of the specificity (true negative rate). In our dataset – a typical sample of tweets from a 24-hour period – the weighting of benign to phishing tweets is considerably higher; there are 7,193 benign tweets for every phishing tweet. We needed our classifier to more accurately identify benign tweets so that there was less benign "noise" in our model's classifications. The reason for this is that our measurement infrastructure at the time could only handle a smaller number of predicted phishing URLs. Our measurement infrastructure could process 55 predicted phishing URLs (39 true positives; 16 false positives) – as produced from our 23-feature classifier (4.7). Wheres [4] would have produced 96,115 predicted phishing URLs (128 true positives; 95,987 false positives). Despite this trade-off between sensitivity and specificity, our classifier has a higher accuracy – 99% – compared to 92.52% in [4].

Based on our 23-feature model, the confusion matrix in 4.7 shows that 39 tweets were correctly classified as phishing during a 24-hour period. However, all of these tweets contained URLs that already resided in the GSB blacklist. It is this result that contradicted prior knowledge that Twitter does not allow blacklisted URLs to be tweeted. This motivated our subsequent measurement studies to determine the facts surrounding blacklisted URLs on Twitter.

4.6.3 Hash Collisions in GSB

As previously described, the GSB blacklist consists of hashed URLs in order to maintain privacy for the whole dataset. Our local copy of GSB is stored as a database of SHA-256 URL hash prefixes; the majority of the hash prefixes are 4 bytes (2^{32} bits). Due to these short URL hash prefixes there is likely to be an increase in the number of collisions as the size of the dataset grows. The average number of collisions in k samples, each a random choice among n possible values is:

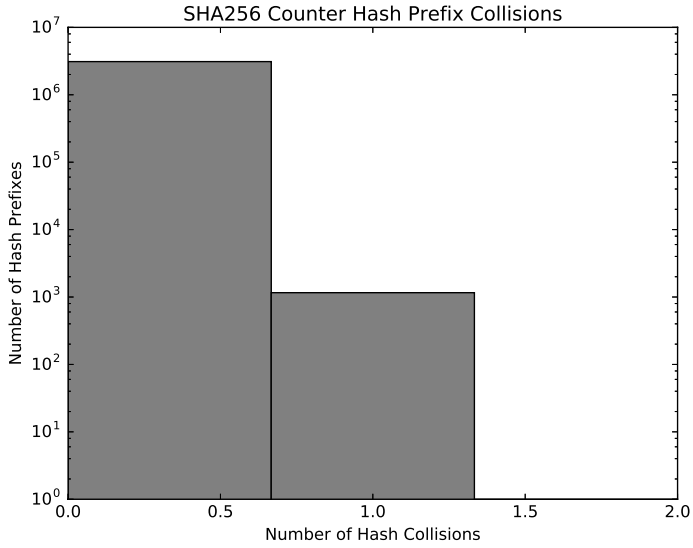


Figure 4.4: Histogram Showing Hash Prefix Collisions for a Counter of 3,107,744 SHA256 Hashes.

$$N(n,k) = k(k-1)/2n$$

In our dataset of 1,731,452 SHA-256 URL hash prefixes there will be approximately 349 collisions. Therefore our GSB measurement calculations are accurate to within 0.02%.

We are unable to see the total number of collisions within our GSB dataset because those collisions have already occurred. Therefore, to test our statistical model, we generated 3,107,744 random SHA256 hashes, then counted the total number of collisions on the prefixes of those hashes. The histogram in Figure 4.4 shows the SHA256 hash prefixes of a counter running up to a total of 3,107,744. The experiment was also repeated on the full hashes which resulted in zero collisions. Figure 4.5 repeats this experiment but with a counter of 31,077,440 to illustrate how increasing the number of hash prefixes affects the number of collisions – with some hash prefixes colliding more than once.

In our example set of 3,107,744 random SHA256 hashes, the calculation is as follows:

$$\text{(3,107,744)(3,107,744-1)}/(2(2^{32})) = 1124.3472879 \text{ collisions}$$

By calculating the number of hash collisions in our GSB dataset, and verifying with a statistical model, we are able to determine the accuracy of our results where hash collisions may occur.

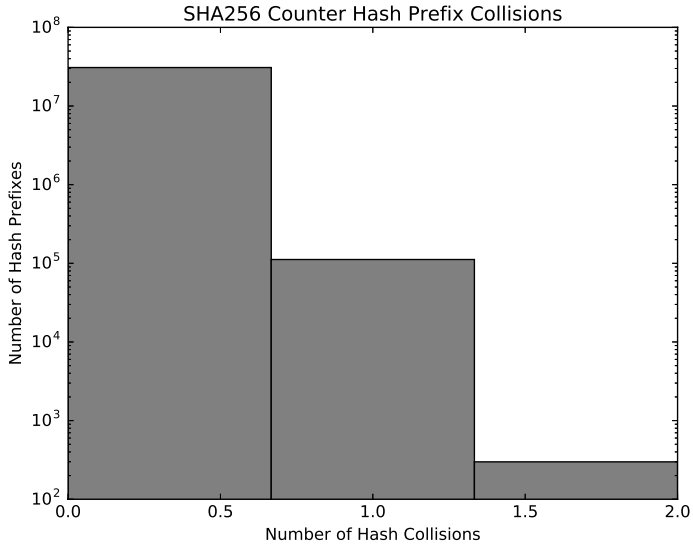


Figure 4.5: Histogram Showing Hash Prefix Collisions for a Counter of 31,077,440 SHA256 Hashes.

4.6.4 GSB Rate Limitations

As previously mentioned, GSB introduced a rate limit during one of our experiments. Although we had already completed one measurement study, we wanted to expand our results and also compare our existing results with additional data. The API rate limit was unexpected and meant we were unable to continue with one of our experiments with the methodology we were using. Therefore we designed an altered methodology and tested it on small sets of data or for small durations of time (e.g. a few hours). Part of this test-driven design process allowed us to determine if our new methodology would be able to check enough URLs – without reaching the rate limit – to answer our research questions whilst maintaining a high level of accuracy.

4.6.5 Update Frequencies

Many of our data sources feature limitations on how frequently we can access them in order to update our system. For most of these, it is simply a case of predetermining how often will retrieve certain data, as is the case with the OP and PT databases.

For the GSB rate limit, we needed to predict how frequently we would send a request to Google's API. The GSB API features 2 types of requests: updating our local copy of the blacklist, and retrieving a URL's full hash (after a local URL hash prefix match) to determine if there is a match.

In order to estimate how frequently we would need to lookup URL hash prefixes in GSB, we analysed the typical percentage of blacklisted URLs within each set. We also needed to include hash collisions that resulted in false positives – i.e. benign URL hash prefixes that collided with malicious URL hash prefixes – since these would constitute a request and therefore count towards out our daily rate limit.

Our technique to determine if a URL has been blocked by Twitter involves checking the *t.co* URL to test if it has been redirected to a webpage that informs us it is blocked. This process involves visiting large numbers of *t.co* URLs to determine their statuses. We achieve this by leveraging CPU threading on our measurement infrastructure in order to simultaneously process multiple URLs at a time. However, this process of sending multiple requests can overload Twitter's servers. Therefore part of our test-driven design process involved carefully testing and adjusting our system to ensure we can test large volumes of URLs without overloading Twitter's servers. Technical details of this system can be seen in Section 4.8.8: *Twitter URL Shortener Investigation System* of the Implementation Section.

4.6.6 Error Reporting

To improve the efficiency of the test-driven development process, whenever an error was detected, an email was sent to the author, detailing the error. This meant that the measurement infrastructure interface did not need to be monitored 27/4 to detect, debug, and fix errors.

A problem that arose due our error reporting setup was that, when a critical error ocurred, a large numbers of emails would sometimes be sent to the author in a short amount of time. Once the specific error had been fixed, many emails would continue to be received by the author. This was because error reporting emails had been delayed by the recipient's mail server – therefore taking, in some cases, many days to arrive. This made the error reporting system misleading and confusing since it appeared an error was still present when it had actually been fixed. To fix this: timestamps were added to the contents of error reporting emails to determine which timeframe the emails related to. This improved the test-driven development process by making it much smoother and more efficient to detect and fix any bugs that arose.

4.7 Methodology Overview

An overview of our key methodology is described in this section. The specific methodology used for each study is described in the relevant chapters (5, 6, 7, and 8).

4.7.1 Blacklist Analysis Study

For our blacklist analysis study (Chapter 5: *Blacklist Analysis Study*) we use the Blacklist Analysis System (BAS) to update and track URL uptake and dropout for each of the 3 blacklists (GSB, PT, and OP). These systems regularly obtain the latest copies of the blacklists and stores them locally in a database on our system. With this data we can then measure, for each blacklist update, the number of URLs that are added and removed. We can also analyse the number of times each URL is added to and removed from a blacklist – and whether or not this happens multiple times. We can also measure the duration that each URL remains in a blacklist for and analyse any intersection between blacklists, timestamps for temporal analysis, volumes, etc.

Our justification for the duration of data captures in our study – 75 days – comes from a combination of existing studies and the limitations of our infrastructure. The duration of the measurement study in Zhang *Et al.* (2013) [310] was 7 days, Kuhrer *et al.* (2012) [148] 80 days, Grier *et al.* (2010) [110] 1 month. Therefore, we believe our duration of data capture is similar to most, and improves on some, existing studies. A limiting factor to our data capture duration is our infrastructure. As described previously (in Section 4.6: *Test-Driven Development*), it was time consuming to ensure our measurement studies remained running without errors. In many cases, when an error occurs in one of our measurement studies, the entire experiment has to be restarted.

4.7.2 Time-of-Post Twitter Study

Our Twitter blacklist delay study (Chapter 6: *Time-of-Post Twitter Study*) uses the Phishing & Malware Tweet Detection System (PMDS) before Google introduced a rate limit on the GSB API. We implement a fast and slow blacklist lookup system: the slow lookup system ensures we can check all URLs we have collected since the experiment began – but is time consuming. The fast lookup system increases our measurement granularity by regularly checking tweeted URLs for blacklist membership – but is limited to a subset of recently tweeted URLs. Timestamps for when URLs were tweeted and when URLs were added to the blacklist(s) are used to calculate delay periods. We also leverage Twitter's Search API to alleviate bias that may arise from using a "small sample" via Twitters Stream API – allowing us to determine tweet timestamps that may not have been included in the Stream API.

We use Bitly's API to determine URL click data for tweeted URLs that are blacklisted. Our justification using Bitly for click metrics is that previous studies [16] show Bitly to be the most popular URL shortening service; 50% of Twitter URLs were shortened with Bitly.

As in our blacklist analysis study in the previous subsection (4.7.1), our justification for the duration of data captures in our study – 2 months – comes from a combination of existing studies and the limitations of our infrastructure.

4.7.3 Web Browser Phishing Detection

Our web browser study (Chapter 7: *Web Browser Phishing Detection*) uses the Web Browser Testing Suite (WBTS). The master controller (MC) will extract a set number of URLs from one of the phishing website sources and save these URLs into a database. The MC will then send this set of URLs to the Testing Machines (TMs), which will individually open each phishing URL in a pre-determined set of browsers. The resulting phishing detection status (phishing or not phishing), as detected by the web browser, shall then be returned to the MC and saved into the database for future processing. The main services that the MC provides are:

1. Extract phishing URLs from pre-determined sources
2. Store phishing URLs and test data in a database
3. Send phishing URLs to TMs for testing
4. Retrieve results data from TMs and save into database
5. Process results from tests stored in database
6. Provide test status for Monitoring System (MS)

The visiting of phishing URLs in web browsers needs to be carried out in a natural way that mimics the behaviour of a human user. The reason for this is that the experiment should be as similar to a real human accessing a phishing website as possible. Therefore, specific features, such as timings and the way in which each web browser is opened, are carefully considered. Ultimately this was achieved by copying each phishing URL into the operating system's clipboard and then using the keyboard shortcut for paste ($ctrl + v$ or $cmd + v$) to enter the URL into the web browser.

An early problem that was detected was that many of the web browsers take at least a few seconds before a phishing website is detected. Therefore, after navigating to a phishing website, a delay of around 10 seconds is applied before assessing whether or not the website has been flagged as phishing. After some trial and error, the most reliable method for determining if a phishing warning has been triggered was to look at the web browser's title. All web browsers change the title of the current web page to that of a phishing warning when the detection has been triggered. Therefore the determination process was a case of checking each web browser's current page title against a pre-determined list of strings.

Bespoke Test Suite Software (TSS) was produced to automate the browser tests. Technical details for the TSS are described in Section 4.8.9. A high level overview of the main testing algorithm is outlined below:

1. Connect to MC and fetch new phishing URL to test. If no URL available, there is no job currently active, wait for 1 minute before checking again
2. For the new phishing URL retrieved from step 1, check this new URL has not just been processed. If it has, another test is probably running, wait 5 seconds before checking again for a new URL (allows for parallel testing)
3. Copy phishing URL to system clipboard
4. Open web browser or focus already opened web browser (this will be one of the pre-determined web browsers for the specific operating system) and wait 3 seconds (mimic typical human behaviour and also allow time for web browser to load)
5. Send keyboard shortcut for new tab ($ctrl + t$ or $cmd + t$) then wait 1 second for tab to load
6. Send keyboard shortcut for paste ($ctrl + v$ or $cmd + v$) then wait 5 to 10 seconds for website to load
7. Check browser title for phishing warning
8. Send phishing status along with web browser and operating system info to MC
9. Close web browser or browser tab with keyboard shortcut ($ctrl + w$ or $cmd + w$)
10. Repeat steps 4 to 9 for each web browser
11. Repeat steps 1 to 10 until there is no new phishing URL (as determined in step 1)

Due to the limitations of the computing power on the virtual machine, only one operating system could be running at a time. Therefore a batch of phishing URLs would be processed on one operating system. Once the test had finished, that operating system would be paused (its state saved on the virtual machine for quick start and stop) and the next operating system started.

4.7.4 Time-of-Click Twitter Study

In our Twitter URL shortener study (Chapter 8: *Time-of-Click Twitter Study*) uses the Twitter URL Shortener Investigation System (TURLSIS). We implemented changes to our methodology to allow for the newly introduced

GSB API rate limit. Key changes include looking up URLs for GSB blacklist membership over 24-hour and, additionally, 7-day periods. This study also analyses the redirection chain of Twitter's URL shortener (*t.co*) to determine if tweeted URLs have been blocked by Twitter.

Our justification for the duration of data captures in our study – 7 days and, additionally, 24 hours – comes from the limitations of the data source we were working with. As previously explained, the GSB rate limit changed during this study, therefore we designed a methodology that would work within GSB's new rate limit whilst still achieving our research aims and answering our research questions.

4.8 Infrastructure & Implementation

Having discussed the overall design and methodology for our experiments, this section explores the details of our technical implementation. We will explore the overall technical setup, along with specific libraries and functions of our infrastructure. The full codebase for **Phish**alytics is available on GitHub at:

`https://github.com/sjbell/phishalytics`

4.8.1 Architecture Overview

Our entire system is implemented on a virtual machine running the Ubuntu operating system, version 16.04 LTS, 8 core CPU, 24 GB RAM. The measurement framework is written in the programming language Python – which consists of approximately 77,259 lines of Python code across 156 Python files. In total, we collected over 1.5TB of data from Twitter, GSB, OP, and PT, during our measurement experiments and analyses.

4.8.2 Interface

Interacting with our measurement infrastructure system is carried out via an SSH connection in a terminal window. The server-side interface leverages GNU Screen [85], a window manager and terminal multiplexer application. Figure 4.6 shows a screenshot of our system during one of our measurement studies. The layout consists of 18 windows; 16 small and 2 large. The two larger windows display a development area and the system monitor (*htop* command: showing CPU and RAM usage, top processes, etc). As seen in Figure 4.6, the 16 smaller windows are labelled s1 to s16. These windows contain the following systems:

- s01: Twitter filter stream (tweets containing URLs). Each character in this window represents the following:

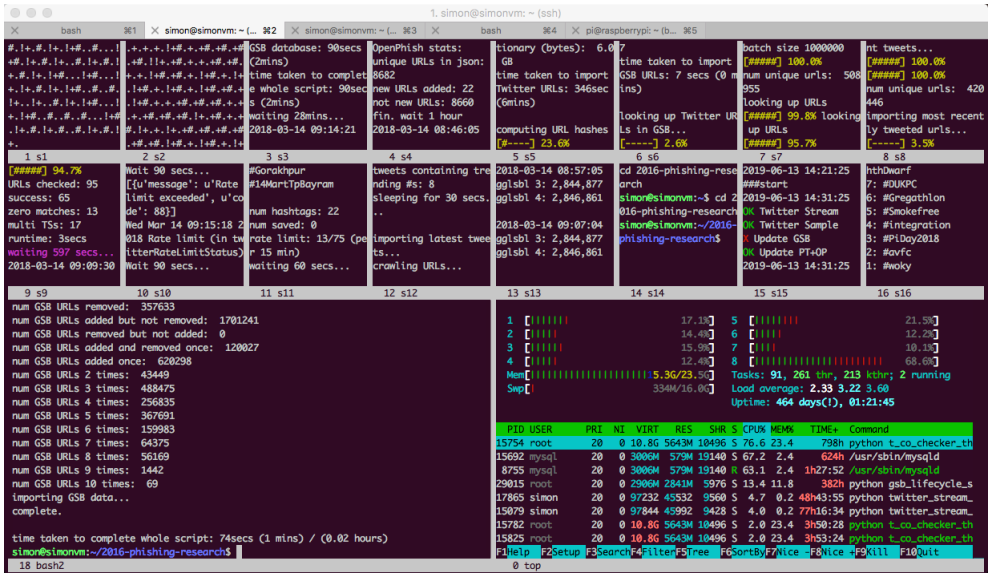


Figure 4.6: Screenshot of our measurement infrastructure system interface running in an SSH terminal window.

  - "#" the tweet that is about to be processed is a retweet
  - "." tweet received from Twitter Stream API for processing
  - "!" for each URL within this single tweet
  - "+" tweet saved to our system's database

- s02: Twitter sample stream (same characters as above)
- s03: Update our local copy of GSB blacklist
- s04: Update our local copies of PT and OP blacklists
- s05: Fast GSB Twitter URL lookup system
- s06: Slow (comprehensive) GSB Twitter URL lookup system
- s07: Slow (comprehensive) OP and PT Twitter URL lookup system
- s08: Fast OP and PT Twitter URL lookup system
- s09: GSB timestamp lookup system
- s10: Twitter search API lookup system
- s11: Retrieve and save current trending hashtags from Twitter API
- s12: Post Twitter collection processing (for metadata such as: lookup redirections chains, num URL hops, landing page URL, calculate Levenshtein distance, determine if trending hashtags used, etc)
- s13: Calculate, update, and compare GSB sizes
- s14: Not currently being used for the present study

- s15: Check everything is functioning correctly, check all feeds are live, etc. Send error notification emails to authors
- s16: Currently trending hashtags on Twitter for London

Not all currently running systems are displayed in windows s1 to s16. The development window (currently showing *18 bash2* in Figure 4.6) can switch between numerous windows that are running various systems and experiments. Windows s1 to s16 were mainly used to display systems that required frequent monitoring and also to provide some reassurance to the author whilst longitudinal measurement experiments were running. Many of the windows feature progress bars and colour coded statuses (such as s15) to aid efficient interpretation by the user.

In windows s1 and s2 we see some tweets that have been processed by our infrastructure but not saved to our database (each instance represented by a "."). This is when the URL in a tweet object does not contain an expanded URL. This occurs when the tweet contains a link to another tweet or Twitter profile – seen when a tweet "mentions" another twitter user or tweet. Our system drops these URLs because they are not relevant to our measurement study.

Window s10 illustrates a rate limit that has been reached for Twitter's Search API. The system will pause its current process and wait until the rate limit has been reset. S11 shows that the current rate limit for Twitter's Trending Hashtags API, at time of screenshot, was 13 out of 75 requests per 15 minutes.

4.8.3 Tweet Collection System (TCS)

Our Twitter collection system uses Twitter's Stream API, implemented via the *Tweepy* [274] library. After authorising *Tweepy* to access Twitter, the *sample()* and *filter()* methods are used to collect sample and URL containing tweets. The filter method uses keywords *"http"* and *"https"* to filter out tweets containing URLs. All data received from Twitter's Stream API, using these two methods, is stored in a MySQL [216] version 5.7.19-0ubuntu0.16.04.1 database in two tables for sample tweets and URL-containing tweets, respectively.

The track parameter is an array of search terms to stream:

```
1 stream = tweepy.Stream(auth, listener)
2 stream.filter(track=[''http,https''])
```

The sample method returns, approximately, a 1% sample of all global tweets:

```
1 stream = tweepy.Stream(auth, listener)
2 stream.sample()
```

4.8.4 URL Redirection Chain Extraction System (RCES)

The URL redirection chain extraction system uses Python's *Requests* library [235] to send an HTTP request for each URL using a Macintosh Safari user agent header so the request appears to come from a regular user via the Safari web browser. The reason for setting this header is so the request extracts the same redirection chain that a legitimate user would see and not a redirection chain that a bot would see – therefore reducing bias in our results. The *Request* library's *Response* object contains a *History* property which consists of a list of *Response* objects that were created to complete the HTTP request. This list is then used to extract the redirection chain for a given URL in our system.

```
1  user_agent = 'Mozilla/5.0 (Macintosh; Intel Mac
       OS X 10_11_5) AppleWebKit/601.6.17 (KHTML, like
       Gecko) Version/9.1.1 Safari/601.6.17'
2  set_headers = {
3    'User-Agent': user_agent
4  }
5  with eventlet.Timeout(10):
6    try:
7      r = requests.head(url, headers=set_headers,
           timeout=(3.05,6), allow_redirects=True)
8      if len(r.history) > 0:
9        chain = []
10       status_code = r.history[0].status_code
11       final_url = r.url
12       for resp in r.history:
13         chain.append(resp.url)
14       r.close()
15       return (str(status_code),
             str(len(r.history)), chain, final_url,
             tweet_id)
16     else:
17       r.close()
18       return (str(r.status_code), 0, [],url,
             tweet_id)
19   except:
20     # handle error
```

4.8.5 URL Click Data Lookup System (CDLS)

The core functionality of CDLS is to retrieve URL click data from Bitly's API. We improve on limitations in existing studies by filtering blacklisted URL click data from Bitly by data range and ensuring clicks come from

the domain name: twitter.com. To filter Bitly click data by date range we provide the required date range to the Bitly API URL click data endpoint. To filter Bitly click data by referrer we use the Bitly API metric by referrer endpoint.

```
1 bitly = bitly_api.Connection(access_token)
2 response = bitly.referrers(bitly_url)
3   for r in response:
4     if 'referrer' in r:
5       if r['referrer'] == 'https://t.co/':
6         bitly_clicks = r['clicks']
```

4.8.6 Blacklist Update and Lookup System (BULS)

We use 3 blacklists in our system: GSB, OP and PT. To implement our GSB lookup system, the library *gglsbl* [84] version 1.4.10 is used. This library allows our system to fetch the latest GSB hash prefixes and also perform lookups against the database. The library uses the SQLite [255] database for storing GSB data. The library contains a method *update_hash_prefix_cache()* which is used to update the URL hash prefix database. This method is called every 10 minutes in the fast GSB lookup system and at the beginning of each cycle of the slow GSB lookup system.

During our experiment, we observed that the GSB blacklist typically contains approximately 4.8 million URL hash prefixes of which approximately 3.1 million are unique. Of these, there are approximately 1 million unique URL hash prefixes labelled malware and approximately 1.8 million unique URL hash prefixes labelled social engineering. The remaining URL hash prefixes labels are not used in our study.

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

4.8.7 Tweet History Search System (THSS)

THSS uses the *Tweepy* library to interact with Twitter's Search API. After authorising *Tweepy* to access Twitter, the *Search* method is used to search

for a given URL. This method will return the oldest tweet in Twitter's search history, that contains a given URL string, if it can be found.

```
1 api = tweepy.API(auth)
2 results = api.search(q=query,
    result_type="recent", count=100, until=tomorrow)
```

4.8.8 Twitter URL Shortener Investigation System (TURLSIS)

Our *t.co* URL checking system determines if Twitter blocks tweeted URLs. The system uses Python's *Requests* library [235] to send an HTTP request to each *t.co* URL, along with a Macintosh Safari user agent header setting so the request appears to come from a regular user via the Safari web browser. We discovered that *t.co* only displays warning pages to web browsers – a warning page was not displayed when this user agent setting was disabled. We use Python's *Threading* library to concurrently check 200 *t.co* URLs approximately every 3.5 seconds. This is achieved by setting the *Requests* library's *connect timeout* value to 3.05 seconds and *read timeout* value to 6.05 seconds. A value slightly larger than a multiple of 3 is used since this is the default TCP packet retransmission window [127]. These settings ensure our system runs efficiently and does not get stuck waiting indefinitely for servers to respond. Additional time is incurred when factoring in computations such as database inserts, GSB URL lookups etc – hence the overall average batch completion time of 3.5 seconds for 200 URLs.

Our *t.co* URL checking system takes approximately 7 hours to check all, approximately 1.5 million unique *t.co* URLs, collected during a 24-hour period (1.5 million / 200 [per batch] * 3.5 seconds = 7.3 hours). Of those approximately 1.5 million *t.co* URLs, our system typically drops approximately 400 (0.03%) connections (i.e. *read timeout* errors). It is important to note that many of these timeouts likely occur because the website being tested is offline therefore a larger timeout value would have minimal impact. We decided to process 200 *t.co* URLs per 3.5 second batch by testing various different timings. When more than 200 *t.co* URLs were processed in a 3.5 second period Twitter's servers returned HTTP 503 status code responses (*temporary overload* [129]). After fine tuning and experimenting with different timing values, we concluded that processing 200 *t.co* URLs per 3.5 second batch worked well – typically returning zero to one 503 errors per 1.5 million *t.co* URLs processed. This means that our system is rate limited and therefore not overloading Twitter's servers with requests.

The method *getStatus* will return the response from Twitter for a given *t.co* URL – containing the redirection chain to determine if the URL was

blocked by twitter:

```
def getStatus(url):

  set_headers = {
    ''User-Agent'': ''Mozilla/5.0 (Windows NT
       6.1; WOW64) AppleWebKit/537.21 (KHTML, like
       Gecko) Mwendo/1.1.5 Safari/537.21'',
  }

  r = requests.head(url, headers = set_headers,
     timeout = (3.05, 6.05), allow_redirects =
     True)
  redirection_chain = []
  if len(r.history) > 0:
    for e in r.history:
      redirection_chain.append(e.url)
  return redirection_chain, r
```

4.8.9 Web Browser Testing Suite (WBTS)

Each Testing Machine (TM) is responsible for running one of the three main operating systems: Microsoft Windows 8.1, Apple Mac Yosemite 10.10.1 or Ubuntu 14.04 desktop edition. The software solution VirtualBox was used to provide the Virtual Machine (VM) environment. The VM was running on a Dell Inspiron laptop with 4 GB of RAM and an Intel i3 dual core CPU (2.53 GHz) running under Ubuntu 14.04 64-bit operating system. Unfortunately it was not possible to run the web browsers Chromium or Opera under Windows nor Firefox under Apple Mac. This was due to technical and time constraints on controlling these browsers through the various programming languages.

The operating systems for each TM are described below. These descriptions also include an overview of the custom-built Test Suite Software (TSS) used to carry out the test:

Microsoft Windows

Microsoft Windows version 8.1 was used to carry out all experiments under the Windows operating system. Three browsers were deployed under this test environment: Microsoft's Internet Explorer version 11, Google's Chrome version 44, and Mozilla's Firefox version 40. Browser updates were disabled throughout the testing phase to ensure the same browser version was consistent across all test.

The TSS for automating browser testing on Windows was written in C#. It is one of the main programming languages used by Windows and provides services for interacting with core operating system functions such as executing applications, accessing the clipboard and focusing between application windows. Therefore C# provided an ideal programming environment for carrying out the main tasks of the automated tests required for this experiment.

Apple Mac

Apple Mac Yosemite 10.10.1 was used to carry out all experiments under the Apple operating system. Two browsers were deployed under this test environment: Apple's Safari version 8 and Google's Chrome version 44. Again, Browser updates were disabled throughout testing.

The TSS programming language used under the Mac environment was AppleScript. The decision to use AppleScript was made because it provided an efficient solution to carry out all of the required automation tasks such as focusing windows and accessing the system clipboard. Initially, the programming language objective C was considered due to its native support on Apple's operating systems. But, due to time constraints in developing a short Objective C application, AppleScript was chosen instead.

Linux

Canonical's Ubuntu version 14.04 was used to carry out all experiments under the Linux operating system. Three browsers were deployed under this test environment: Google's Chrome version 44, Chromium version 43, and Mozilla's Firefox version 39. Browser updates were disabled.

The TSS programming language Python was used for the Linux environment because Ubuntu has no equivalent native language such as Microsoft's C# or Apple's Applescript. Instead, the library *gtk.cipboard* was used to access the system clipboard and the package *xdotool* (accessed via system calls) was used to focus application windows and enter keyboard presses.

Monitoring System

The Monitoring System (MS) ran on a Raspberry Pi [284] computer. The Raspberry Pi's general purpose input/output (GPIO) pins were used to connect a breadboard with a multicoloured LED along with appropriate resistors. The TSS then allowed the Raspberry Pi to connect to the remote web server to determine the current status of any tests. The Raspberry Pi could then power the multicoloured LED to display the following three statuses:

- **White:** no test running

- **Green:** test running, no problems detected
- **Red:** problem, test halted

CHAPTER SUMMARY

This chapter described our measurement infrastructure: ***Phish****alytics*; including the design architecture, 4 core systems, and additional shared components.

We described the methodology, design decisions, data sources, experimental set up, and technical implementation for our various measurement studies. We also discussed why the chronological order of our studies differs to the presentation order of our subsequent study chapters.

We described various challenges of conducting longitudinal measurement studies, such as dealing with the ever-changing landscape. For example: the GSB API introduced a rate limit after our *time-of-post* Twitter study, resulting in us changing our methodology for the *time-of-click* study.

Over the next 4 chapters we will present our 4 research studies. We discuss our results in more detail, and addresses various topics raised in previous chapters, in Chapter 9: *Discussion*.

5

Blacklist Analysis Study

An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank

OUTLINE

This chapter presents an edited version of our research paper: BELL, S., AND KOMISARCZUK, P. An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank. In *Proceedings of the Australasian Computer Science Week Multiconference* (2020).

Blacklists play a vital role in protecting internet users against phishing attacks. The effectiveness of blacklists depends on their size, scope, update speed and frequency, and accuracy – among other characteristics. In this study we present a measurement study that analyses 3 key phishing blacklists: Google Safe Browsing (GSB), OpenPhish (OP), and PhishTank (PT). We investigate the uptake, dropout, typical lifetimes, and overlap of URLs in these blacklists.

During our 75-day measurement period we observe that GSB contains, on average, 1.6 million URLs, compared to 12,433 in PT and 3,861 in OP. We see that OP removes a significant proportion of its URLs after 5 and 7 days, with none remaining after 21 days – potentially limiting the blacklist's effectiveness. We observe fewer URLs residing in all 3 blacklists as time-since-blacklisted increases – suggesting that phishing URLs are often short-lived. None of the 3 blacklists enforce a one-time-only URL policy – therefore protecting users against reoffending phishing websites. Across all 3 blacklists, we detect a significant number of URLs that reappear within 1 day of removal – perhaps suggesting premature removal or re-emerging threats. Finally, we discover 11,603 unique URLs residing in both PT and

OP – a 12% overlap. Despite its smaller average size, OP detected over 90% of these overlapping URLs before PT did.

5.1 Introduction

As we explored in Chapter 2: *Background*, phishing attacks lure their victims into revealing sensitive information such as passwords and credit card numbers by spoofing legitimate organisations. Phishing campaigns are a dangerous threat in the cyber world and the number of these attacks continues to grow. Phishing blacklists are a popular defence strategy that aim to protect people from phishing attacks. These blacklists typically contain known phishing URLs, providing an access control list which is used to prevent users from visiting these dangerous websites

For these phishing blacklists to be effective they need to be updated quickly and regularly to protect users from emerging phishing attacks. URLs should be removed from a blacklist when their website is no longer a threat – so as not to impact website visitors once it is safe – but also added again if that same website becomes a threat in the future. The number of URLs contained in a blacklist can contribute to its effectiveness; a small set of niche blacklisted phishing URLs will not provide a user with full protection compared to a large and comprehensive blacklist with a wide net. It is also important to understand the inner-workings of these blacklists as this can help determine how effective they will be at protecting people against phishing attacks.

In this chapter we study 3 blacklists: GSB, PT, and OP, to determine uptake, dropout, typical lifetimes, and any overlap of URLs in these blacklists. Over a 75-day measurement period we regularly retrieve the latest copy of the 3 blacklists and store timestamps for when URLs are added and removed from each blacklist. Using this data we can then calculate various differences between timestamps to carry out our measurements. We discover that, in total, 1,731,452 URLs are aded to GSB; 52,234 to OP; and 48,473 to PT. Throughout our measurement study the average number of URLs contained within each blacklist is: 1,581,351 for GSB; 3,861 for OP; and 12,433 for PT. This shows that GSB is by far the largest blacklist in our study. We also see 17 times more URLs added to GSB than PT and OP combined. The sheer volume of URLs in the GSB blacklist compared to PT and OP combined will likely make GSB a more effective weapon to protect users against phishing attacks.

By measuring URL durations in blacklists we discover that the OP blacklist removes a significant volume of URLs from its dataset after a duration of 5 and 7 days; no URLs remain in OP for more than 21 days. Therefore potentially limiting OP's effectiveness at protecting users from phishing

attacks. We see that, across all 3 blacklists, as time increases, fewer URLs remain in the blacklists. This is because, once blacklisted, phishing URLs are often short-lived.

Through analysing URLs that reappear in blacklists we determine that none of the 3 blacklists enforce a one-time-only URL policy in their dataset; URLs reappear in the blacklists if they continue or re-emerge as a threat. This is good for users because it means they will be protected against reoffending phishing websites. We also show that large numbers of URLs reappear in all 3 blacklists within 1 day of removal – suggesting that these URLs were either removed too soon or that they came back online again.

As a result of comparing the PT and OP blacklists we discover that 11,603 unique URLs reside in both of these blacklists, which is 12% of the total number of URLs added to both blacklists. Despite its smaller average size – seen in the earlier measurement – OP detected over 90% of these overlapping URLs before PT did.

To the best of our knowledge, our study is the first to analyse uptake, dropout, typical lifetimes, and any overlap of URLs in the blacklists: GSB, PT, and OP.

We organise the remainder of this chapter as follows. Section 5.2 explores previous studies related to our work. Section 5.3 provides an overview of our experiments. Section 5.4 provides an overview of our methodology. Section 5.5 presents our key measurement results and interpretations, followed by our conclusion in Section 5.6. The results of our study are discussed in Section 9.2: *Blacklist Analysis Study*.

5.2 Related Literature Summary

In Section 3.3: *Threat Intelligence & Blacklists* we explored a number of existing studies that relate to our study by investigating threat intelligence and blacklists. In Section 3.3: *Guo et al. (2019)*, Guo *et al.* (2019) [162] measured the characterisation of threat intelligence to understand how effective these methods are as defence mechanisms. Their study defined 6 threat intelligence metrics (volume, intersection, unique contribution, latency, coverage, and accuracy) to help determine the effectiveness of a threat intelligence source. In our study we explore some of these metrics as part of our investigation into the 3 blacklists we are studying.

2 further studies that we explored in Section 3.3 [310, 183] analysed blacklist characterisation and overlap – however these studies take a very general look at blacklists as a whole, covering various genres of blacklists. Whereas our study focuses on just the characterisation and overlap of phishing blacklists. Zhang *Et al.* (2013) [310] analysed 9 blacklists from 3 different categories: spam (CBL, BRBL, SpamCop, WPBL, and UCE-

PROTECT), phishing/malware (SURBL, PhishTank, hpHosts), and active attack/probing behaviour (Dshield). Key results show a significant overlap within the same category of blacklists: BRBL and CBL (the 2 largest spam blacklists) cover about 90% of other spam-related lists; hpHosts, PhishTank, and SURBL (phishing/malware blacklists) also significantly overlap. Whereas overlaps between categories are trivial. The results from [310, 183] show that general blacklists often do not overlap because each blacklists focuses on a specific type of attacks (e.g. phishing, malware, spam, etc). The methodology used by [310] groups phishing and malware attacks into the same category. However, there are a number of key differences between these two attacks, therefore grouping them together may affect the results. Also, [310] did not include OpenPhish in their study and their measurement period was limited to 7 days. In this study we want to compare blacklists focus on phishing attacks.

Existing literature, such as [161, 117, 64], describe various blacklist datasets in terms of size, etc. However, existing studies do not specifically investigate and measure the characterisations of the blacklists themselves – since they are usually part of a broader set of research aims. In our study, we will analyse the phishing blacklists: Google Safe Browsing, OpenPhish, and PhishTank to explore uptake, dropout, typical lifetimes, and any overlap of URLs in these blacklists. Our purpose is to help the research community gain a more detailed understanding of these 3 blacklists. To the best of our knowledge, ours is the first study to analyse these 3 blacklists in such a way

5.3 Overview of Experiments

The key experiments we carry out in this study are:

1. Analysis of blacklists: PT, OP, and GSB, to determine number of URLs in each and how their sizes vary over time
2. Measure how long URLs remain in each blacklist for
3. Measure and analyse blacklisted URLs that are removed from then re-added to the same blacklist; timings between reappearance
4. Comparison of URLs between blacklists and detection times of overlapping URLs

5.4 Methodology

Our core methodology involves regularly retrieving the latest copy of the 3 blacklists and storing each URL that is added or removed along with the timestamp of when this occurred. To calculate the total number of URLs

in each blacklist we count the total number of entries in each JSON file for PT and OP and the total number of rows in the SQLite Database for GSB. We also remove duplicates to determine how many of these URLs – or hash prefixes in the case of GSB – are unique. To count the total number of domain names in each blacklist we extract the domain from all URLs in the blacklists then group and total these.

We use the aforementioned URL added/removed timestamps to calculate the duration each URL remained in the blacklists for. URLs which have not been removed from a blacklist currently still reside in that blacklist, therefore we use the current timestamp – at time of measurement – to calculate duration in blacklist. Using this, we show the total number of URLs in each blacklist that did not have a removal timestamp and are therefore still in the blacklist. Since each blacklist only contains a list of URLs – not a list of URLs to be removed – we set the removal timestamp for a given URL to when our system sees that a previously added URL no longer appears in the blacklist.

Our local copy of GSB is stored as a database of SHA-256 URL hash prefixes; the majority of the hash prefixes are 4 bytes (2^{32} bits). Due to these short URL hash prefixes there is likely to be an increase in the number of collisions as the size of the dataset grows. The average number of collisions in k samples, each a random choice among n possible values is: $N(n,k) = k(k-1)/2n$. In our dataset of 1,731,452 SHA-256 URL hash prefixes there will be approximately 349 collisions. Therefore our GSB measurement calculations are accurate to within 0.02%.

To measure URL reappearance in blacklists, we analyse all URLs in each blacklist that have been added more than once. We calculate the duration of time a reappearing URL was included in the blacklist and the duration of time that URL was excluded from the blacklist – we repeat this for the number of times a URL was added to a blacklist. If, on the final inclusion timestamp for a reappearing URL in a blacklist, there is no removal timestamp then we assume the URL is still in the blacklist (as before).

5.5 Results

5.5.1 Overview of Blacklists

This section analyses the number of URLs added to and removed from each of the 3 blacklist: PT, OP, and GSB, during our 75-day measurement study from March to June 2019. Table 5.1 provides an overview of the 3 blacklists, showing total number of unique URLs added, removed, not removed, and added and removed once, in each blacklist during our measurement experiments from March to June 2019. The number of URLs not removed

	PT		OP		GSB
	URLs	Domains	URLs	Domains	URLs*
Added	48,473	20,458	52,234	14,721	1,731,452
Removed	33,245	15,327	46,866	13,315	633,321
Not removed	15,228	6,729	5,368	1,774	1,098,131
Added and removed once	30,967	14,409	43,103	12,147	113,530

Table 5.1: Overview of blacklists: PT, OP, and GSB showing total number of unique URLs and domains added, removed, not removed, and remaining in each blacklist. Measured between March and June 2019. *Number of SHA-256 URL hash prefixes for GSB.

shows how many URLs were added to each blacklist but which were not removed during our measurement – therefore we conclude that these URLs remain in the blacklist at time of measurement. These results show that GSB is a considerably larger blacklist; with 33 times as many URLs added to GSB compared to OP and over 17 times as many URLs added to GSB compared to PT and OP combined. The increased size of GSB, compared to PT and OP, suggests that it may detect more URLs and therefore be more effective at detecting phishing websites compared to PT and OP.

When comparing the number of domains to the number of URLs for PT and OP, in Table 5.1, we see that there are at least twice as many URLs compared to domains for each data set. This suggests that each domain has about 2-3 blacklisted URLs. However, the average number of URLs per domain name is 1. The 10 most frequent domain names added to PT consist of 4,566 URLs – 1% of the dataset. The 3 most frequent domain names in PT consist of 1,459; 829; and 388 URLs, respectively. The 10 most frequent domain names added to OP consist of 8,412 URLs – 16% of the dataset. The 3 most frequent domain names in OP consist of 3,487; 1,965; and 6,55 URLS, respectively. This shows that the majority of domain names appear in both the OP and PT blacklists only once but that a small number of domain names contain multiple different blacklsited URLs. The most frequent domain names in both blacklists appear significantly more times than any other domain in the dataset.

The 3 box plots in Figure 5.1 show the number of URLs in each of the 3 blacklists: PT, OP, and GSB, on each update. PT and OP were updated once per hour and GSB was updated every 5 minutes. All 3 blacklists were updated 24/7 throughout the measurement experiment which ran from March to June 2019. These box plots show that the number of URLs in the PT blacklist ranged from 9,313 to 15,500 with a median of 12,433;

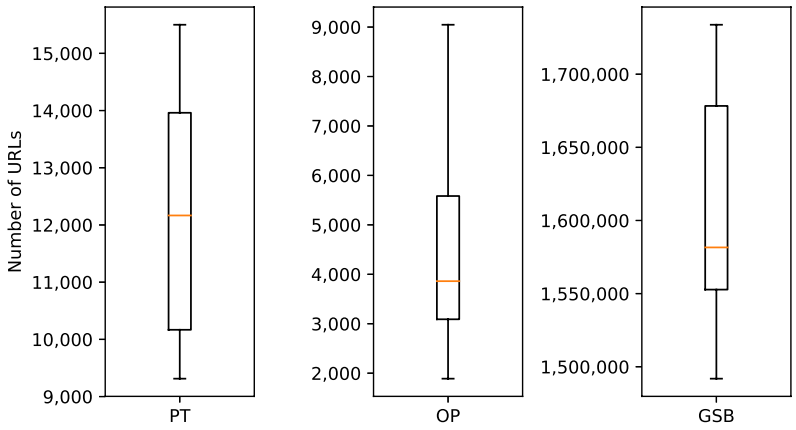


Figure 5.1: Box plots showing number of URLs in blacklists: PT, OP, GSB, at each update. Measured between March and June 2019.

the lower quartile was 10,174 and the upper quartile was 14,041. The OP blacklist saw a range of 1,889 to 9,047 URLs with a median of 3,861; the lower quartile was 3,096 and the upper quartile was 5,748. Finally, the number of URLs in GSB ranged from 1,491,850 to 1,733,813 with a median of 1,581,351; the lower quartiles was 1,551,780 and the upper quartile was 1,677,262. These figures show that the number of URLs in all 3 blacklists stayed within reasonably consistent ranges throughout the measurement study. The range in number of URLs in these blacklists varied by 6,187 in PT; 7,158 in OP; and 241,963 in GSB. This range difference compared to the median number of URLs for each blacklist is PT: 50%, OP: 185%, and GSB: 15%. This shows that the range of URLs in the OP blacklist, during our measurement study, was considerably greater then the average number of URLs we saw in the blacklist. This is likely due to large numbers of URLs that are removed from the OP blacklist, therefore keeping its average size down, and may suggest that URL durations in OP are relatively short. In GSB, the range in number of URLs was relatively small compared to the average number of URLs – suggesting that URLs may remain in the blacklist for some time.

Overall, the findings in Table 5.1 and Figure 5.1 show that GSB is by far the largest of the 3 backlists, in terms of number of URLs added, and also sees the greatest number of URLs that remain in the blacklist throughout our measurement study. We see that the median number of URLs contained within GSB, throughout our measurement period, is just over 1.5 million. In comparison, the median number of URLs in PT was 12,433 and in OP was 3,861 – both over 99% less than GSB's median. Also, 97% fewer URLs were added to both PT and OP than GSB – this further illustrates the scale

Category	URLs
Threat Type	
Threat type unspecified	0
Malware	362,230
Social engineering	8,718,240
Unwanted software	599,546
Potentially harmful application	37,790
Platform Type	
Platform type unspecified	0
Windows	1,609,928
Linux	1,609,928
Android	20,451
OSX	1,609,928
iOS	37,790
Any platform	1,609,927
All platforms	1,609,927
Chrome	1,609,927
Threat Entry Type	
Threat entry type unspecified	0
URL	9,717,803
Executable	0

Table 5.2: Overview of GSB blacklists showing total number of unique SHA-256 URL hash prefixes in each category.

of GSB and how many URLs are added. Interestingly, during our study, the OP blacklist saw 3,761 more URLs added to it than PT. However, the median number of URLs residing in OP was 8,572 less than PT's median – a 69% decrease. This shows that even though 3,761 more URLs were added to OP during our study, the average number of URLs in OP remains low – possibly due to more frequent cleansing of the OP dataset. Due to the higher number of URLs residing in GSB compared to OP and PT, it is likely that GSB would catch a greater number of phishing URLs when deployed to check a random feed of URLs. Therefore GSB may be more effective than PT and OP at protecting users from phishing attacks due to its greater size.

5.5.2 GSB Categories

This section analyses the total number of URLs in each category of the GSB blacklist. Table 5.2 provides an overview of the GSB blacklist, showing total number of unique SHA-256 URL hash prefixes in each category. The three main categories within GSB are: ***threat type***, ***platform type***, and ***threat***

Category	URLs
Malware	362,230
Windows	57,223
Linux	57,223
Android	0
OSX	57,223
iOS	18,895
Any platform	57,222
All platforms	57,222
Chrome	57,222
Social engineering	8,718,240
Windows	1,453,040
Linux	1,453,040
Android	0
OSX	1,453,040
iOS	0
Any platform	1,453,040
All platforms	1,453,040
Chrome	1,453,040
Unwanted software	599,546
Windows	99,665
Linux	99,665
Android	1,556
OSX	99,665
iOS	0
Any platform	99,665
All platforms	99,665
Chrome	99,665
Potentially harmful application	37,790
Windows	0
Linux	0
Android	18,895
OSX	0
iOS	18,895
Any platform	0
All platforms	0
Chrome	0

Table 5.3: Overview of GSB blacklist showing total number of unique SHA-256 URL hash prefixes in categories: *Threat Type* and *Platform Type*, combined.

entry type. The subcategories of ***threat type*** are: *threat type unspecified*, *malware*, *social engineering*, *unwanted software*, and *potentially harmful software*. The subcategories of ***platform type*** are: *Windows*, *Linux*, *Android*, *OSX*, *iOS*, *any platform*, *all platforms*, and *Chrome*. The subcategories of ***threat entry type*** are: *threat entry type unspecified*, *URL*, and *executable*. In Table 5.3, categories ***threat type*** and ***platform type*** are combined to show the total number of URLs in both.

Tables 5.2 and 5.3 show that, of the 4 main threat type categories, *social*

engineering contains the greatest number of total URLs at 8,718,240. Of this total, 1,453,040 URLs are unique; this number remains consistent across all platform types within the *social engineering* category. This is because phishing attacks are not software or platform specific; they rely on human presence for the attack be effective. There are 599,546 total URLs categorised as *unwanted software*, of which 99,665 unique; URLs are consistently shared across all platforms except *Android* which sees 1,556 unique URLs. In the *Malware* threat type category: there are 362,230 total URLs, of which 57,223 unique URLs are on *Windows*, *Linux*, and *OSX* platforms, while *iOS* sees 18,895 unique URLs and *Android* sees 0 URLs. Finally, the *potentially harmful application* category has 37,790 total URLs, of which 18,895 unique URLs are categorised to the *Android* and *iOS* platform, while the remaining platforms see 0 URLs. These figures show that GSB contains more social engineering URLs in its blacklist than other threat type. Suggesting that GSB may be more effective at detecting phishing URLs than other types of threats listed within its categories.

KEY FINDINGS

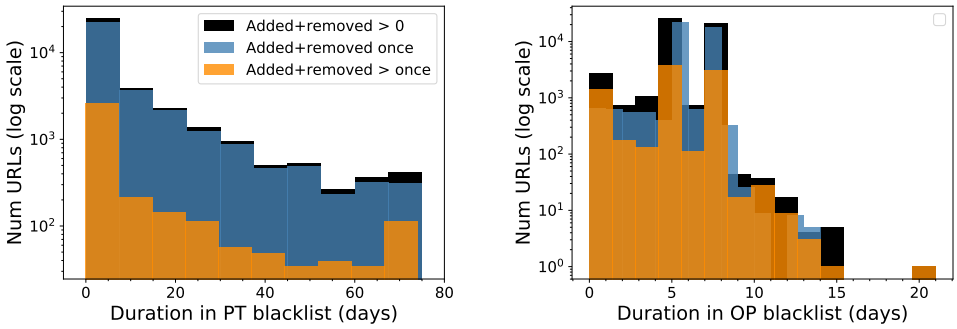
The GSB blacklist contained an average of 1,581,351 URLs, compared to 12,433 in PT and 3,861 in OP. We see 17 times more URLs added to GSB than PT and OP combined.

Social engineering URLs make up the bulk of all URLs in GSB. This makes GSB the largest phishing blacklist in our study; suggesting that GSB should detect a greater number of URLs – therefore making it a more effective blacklist than PT and OP.

The average number of URLs in the OP blacklist is 93% less than than the total number of added URLs to OP – suggesting that OP enforces strict limits on how long URLs remain in its dataset for.

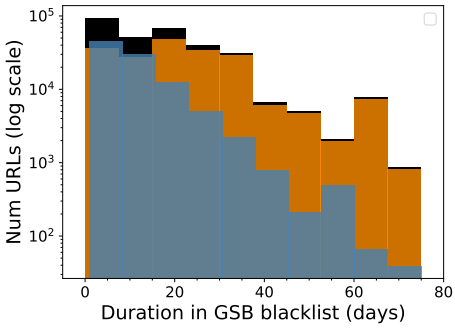
5.5.3 URL Durations in Blacklists

In this section we analyse how long URLs remain in each of the 3 blacklists (PT, OP, GSB) for. Figures 5.2a to 5.2c are histograms showing URL durations in the 3 blacklists, in days, for URLs which were added to and removed from each blacklist at least once, only once, and greater than once. These results were measured between March and June 2019. The y-axes of these histograms are shown on a logarithmic scale to make the results clearer despite a wide variance in range.



(a) PT

(b) OP



(c) GSB

Figure 5.2: Histograms of URL durations (days) in blacklists: PT, OP, GSB, for URLs that are added to and removed from each blacklist at least once, only once, and greater than once. Logarithmic y-axes. Measured between March and June 2019.

In Figure 5.2a we see that the most frequent duration for URLs in the PT blacklist, at just over 10,100 URLs, is between 0 and 5 days. Further analysis of this data revealed a frequency of 14,000 URLs with a duration between 0 and 1 day. We see that about 5,000 URLs remain in the blacklist for between 5 and 15 days. The histogram displays a skewed right pattern; we see the frequency of URLs decrease as the duration in the blacklist increases. There is an increase in URL frequency to the right of the graph at 65+ days – this suggests that PT may possibly cleanse its database at this time therefore resulting in a large numbers of URLs being purged. These durations show that a lot of phishing websites that appear in PT are removed within 24 hours. This may be because these websites are taken offline soon after appearing in the blacklist and therefore no longer pose a threat to users. We see that some URLs remained in the blacklist for the entire duration of our experiment which may suggest that phishing websites stay in the PT

blacklist while they continue to pose an active threat to users. However, the increase in number of URLs removed at a duration of 65+ days in the blacklist is a concern **if** these websites are still actively serving phishing content at time of removal. An analysis of the contents of these websites may provide a clearer picture of why these URLs are removed at 65+ days. We also see that a greater number of URLs are added to the PT blacklist only once. The median duration for URLs in the PT blacklist is 2 days.

The histogram in Figure 5.2b shows URL durations in the OP blacklist. The multimodal pattern in this histogram reveals a spike at the 5 and 7 day durations along with a maximum duration of 21 days. For URLs that are added and removed only once, we see 25,578 URLs with a duration of 5 days and 24,790 URLs with a duration of 7 days – all other duration days see less than 650 URLs each. There are significantly fewer URLs in the blacklist after the 7 day duration and just over 10 URLs between the 10 and 14 day durations. This suggests that the OP blacklist may carry out cleansing of its dataset on URLs which have been in the blacklist for 5 days and 7 days along with a maximum duration of 21 days. It may be that websites which have been taken offline are checked and removed after being in the blacklist for 5 days and that this check is carried out again after 7 days. No URLs remain in OP after 21 days – which may reduce the effectiveness of OP if used to protect users from phishing attacks. Since any phishing websites that stays online for over 21 days may no longer appear in the blacklist, a user may believe that these websites are safe – when they are not. This poses serious security issues about using the OP blacklist to protect people from phishing attacks. Compared to PT, OP has significantly more URLs that are added to the blacklist more than once. The median duration in the OP blacklist is 5 days.

Figure 5.2c shows a histogram of URL durations in the GSB blacklist. We use GSB's Update API to retrieve the latest copy of the blacklist for these experiments. URLs are encrypted as SHA-256 hash prefixes in the local database therefore there are a number of hash collisions in our results. This happens when 2 different URLs share the same hash prefix and would appear in our results as an inaccurate URL duration. For example we saw a number of URLs in our dataset which had negative durations in GSB. This is because the removal timestamp of URL hash prefix has matched a different URL's hash prefix – for a URL that was removed before the original URL was added – therefore showing as a negative duration in the blacklist. We filter our results to only show URLs with positive durations, however, a small number of the results shown are likely to still contain collisions.

Figure 5.2c shows a skewed right pattern, with the greatest frequency of URLs having the shortest duration in GSB. As duration increases, the frequency of URLs decreases. This is likely because a lot of URLs are taken offline soon after they appear in a blacklist. A possible reason for this is that

the URL's hosting provider becomes aware of the blacklisted URL on their server and therefore terminates the related account. Another reason is that blacklisted websites are likely to see a reduction in visitor traffic, due to visitors being unable to access the blacklisted site, therefore attackers may quickly move on and set-up a new website. There is a slight increase in frequency of URLs at the 53 to 60 day duration period, for URLs added once, and 60 to 70 day period for URLs added more than once. This is possibly due to GSB cleansing the dataset at this duration for each URL although we continue to see URLs remain in the blacklist for longer than 60 days – albeit less frequently. As with the PT blacklist, there is no apparent limit on the duration of which URLs remain in GSB other than the potential 53 to 70 day duration cleanse. The median duration in the GSB blacklist is 10 days. A greater quantity of URLs were added to GSB more than once compared to just once. Overall, we see that there is a steady decrease in the frequency of URLs as their durations in the blacklist increase – which is to be expected as blacklisted websites are often taken offline shortly afterwards.

KEY FINDINGS

Across all 3 blacklists: as time increases, fewer URLs remain in the blacklists. This is because, once blacklisted, phishing URLs are often short-lived.

The OP blacklist limits the majority of URLs in its dataset to a duration of either 5 or 7 days; no URLs remained in OP for more than 21 days – therefore potentially limiting the blacklists effectiveness.

5.5.4 URL Reappearance in Blacklists

In this section we investigate URLs that are re-added to a blacklist, at a later date, after having previously been removed. This may happen if an attack website is deemed to be safe and is therefore removed from a blacklist - but then later becomes a threat again so is re-added to the blacklist.

Table 5.4 shows, for each of the 3 blacklists, the number of times each URL was added. This clearly shows how many URLs were added to a each blacklist only once, and how many URLs were added more than once. We see that, for PT, over 95% of all URLs are added to the blacklist only once, just over 3% are added only twice, and 0.01% are added only 3 times. No URLs were added to PT more than 6 times during our measurement study. Similarly, in the OP blacklist, we see 93% of all added URLs are only added once, and 5% are added only twice. We see an increase in the number

Num times added to blacklist	Num URLs		
	PhishTank	OpenPhish	GSB
1	47,127	49,128	620,089
2	1,571	2,604	43,524
3	636	726	488,474
4	9	334	256,836
5	1	148	367,686
6	1	24	159,992
7	0	1	64,382
8	0	1	56,166
9	0	0	1,447
10	0	0	69

Table 5.4: Overview of blacklists: PT, OP, and GSB, showing number of times each URL was added. Measured between March and June 2019.

of URLs added between 3 and 6 times in OP compared to PT; no URLs were added to OP more than 8 times. For the GSB blacklist, we see that the highest frequency – 30% of all added URLs – were only added once. Interestingly, we see a significant drop in the number URLs that were added to the blacklist twice – just 2% of all URLs added to GSB. The number of URLs added 3 times to GSB increases to 24% of all URLs. The reason for this dip in number of URLs added to GSB between 1 and 3 times may be that if a URL is added to the blacklist twice then there is a significantly higher chance that it will continue to reappear. Hence URLs appearing 3 to 8 times are seen more frequent than appearing just twice.

The GSB blacklist is much larger then PT and OP, as a result, we see a significant increase in the number of times URLs were re-added to the to blacklist. To help visualise all of this data, the frequency of URLs that are re-added to the GSB blacklist is represented as a histogram, seen in Figure 5.3. This histogram shows us that the highest frequency of URLs were added to the blacklist between 1 and 6 times. The maximum number of times URLs were added to the blacklist was 18 and we see over 10 URLs were added to the GSB blacklist between 15 and 18 times. Although there will be some URL hash prefix collision within these results, we can still see that GSB allows websites to be re-added to its blacklist multiple times. This may be due to GSB frequently monitoring previously blacklisted websites – that have since been removed from the backlist – to determine if they reoffend. When such websites are found to be hosting malicious content again then they might be re-added to the blacklist

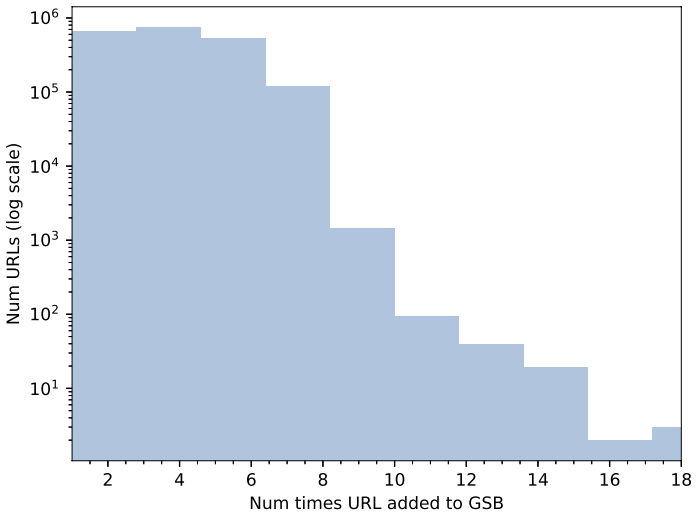
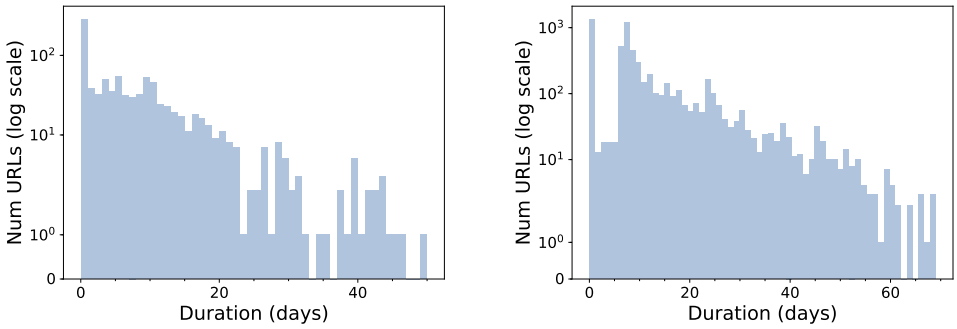


Figure 5.3: Histogram of number of times each URL was added to GSB blacklist. Logarithmic y-axis. Measured between March and June 2019.

Overall, we see that, for all 3 blacklists, once a URL has been removed it can reappear again at a later date. This shows that none of the blacklists enforce a one-time-only URL policy in their dataset and that all 3 blacklists will re-add URLs if they continue or re-emerge as a threat. This is good for users because they will be protected against reoffending phishing websites.

To further explore URLs that are re-added to blacklists, we calculate the duration between removal and reappearance timestamps for all URLs that are re-added to the 3 blacklists. These durations are shown as histograms in Figures 5.4a to 5.4c. We use smaller bin widths in these histograms to produce finer granularity results. Although these reductions in bin widths produce multimodal graphs, we still see a general skewed right multimodal pattern in all 3 of these histograms. This shows that, in all 3 blacklists, fewer URLs reappear as the time since they were removed increases. This is understandable because you would expect the majority of phishing URLs to be taken offline as the duration of time since they were added to a blacklist increases.

We see delays between URL removal and reappearance in the PT blacklist in Figure 5.4a. In this histogram, 285 URLs reappear in the PT blacklist within 1 day; this is the most frequent reappearance delay representing 12% of all reappearing URLs in the blacklist. In comparison, just 38 URLs reappear in the PT blacklist 1 day after being removed. We still see over 20 URLs re-added to PT after more than 30 days since the URLs were originally removed. No URLs reappeared in PT after 50 days. An interesting example of the make-up of a phishing URL in this dataset is:



(a) PT

(b) OP

(c) GSB

Figure 5.4: Histograms of URL durations (days) between URLs being removed and re-added in blacklists: PT, OP, GSB. Logarithmic y-axes. Measured between March and June 2019.

http://www.facebook.com.https.s1.[redacted].com

The domain name *[redacted].com* includes the subdomains *www*, *facebook*, *com*, *https* and *s1*. The results of combining these subdomains is that, to a user, this gives the appearance that the URL leads to *facebook.com* – illustrating a very common masquerading technique used by phishers. We have redacted the actual phishing domain from this example URL for privacy and security. This URL was re-added to PT 46 days after removal.

Figure 5.4b shows delays between URL removal and reappearances in the OP blacklist. 779 URLs are re-added to the blacklist within 1 day of removal and 564 URLs are added 1 day after removal. There is a noticeable drop in the number of URLs re-added to OP between 2 and 5 days after removal. The number of URLs then increases again from 6 days after removal. This relates to the pattern seen in Figure 5.2b (results Section 5.5.3: *URL durations in blacklists*), where there is a peak in number of URLs that remain in OP for 5 and 7 days. This may suggest that OP does not

allow certain URLs to reappear in its dataset, within a certain time period, if they have been previously cleansed.

Delays between URL removal and reappearance in the GSB blacklist are shown in Figure 5.4c. We see that over 3,200 URLs are re-added to the blacklist within 1 day, and 1,000 URLs reappear 1 day after being removed. Interestingly, there is a peak of over 500 URLs that reappear in the blacklist 26 days after removal. This may be where one specific campaign – which had previously been neutralised – later became a threat again and therefore reappeared in the blacklist.

KEY FINDINGS

Overall, we see that none of the 3 blacklists enforce a one-time-only URL policy in their dataset therefore all 3 blacklists re-add URLs if they continue to be, or re-emerge as, a threat. This is good for users because they will be protected against reoffending phishing websites.

We also see that a large number of URLs reappear in the blacklists within 1 day of removal – suggesting that these URLs were either removed too soon or that they came back online again.

5.5.5 Blacklist Overlap

In this section we explore how many URLs reside in both the PT and OP blacklists. We do not analyse URLs that also reside in GSB because URLs are encrypted in the GSB blacklist. In total 11,603 unique URLs – consisting of 6,079 unique domain names – appeared in both the PT and OP blacklists during our measurement study carried out between March and June 2019. The 10 most frequent domain names, appearing in bot PT and OP, consist of 998 URLs; less than 1% of the dataset.

Figure 5.5 is a histogram showing the difference (in days) between PT and OP first detection times for all URLs residing in both blacklists. Positive values indicate that PT detected a URL before OP, negative values indicate OP detected a URL before PT. For example: if a phishing URL appears in PT on April 1 and then in OP on April 30 then the difference in detection times between PT and OP is 30 days. If a phishing URL appears in OP on April 1 and then in PT on April 30 then the difference in detection times between PT and OP is -30 days (i.e., OP detected the URL before PT). The histogram shows that 807 URLs were first detected by PT and that 9,990 URLs were first detected by OP. Both blacklists detect 814 URLs within 1 day. These results show that OP detected 92% more URLs before PT did –

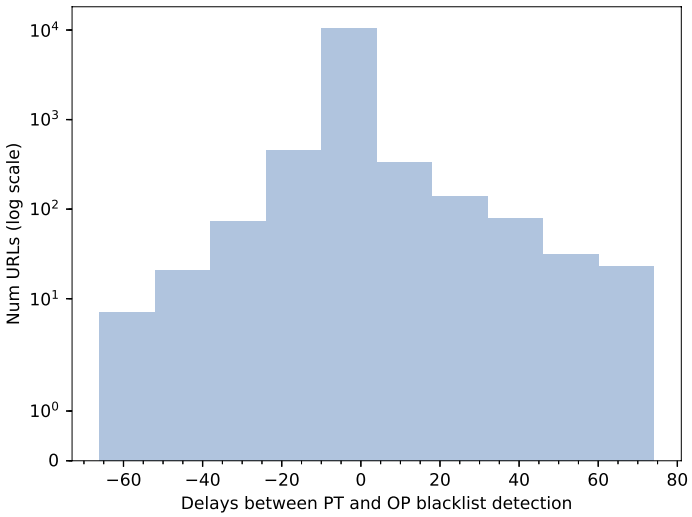


Figure 5.5: Histogram of delays (in days) between PT and OP detecting URLs. Positive values indicate PT detected URLs before OP, negative values indicate OP detected URLs before PT. Logarithmic y-axis. Measured between March and June 2019.

suggesting that OP detects phishing URLs more quickly than OP. However, there are lead and lag times of over 60 days for both blacklists – meaning that both blacklists took at least 2 months to detect some URLs that had already been detected by the other blacklist. Overall, PT saw the greatest number of URLs with delays of over 60 days. Our experiments ran for just over 70 days which defines the upper delay limit for this study.

Figure 5.6 shows the difference (in hours) between PT and OP first detection times for all URLs residing in both blacklists. This histogram shows the first 24 hours of lead and lag times for both PT and OP and is represented on a linear scale for clarity. We see that, in the first 24 hours, 894 URLs were first detected by PT and 9,697 URLs were first detected by OP. Both blacklists detect 1,020 URLs within 1 hour. We cannot increase our measurement granularity any further than 1 hour because both blacklists are updated once per hour. The reason we see more URLs detected in our hourly difference measurements (compared to our daily difference measurements) is because we are performing a frequency analysis of all URL timings during our measurement timeframe; we have finer granularity of the hourly delay timings (compared to daily delay timings). In these results we see that OP detected 91% more URLs before PT did – again, illustrating OP's faster detection times. We see lead and lag times up to the maximum duration, 24 hours, for both blacklists – i.e., both blacklists see up to a 24-hour delay. However, significantly more URLs were first detected

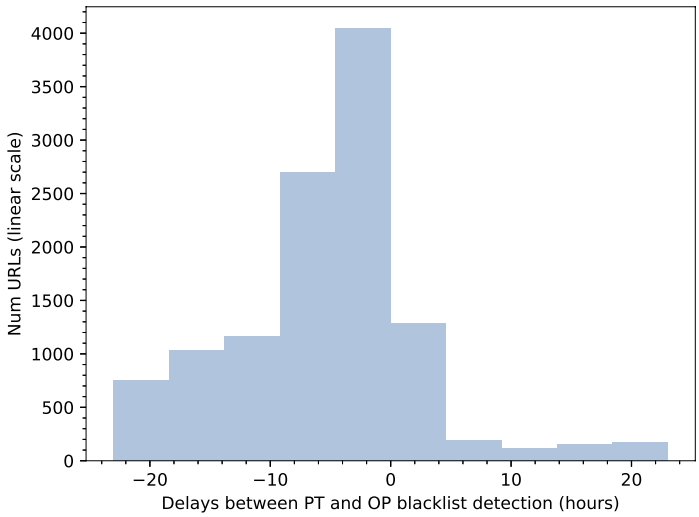


Figure 5.6: Histogram of delays (in hours) between PT and OP detecting URLs – limited to first 24 hours. Positive values indicate PT detected URLs before OP, negative values indicate OP detected URLs before PT. Linear y-axis. Measured between March and June 2019.

by the OP blacklist.

The reason OP detects over 90% of URLs before PT is likely because the PT blacklist is a community based network that relies on people submitting potential phishing URLs. Members of the community then vote whether these submitted potential phishing URLs are genuinely phishing or not. This process of submitting and then manually verifying phishing URLs takes time. Whereas the OP blacklist uses autonomous algorithms to detect zero day phishing websites. Automated detection is faster and does not require multiple people to vote – therefore detection times are reduced.

As previously mentioned, we only check URLs that reside in both PT and OP. In future work we would like to include GSB so we can compare the detection times between all 3 blacklists.

KEY FINDINGS

We see that 11,603 unique URLs reside in both the PT and OP blacklists and that OP was faster at detecting over 90% of these URLs. However, both blacklists have large lead and lag time delays between each other of over 60 days – illustrating how far behind the other blacklist can be.

OP's automated approach to phishing detection likely explains its faster detection rates whereas PT's manual, community-driven verification approach may explain its lag.

5.6 Conclusion

This measurement study analysed 3 key phishing blacklists: Google Safe Browsing (GSB), OpenPhish (OP), and PhishTank (PT). We investigated the uptake, dropout, typical lifetimes, and considered the overlap of URLs in these blacklists. During our 75-day measurement period we observed that GSB contained an average of 1,581,351 URLs, compared to 12,433 in PT and 3,861 in OP. GSB is seen as a ground truth with respect to blacklisting resources and we saw in this study 17 times more URLs added to GSB than PT and OP combined. The sheer volume of URLs in the GSB blacklist could make GSB an effective weapon in the protection of users against phishing attacks through URL blacklisting.

Our measurements revealed that the OP blacklist removed a significant volume of URLs from its dataset after a duration of 5 and 7 days; no URLs remained in OP for more than 21 days. Therefore potentially limiting OP's effectiveness at protecting users from phishing attacks. We saw that, across all 3 blacklists, as time increased, fewer URLs remained blacklisted; phishing URLs are often short-lived. We determined that none of the 3 blacklists enforced a one-time-only URL policy in their dataset; URLs reappeared in the blacklists if they continued or re-emerged as a threat. This is good for users because they will be protected against reoffending phishing websites. We also showed that a significant number of URLs reappear in all 3 blacklists within 1 day of removal – suggesting that these URLs were either removed too soon or that they came back online. Finally, we compared the PT and OP blacklists and discovered that 11,603 unique URLs resided in both of these blacklists – a 12% overlap. Despite its smaller average size, OP detected over 90% of these overlapping URLs before PT did.

6

Time-of-Post Twitter Study

Catch Me (On Time) If You Can: Understanding the Effectiveness of Twitter URL Blacklists

OUTLINE

This chapter presents an edited version of our research paper: BELL, S., PATERSON, K., AND CAVALLARO, L. Catch Me (On Time) If You Can: Understanding the Effectiveness of Twitter URL Blacklists. *arXiv preprint arXiv:1912.02520* (2019).

With more than 500 million daily tweets from over 330 million active users, Twitter constantly attracts malicious users aiming to carry out phishing and malware-related attacks against its user base. It therefore becomes of paramount importance to assess the effectiveness of Twitter's use of blacklists in protecting its users from such threats. We collected more than 182 million public tweets containing URLs from Twitter's Stream API over a 2-month period and compared these URLs against 3 popular phishing, social engineering, and malware blacklists, including Google Safe Browsing (GSB). We focus on the delay period between an attack URL first being tweeted to appearing on a blacklist, as this is the timeframe in which blacklists do not warn users, leaving them vulnerable. Experiments show that, whilst GSB is effective at blocking a number of social engineering and malicious URLs within 6 hours of being tweeted, a significant number of URLs go undetected for at least 20 days. For instance, during one month, we discovered 4,930 tweets containing URLs leading to social engineering websites that had been tweeted to over 131 million Twitter users. We also discovered 1,126 tweets containing 376 blacklisted Bitly URLs that had a combined total of 991,012 clicks, posing serious security and privacy

threats. In addition, an equally large number of URLs contained within public tweets remain in GSB for at least 150 days, raising questions about potential false positives in the blacklist. We also provide evidence to suggest that Twitter may no longer be using GSB to protect its users.

6.1 Introduction

As we saw in Section 2.13, Twitter features a large user base – including politicians, celebrities, societal influencers – which makes Twitter an attractive target for malicious users to carry out phishing and malware attacks. One of the main ways these attacks are carried out is by leading victims to a malicious site, by including one or more URLs in a tweet, whereby the attack can occur.

Phishing attacks on Twitter have been known to lure victims in by offering verification on the social network but instead take them to a fake login page to steal their Twitter username and password [45], while malware attacks have included drive-by-download links contained within tweets, cross-site scripting attacks [21], and Android malware that is controlled by tweets [78].

One of the ways in which Twitter is improving its security for users is by implementing numerous rules [278] that govern what type of content users – and politicians [280, 144] – of the platform can and cannot send. In 2009, it was reported [200] that Twitter had started to use the phishing and malware blacklist Google Safe Browsing (GSB), already used by popular web browsers to filter out and protect its users from attack URLs. We provide evidence that suggests Twitter is not using GSB effectively to protect its users.

Our study aims to assess how effective Twitter's use of blacklists is in protecting its users from phishing and malware attacks. In particular, we focus on the delay period between an attack URL first being tweeted to appearing in one of 3 defined blacklists, as this is the timeframe in which blacklists do not warn users against the attack. We collected over 182 million public tweets containing URLs from Twitter's Stream API over a 2-month period and compared these URLs against 3 popular phishing, social engineering, and malware blacklists that are used in leading web browsers, antivirus solutions, and other online protection technologies.

During one month we discovered 4,930 tweets containing URLs leading to social engineering websites that had been tweeted to over 131 million Twitter users. The majority of URLs contained within these tweets took between 20 and 30 days to appear in GSB. We focus on GSB because it is the main protection used in popular web browsers. In the same month we also discovered 1,126 tweets containing 376 blacklisted Bitly URLs that

had a combined total of 991,012 clicks – these Bitly URLs represent 11% of the total blacklisted social engineering URLs in our dataset for that month. This demonstrates that Twitter users are clicking on and being exposed to dangerous websites.

We also discovered that, while the GSB blacklist is effective at blocking a large number of social engineering and malicious URLs within 6 hours of being tweeted, a large number of URLs go undetected for at least 20 days, with users potentially exposed to attacks during this delay. In addition, an equally large number of URLs contained within public tweets remained in the GSB blacklist for at least 150 days, potentially raising issues with false positives in the blacklist.

Twitter provides a Stream API to access a source of live tweets. There are 3 ways of accessing this API: the filter/sample, decahose, and firehose streams. These feeds contain, approximately, 1%, 10%, and 100% of all public tweets, respectively. The filter/sample feed is free to access, while the decahose and firehose feeds come at a substantial cost. Our study made use of Twitter's filter/sample stream. There are methodological limitations to using this smaller sample feed. For example, URLs of interest may not be contained in the feed we receive. We compensate as much as possible for this, with techniques such as using Twitter's Search API to determine original tweet date instead of relying what our 1% sample tells us.

To the best of our knowledge ours is the first in-depth study that specifically focuses on the impact of blacklist delays on Twitter traffic. Our study provides a present-day snapshot of the current state of phishing and malware URLs being posted to Twitter. A previous study from 2010 [110] took important first steps in this direction, but Twitter's active user base has grown from 30 million users in 2010 to 330 million users in 2017 [259] and the number of daily Tweets has grown from 35 million in 2010 to over 500 million in 2017 [130]. We replicate the experiment of [110] (to the extent we can in the face of missing details in [110]) but also present a more comprehensive and detailed analysis of malicious URLs on Twitter. In particular, we introduce a new methodology to measure delay from first tweet to membership in the GSB blacklist to determine effectiveness of Twitter URL blacklists. We are also able to determine worst-case scenario delay periods, and we measure the duration of time that URLs stay in GSB.

We organise the remainder of this chapter into the following sections: Section 6.2 introduces key related work, Section 6.3 describes our methodology, Section 6.4 provides an overview of our experiments, Section 6.5 presents our results, and Section 6.6 provides concluding remarks. The results of this study are discussed in Section 9.3: *Time-of-Post Twitter Study*.

6.2 Related Literature Summary

As we saw in Section 3.3: *Ludl et al. & (2007) & Sheng et al. (2009)*, Ludl *et al.* and Sheng *et al.*, illustrated that blacklists can be an effective ground truth – but that temporal aspects of blacklists should also be considered. The experiments carried out in this thesis focus on Twitter as a delivery platform for phishing attacks rather than e-mail.

Whilst existing studies have looked at the phishing landscape in terms of detecting and preventing phishing attacks, they have not focused specifically on the relationship between blacklists and phishing and malware attacks on Twitter. However, in a 2010 study, Grier *et al.* [110] characterised phishing, malware and scam URLs posted to Twitter. As part of their broad study, of which their overall aim was to characterise spam on Twitter, they analysed blacklist delays and performance, looking at the blacklists GSB, Joewein, and URIBL. One of their main findings was that malicious URLs either appeared in the GSB blacklist, on average, 29.58 days before being tweeted or, if the URLs were not blacklisted at time of tweeting, it took, on average, 24.9 days for the GSB blacklist to detect the URLs. Phishing URLs either appeared in the GSB blacklist, on average 2.57 days before being tweeted, or, if not in the blacklist at time of tweet, an average of 9.01 days after tweeting. Grier *et al* also produced evidence showing 100,000 phishing and malware URLs were posted to Twitter during a 1 month period. The study provided evidence showing that spam URLs received 1.6 million clicks from Twitter users. In June 2010 Twitter announced [279] that it was working on its own URL shortening service – this service launched a year later in June 2011 [42]. Therefore the 2010 study was conducted before Twitter implemented its URL shortener, *t.co*.

Whilst the Grier *et al.* study [110] looked at the delay for tweeted URLs to appear in a blacklist, it treated multiple tweets of the same URL as being unique, independent events. We take a different approach. We focus on the delay time between when a blacklisted URL is ***first*** tweeted to when it first appears in a blacklist such as GSB. We believe this provides a more accurate measurement to ascertain the effectiveness of Twitter URL blacklisting. This is because it enables us to determine how long users are exposed to a specific attack URL since it was first posted to Twitter. One of the main problems with the methodology in [110] is that a URL may be tweeted at a certain point in time, then tweeted again on multiple occasions at much later dates, closer to the point at which that URL becomes blacklisted. This then skews the results because the average delay time for that URL to become blacklisted, when calculated using all tweet times containing that URL, will appear to be smaller than the time of first tweet to blacklist delay. This will tend to underestimate the exposure of users.

A missing detail from [110] is how the historical blacklist data from

GSB was obtained. Our study uses timestamps of when URL hash prefixes were downloaded into our local copy of GSB to determine when a URL first appeared in the GSB blacklist. Grier *et al.* [110] were also not specific about which version of Twitter's Stream API they use, other than mentioning that it is a 10% feed. It is important to note that a 10% feed in 2010 will have produced approximately 3.5 million tweets per day – similar to the 3 million tweets per day that we collect in our study. The methodology section of our study explains what version of Twitter's Stream API we used, in an effort to improve the reproducibility of our study.

It is important to note that the aim of [110] was to characterise spam on Twitter, looking at phishing, malware, and scams; that study touched on blacklist performance as part of an overall, broad analysis of spam on Twitter. In contrast, our study aims to assess how effective Twitter's use of blacklists are at protecting its users from phishing and malware attacks. In contrast, we carry out a more fine-grained and in-depth study into the effectiveness of blacklists on Twitter, particularly focusing on delay periods. As well as replicating the relevant experiments from [110], we also introduce a new methodology to measure the delay between when a blacklisted URL is ***first*** tweeted to when it first appears in a blacklist. We also add the PhishTank and OpenPhish phishing blacklists to our study. Finally, and importantly, we check redirection chains for each tweeted URL, since blacklisted URLs may be hidden in such chains.

6.3 Methodology

As described in the previous subsection, tweets containing URLs are collected from the Twitter stream and saved into a local database. 3 blacklists are also stored in the local database. In order to determine which tweeted URLs appear in the 3 blacklists two systems are used: fast and slow. The fast system checks the 3 million most recently tweeted URLs (equivalent to about 24 hours of tweets) against the GSB blacklist every 10 minutes and the OpenPhish and PhishTank blacklists every hour. We determined this 10 minute update frequency by carrying out a small-scale study to observe how frequently the GSB Update API blacklist updates. OpenPhish and PhishTank refresh their blacklists every 60 minutes. The slow system checks all tweeted URLs we have collected since our experiment began and performs a lookup on the latest versions of all 3 blacklists. This slow lookup system will complete its cycle of all URLs relatively quickly at first but increase in duration as the number of URLs in the experiment grow. The main slowdown in this lookup system is that the GSB API requires any hash prefix match to be sent to GSB's servers for the full hash to be downloaded then checked for a match. This system is necessarily slower

in its operation, taking a number of hours to complete a pass over our full collection of tweets. As previously mentioned, in this study our access to GSB was not rate limited, therefore we could lookup all URLs in our study for blacklist membership. The reason for these two lookup systems (fast and slow) is because GSB does not include a "time of inclusion" for blacklisted URLs. This system helps us to determine when URLs appear in the GSB blacklist, with finer resolution on URLs that are Tweeted within 24 hours. The outcome is that we can produce more accurate results in our measurements.

During the experiment it was discovered that the library implementation we use for GSB's Update API also stores timestamps for when blacklisted URL hash prefixes were added to the local database. We then built a system to lookup each blacklisted URL's hash prefix timestamp to determine when each URL was added to our local copy of the blacklist. The GSB Update API library stores each blacklisted URL as a 4-byte SHA256 hash prefix; due to the small size of these URL hash prefixes, there is a chance that collisions may occur. Because of this, only hash prefix lookups that had zero collisions were used for the experimental results. This additional system complements the previously mentioned fast and slow lookup systems because the new system will produce more accurate results for when a URL is already in GSB – particularly if a URL has been in GSB for a significant amount of time. The fast and slow systems are still required for when Tweeted URLs are not in GSB at time of tweet.

When calculating the time from a tweet appearing in the Twitter Stream feed to appearing in one of the 3 defined blacklists, some tweeted URLs may have previously appeared on Twitter prior to being received in the Twitter Stream feed. To compensate for this, we carry out another experiment. In this experiment, when computing delays from time of tweet to time of blacklist appearance, we built a system to lookup each blacklisted URL in Twitter's Search API. Our system can determine when the URL was first tweeted; this timestamp can then be used to calculate the delay between first tweet and first blacklist appearance, therefore increasing the accuracy of the measurement. Limitations of using this approach, as stated in Twitter's Search API documentation, are that it is limited to 7-10 days, it is not an exhaustive source of tweet. Therefore not all tweets will be indexed or made available via the search interface.

6.4 Overview of Experiments

Our first experiment analyses tweets collected from Twitter's Stream API with the sample method; these sample tweets are collected during the same time frame as the URL-containing tweets. This experiment shows

us the ratio of URL containing to non-URL containing tweets along with a breakdown of the numbers of tweets received per day.

Our second experiment replicates one of the experiments carried out by Grier *et al.* [110] in which the delay from a URL being tweeted to appearing in the blacklists is calculated. This experiment shows what has changed since the 2010 study – particularly since Twitter's active user base has grown from 30 million in 2010 to over 330 million in 2017 and total number of daily tweets has grown from 35 million in 2010 to over 500 million in 2017.

Our third experiment uses a different methodology to the 2010 study [110], in that we use the timestamp for when a blacklisted URL was ***first*** tweeted to calculate delay to first appearing in a blacklist. If a URL is tweeted multiple times then only the first tweet to contain that URL will be used to calculate delay. This measurement is important as it allows us to determine how long it takes for URLs to appear in blacklists after they are first tweeted. This experiment also includes the PhishTank and OpenPhish databases.

Our fourth experiment is an improvement on the previous experiment in that the Twitter Search API system is used to determine when a URL was first tweeted. Within Twitter's Search API limit, of 180 calls per 15 minute window, URLs that appear in blacklists are searched for on Twitter to determine their original tweet date. This allows us to determine, with more accuracy than the previous experiment, when a URL was first Tweeted (i.e. if we did not receive the original tweet containing a given URL in our Twitter Stream). This also provides us with the worst case scenario measurement.

Our fifth experiment analyses for how long blacklisted tweeted URLs remain in the GSB blacklist for. In order to carry out this experiment the timestamp for when a URL first appears in the Twitter Filter (URL) Stream is compared against the last time the system matched the same URL in the GSB blacklist. The difference between these two timestamps is used as the measurement.

6.5 Results

6.5.1 Twitter Dataset Analysis

This first results subsection provides an overview of the dataset obtained from collecting tweets via the Twitter Stream API *Sample* method during October and November 2017. We collect, approximately, 3.4M sample tweets and 3M URL-containing tweets per-day. Overall, the Twitter Stream Sample collected 105,306,234 tweets in October 2017 and 100,817,746 tweets in November 2017; of these, only 23% contained URLs. The Twitter Stream Filter (URL) collected 91,871,659 tweets in October 2017 and

	October	November
Twitter Sample	105,306,234	100,817,746
URL	24,085,266	23,478,257
Non-URL	81,220,968	77,339,489
Twitter Filter (URL)	91,871,659	90,719,779

Table 6.1: Total number of collected Twitter sample and Twitter filter (URL) stream Tweets, October and November 2017.

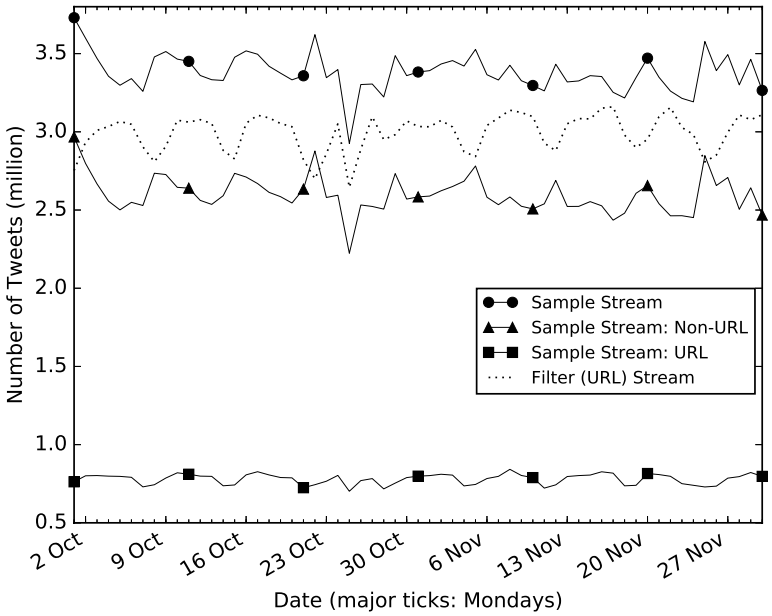


Figure 6.1: Total Tweets collected per day: sample stream & filter (URL) stream API, October and November 2017.

90,719,779 in November 2017, as shown in Table 6.1. Figure 6.1 shows the per-day total number of Twitter Sample Stream tweets, including URL and non-URL containing tweets, along with total number of Twitter Filter (URL) Stream tweets collected in October and November 2017.

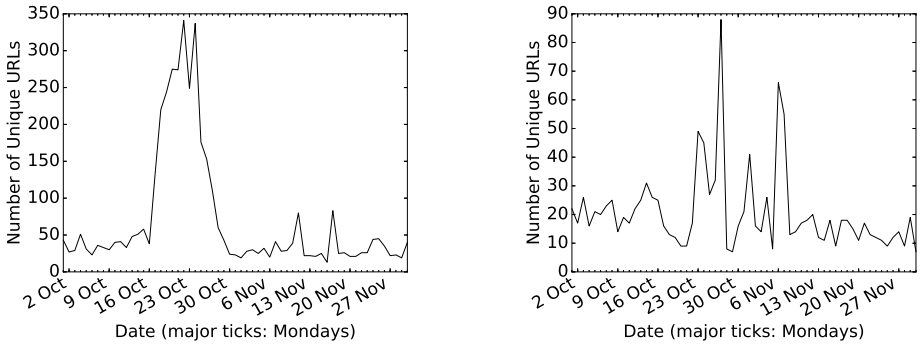
There are 10,029 unique URLs that first appeared in the Twitter Stream Filter (URL) in either October or November that subsequently appeared in one of the GSB, OpenPhish or PhishTank blacklists at some point during our experiments. Of these URLs, 5,464 appeared in one of the blacklists within 1 month before or after first appearing in the Twitter Filter (URL) Stream, as

	Oct '17		Nov '17	
Blacklist	**URLs**	**Domains**	**URLs**	**Domains**
GSB SE*	4,912	397	2,495	268
GSB SE†	3,273	212	930	182
GSB SE§	295	89	294	73
GSB Malware*	1,563	250	1,054	144
GSB Malware†	718	82	543	65
GSB Malware§	230	37	131	29
OpenPhish*	2	2	1	1
OpenPhish†	1	1	1	1
PhishTank*	2	2	4	3
PhishTank†	1	1	4	3

Table 6.2: Number of unique, blacklisted social engineering (SE) and malware URLs & domains first tweeted in October and November 2017.
*Blacklisted anytime during experiment.
†Blacklisted within 1 month from first tweet date.
§Blacklisted within 1 month from first tweet date and using Twitter's Search API to determine URL first tweet date.

seen in Table 6.2. It is interesting to note that only 9 URLs from OpenPhish and PhishTank appeared in the Twitter Filter (URL) stream during the October and November timeframe. In October, of the 2 OpenPhish URLs that were tweeted, 1 had been added to the OpenPhish blacklist on the 22nd August 2017 and the other had a delay of 12 days from date first tweeted to appearing in the blacklist. Of the 2 PhishTank URLs from October: one had been tweeted on the 15th October 2017, but was blacklisted by OpenPhish on 1st September 2017, the other was blacklisted approximately 5 minutes after being tweeted. For November: the 1 OpenPhish URL appeared in the blacklist approximately 5 minutes after being tweeted. For the 4 PhishTank URLs, blacklist delays were approximately 32 minutes, 35 minutes, 21 days and 9 days after tweet. Considerably fewer tweeted URLs appeared in the OpenPhish and PhishTank blacklists compared to GSB. One reason for this difference may be that the GSB blacklist contains approximately 3 million URLs whereas the PhishTank and OpenPhish blacklists contain 28,000 URLs combined; there are fewer URLs for PhishTank and OpenPhish to detect. Another possibility is that Twitter is using the PhishTank and OpenPhish blacklists and therefore preventing users from tweeting URLs contained within these blacklists. However, if that were the case, then we would still see URLs in the Twitter Stream before they appear in the OpenPhish or PhishTank blacklists.

Figures 6.2a and 6.2b show the total number of unique URLs per day



(a) Social Engineering, Oct & Nov 2017 (b) Malware, Oct & Nov 2017

Figure 6.2: Total unique first tweeted social engineering & malware URLs per day that first appeared in GSB blacklist within 1 month before or after Tweet in October & November 2017.

that first appeared in the Twitter Filter (URL) Stream in the given month that subsequently first appeared in the GSB Blacklist, as either social engineering or malware, within 1 month before or after appearing in the Twitter Filter (URL) Stream for October and November 2017.

KEY FINDINGS

These results show that we collected, approximately, 3.4M sample and 3M URL-containing tweets per day throughout October and November 2017. Of these, 5,464 unique URLs appeared in one of the 3 blacklists within 1 month before or after first appearing in the Twitter Stream Filter (URL). This volume of tweets provides us with a good amount of data to explore delay times, click metrics, and overall time in GSB in the upcoming sections.

We also see there are only 9 URLs from the OpenPhish and PhishTank blacklists. This may possibly be because the PhishTank and OpenPhish blacklists contain fewer URLs (approximately 28,000) compared to GSB (approximately 3 million).

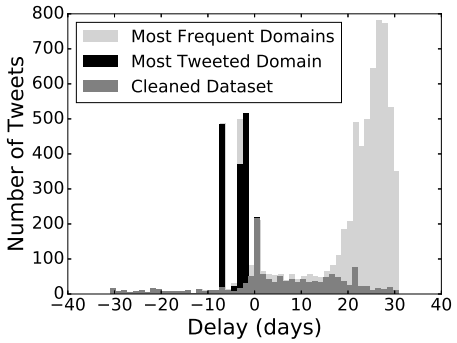
6.5.2 Blacklist Delays – All Blacklisted Tweets

In this subsection we replicate one of the experiments carried out in 2010 by Grier *et al.* [110]. It is important to note that it is difficult to replicate their study exactly because their methodology is not completely explained

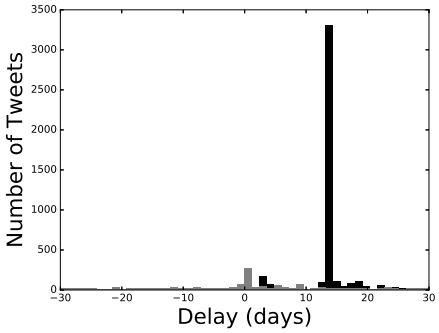
in their paper. Specifically, they do not explain how a historical copy of the GSB blacklist is acquired or if they allow a delay period of 1 month before and after every URL is tweeted. To the best of our knowledge, based on their paper, this is a replication of one of the experiments in their study.

In this experiment the delay period for a tweeted URL to appear in the GSB blacklist is calculated using time of tweet to time first appearing in blacklist. If a URL is tweeted multiple times then each posting is treated as a unique, independent event. This is the same methodology used by [110]. In our results a negative delay value represents a URL that appears in the blacklist before it is tweeted and a positive delay value represents a URL that appears in the blacklist after being tweeted. This is because we are measuring the delay from a URL being tweeted to first appearing in a blacklist, so a delay value of 20 days means it took 20 days from that URL being tweeted to appearing in a blacklist. The Grier *et al*. [110] study uses lead and lag times in their measurements, where a lead time signifies a URL that appears on Twitter before being blacklisted and a lag time is used to denote a URL that appears in a blacklist after being tweeted. As a result, their lead times are positive and lag times are negative values.

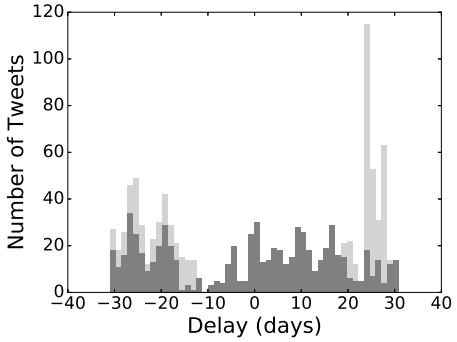
Our first experiment looks at URLs that were tweeted during October and November 2017 which were subsequently labelled as social engineering in the GSB blacklist within 1 month before or after being tweeted. We believe this is the most accurate way to carry out this measurement since the same timeframe is applied to all individual tweets, regardless of when they were tweeted in the month. This methodology is not defined in [110] so it may affect the comparison. Timestamps for when tweets are received from the Twitter Filter (URL) Stream are used as tweet date and URL hash prefix timestamps from the GSB blacklist library are used to determine time first appeared in GSB blacklist to calculate total delay from tweet to blacklist, as described in the methodology section of our study. During this experiment a total of 7,597 tweets containing social engineering URLs in the GSB blacklist were recorded in October and 5,193 in November, as seen in Figures 6.3a and 6.3b. We then carry out the same experiment for malware URLs: a total of 1,110 tweets containing malware URLs embedded the GSB blacklist were recorded in October and 914 in November, as seen in Figures 6.3c and 6.3d. An additional step we take in our experiments, which was not carried out in [110], is to further investigate anomalies in these results and to also clean the results by removing the most frequent domain names. This is explained further in the next subsection. For tweets containing blacklisted social engineering URLs In October there is a spike of tweets at -8 days and again between -2 and -4 days. These spikes are caused by one domain name. For tweets containing blacklisted social engineering URLs in November, there is a peak of 3,316 tweets that have a delay time of between 13 and 14 days, as seen in Figure 6.3b. This spike is caused by



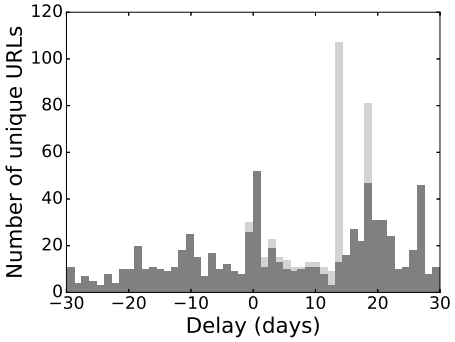
(a) Social Engineering Oct 2017



(b) Social Engineering Nov 2017



(c) Malware Oct 2017



(d) Malware Nov 2017

Figure 6.3: Delay time for all tweets containing GSB blacklisted URLs (including most frequent domain names) labelled social engineering and malware, November and October 2017.

	GSB SE		GSB Malware	
	Oct	Nov	Oct	Nov
Avg. lag (days)	22.62	12.71	17.99	15.02
Avg. lag – top domains removed (days)	11.05		13.37	15.55
Avg. lead (days)	-5.39	-12.18	-19.41	-12.57
Avg. lead – top domains removed (days)	-5.46		-18.24	-12.73

Table 6.3: Average delay times for all tweeted blacklisted social engineering (SE) and malware URLs. Lead and lag times indicate appearing in blacklist before or after being tweeted, respectively.

one domain name.

When comparing our results to the 2010 study [110] it is important to remember that their study had access to a 10% Twitter feed of approximately

	Google Phishing	Google Malware
Avg. lead period (days)	2.57	29.58
Avg. lag period (days)	-9.01	-24.9

Table 6.4: For comparison: Grier *et al* (2010) [110] results: blacklist performance, measured by the number of tweets posted that lead or lag detection. Positive numbers indicate lead, negative numbers indicate lag.

35 million tweets per day in 2010; our 2017 study collects approximately 3 million URL-containing tweets per day – comparable numbers. The first noticeable difference is that there are a greater number of overall tweets containing blacklisted social engineering URLs in our study (Figures 6.3a and 6.3b). Whereas [110] sees a greater number of tweets containing malware URLs appear in GSB after they have been tweeted. We see significantly more tweets containing blacklisted URLs appearing on Twitter after they have appeared in the GSB blacklist for both social engineering and malware URLs. In our study, for social engineering tweets in November, the delay with the greatest number of tweets is 13.5 days with approximately 3,275 tweets. In [110], the delay with the greatest number of tweets approximately -6 days with approximately 58 tweets. In our study, when looking at social engineering tweets in October, the delay with the greatest number of tweets is 26 days with 790 tweets. This shows that, in our results, there is a greater volume of social engineering tweets appearing on Twitter. The results in [110] show that the average lag time for social engineering tweets is 9.01 days and the average lead period is -2.57 days. For malware tweets the average lag time is 24.90 days and the average lead time is -29.58 days. The results for the average lead and lag times for our experiments can be seen in Table 6.3; [110] can be seen in Table 6.4. These figures show that the lag time averages can vary depending on if the most frequent domain names are included in the calculation, as is the case for social engineering and malware tweets in October.

KEY FINDINGS

One of the most significant differences between our results and those in 2010 [110] is that, in our results, there are substantially more URLs being posted to Twitter ***after*** they appear in the GSB blacklist, compared to [110].

This suggests that Twitter has altered its filtering process to allow some URLs blacklisted by GSB to be tweeted or they may have stopped using the GSB blacklist altogether and built their own URL filtering system.

6.5.3 Blacklist Delays – From Time of First Tweet

In this subsection we use a different methodology to [110] in that the timestamp for when a blacklisted URL was ***first*** tweeted is used to calculate delay to first appearing in a blacklist. This new methodology is important as it allows us to determine how long it takes for URLs to appear in blacklists after they are first tweeted – therefore calculating how long users are exposed to attacks for. If a URL is tweeted multiple times then only the first tweet to contain that URL will be used to calculate delay. One of the main problems with the measurement carried out in [110] is that a URL may be tweeted at a certain point in time, then tweeted again on multiple occasions at a much later point in time; closer to the point at which that URL becomes blacklisted. This then skews the results because, in this example, the average delay time for that URL to become blacklisted, when calculated from all tweet times containing that URL, will be less when compared to just the time of first tweet to blacklist delay.

In this experiment we look at unique URLs that were first tweeted during October and November 2017 which were subsequently labelled as social engineering in the GSB blacklist within 1 month before or after being tweeted. Timestamps for when tweets were received from the Twitter Filter (URL) Stream are used as the tweet date and URL hash prefix timestamps from the GSB blacklist library are used to determine time first appeared in GSB blacklist to calculate total delay from tweet to blacklist, as described in the methodology section. During this experiment a total of 3,273 unique social engineering URLs were recorded in October and 930 in November.

During October the majority of social engineering URLs saw a delay period of approximately 18 to 26 days from being tweeted to appearing in the GSB blacklist. Upon further investigation it was discovered that 7 domain names accounted for 76% of the total dataset, 2,487 URLs, as shown in Table 6.5. All of the URLs contained within this dataset are *HTTP*;

Domain	TLD	Number of URLs
1	.cn	614
2	.cn	582
3	.com	554
4	.com	273
5	.cn	203
6	.cn	188
7	.life	73

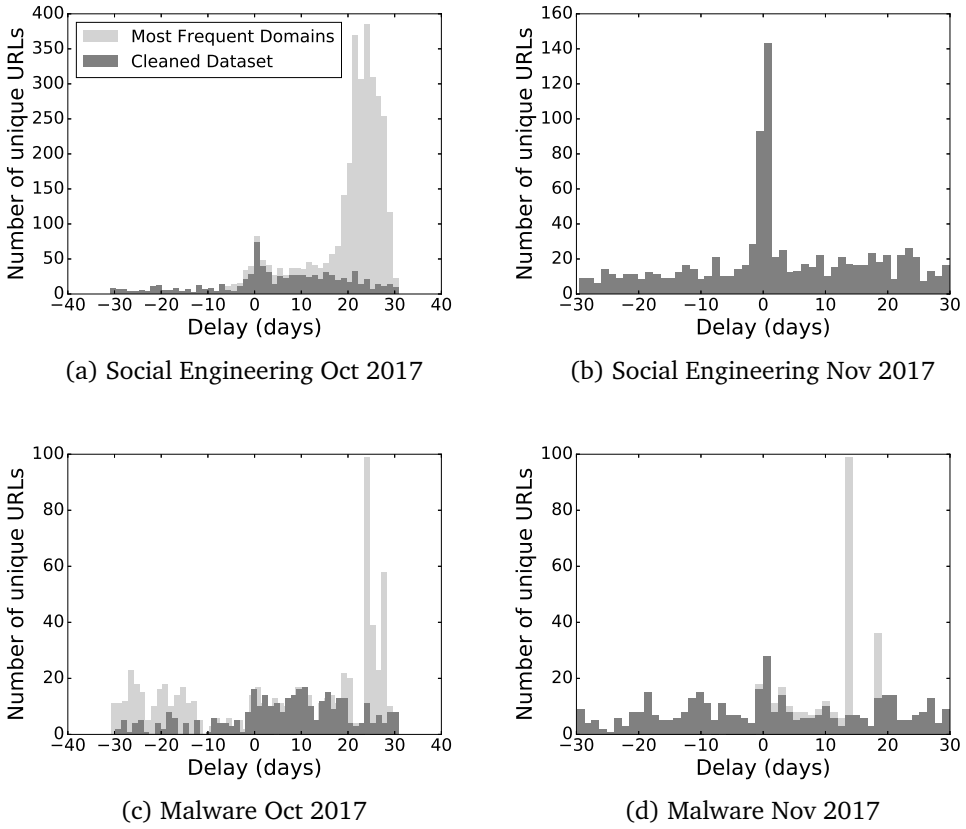
Table 6.5: October 2017 seven most frequent social engineering domains tweeted (domain names redacted).

none of them are *HTTPS*. We extract the domain name for each URL, as per these examples:

- *http://**example.com**/some-web-page.html*
- *http://subdomain.**example.com***
- *https://**example.com**/some-secure-page.html*

Figure 6.4a shows frequency distribution for both the original 3,273 URLs along with the remaining 426 URLs after the top 7 domains names have been removed. This histogram, with the top 7 URLs removed, shows that the majority of URLs appeared in the GSB blacklist within 6 hours of being tweeted, as seen, after being zoomed in to 24 hours, in Figure 6.5a. A similar pattern is also seen in November where there is a peak at around 6 hours, as seen in Figure 6.5b, although still a high number of URLs are blacklisted between 6 and 24 hours. Figures 6.4a-6.4d and 6.5a-6.5b show the delay period between tweet and blacklist, with the number of unique URLs on the *y* axis and delay period along the *x* axis. A delay period greater than zero means that the URL appeared in the GSB blacklist after it appeared on Twitter. A delay of less than zero means that it was already in the GSB blacklist at time of Tweet. The negative delay values, in these graphs, show that large numbers of URLs were tweeted ***after*** they appeared in the GSB blacklist. As with the previous subsection, this further suggests that Twitter are either not using the GSB blacklist or are allowing some URLs in GSB to be tweeted. This means that Twitter users are exposed to social engineering and malware attacks.

In terms of the impact of these tweets, looking at just the top 7 most frequent domains that were first tweeted in October 2017 and appeared in GSB within 1 month before or after being tweeted, these 2,487 unique URLs were tweeted by 1,227 individual Twitter accounts, making up 4,930



(a) Social Engineering Oct 2017

(b) Social Engineering Nov 2017

(c) Malware Oct 2017

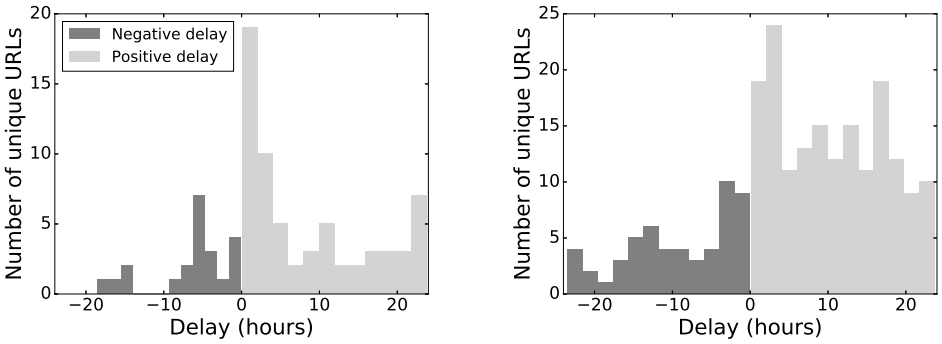
(d) Malware Nov 2017

Figure 6.4: Delay from time of URL first tweet to appearing in GSB blacklist (including most frequent domain names) labelled social engineering and malware, November and October 2017.

total tweets. These 1,227 Twitter accounts have a combined number of 131,116,820 followers giving a sense of the total number of Twitter users potentially exposed to these social engineering tweets.

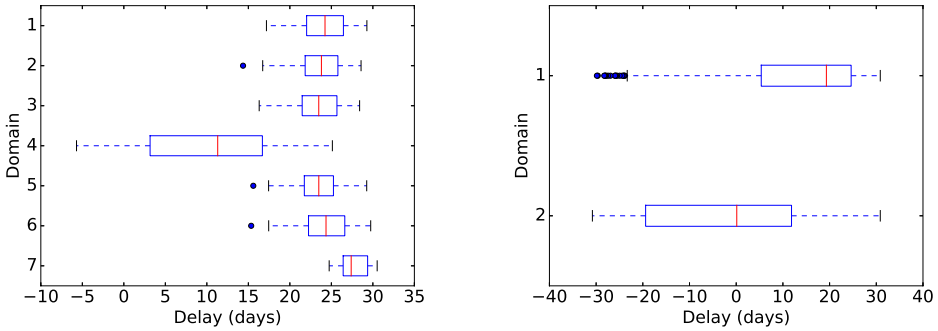
Figure 6.6a shows the distribution of these top 7 domains names in the dataset showing that Domains 1, 2, 3, 5 and 6 appear predominantly towards the 16 to 30 day delay mark with a few outliers around the 14 to 16 day mark, the majority of Domain 4 spans from -5 to 26 days and Domain 7 stays around the 25 to 31 day mark.

When comparing the most frequently tweeted domains that are flagged as social engineering in GSB in October and November there are 4 domains names that appear in both months. This shows that, during the two months in which we collected data from Twitter and GSB, there were a number of large campaigns that spanned across both of these months. One of these domain names is in the Alexa top 100, suggesting that this website had become compromised, potentially by some sort of social engineering advert.



(a) Social Engineering Oct 2017 (b) Social Engineering Nov 2017

Figure 6.5: Social engineering URLs: delay from tweet to first appearing in GSB blacklist (Figures 6.4a and 6.4b), first 24 hours, October & November 2017.



(a) Phishing Oct 2017 (b) Malware Oct 2017

Figure 6.6: Box plots showing most frequent social engineering & malware domains for October 2017.

One theory as to why Twitter continues to allow URLs from this domain, and others like it, to be tweeted on its network is because the main web browsers (such as Chrome, Safari, Firefox etc.) have built-in protection – which should prevent users from visiting dangerous websites. Twitter can then outsource the protection of its users to the web browsers. This is also the case on both the Android and iOS Twitter apps whereby links are scanned by the Chrome and Safari web browser blacklists. One of the main weaknesses to this approach is that there may be an attack space when web browsers update their blacklists. If a user visits a newly blacklisted website, but their web browser has not updated their local copy of the blacklist, then the user will be allowed to visit the dangerous website without any warnings – exposing them to the attack.

When analysing the target of the social engineering campaign tweets in October many of the tweets appear to be using click-bait techniques. These tweets often use misleading titles to promote, for example, health techniques with little evidence to backup their claims. Examples of tweets seen in our dataset include “This Is What Happens When You Press This Point Near Your Ear For One Minute” and “This Leaking From Your Eye Can Be a Sign of a Dangerous Eye Infection”. These click-bait techniques are commonly used to attract large numbers of people to a website in order to generate revenue from adverts.

We then repeat the experiment, only this time analysing all tweeted malware URLs, as classified by GSB. A total of 718 unique malware URLs were recorded during the same timeframe in October 2017 and 543 unique malware URLs in November 2017. When looking at the frequency distribution of delays, the largest peak of GSB blacklisted malware URLs in October occurs at approximately 25 days and was caused by 2 domain names (consisting of 219 and 161 URLs) making up 39% of the October dataset of 718 URLs. The total number of tweeted Malware URLs in October can be seen in Figure 6.4c and shows frequency distribution for the month including the 2 outlying domain names. Figure 6.6b shows the distribution of these top 2 domain names in the dataset, showing that Domain 1 mostly covered days 5 to 25, with its median at approximately 19.5 days. Domain 2 is predominantly spread over the -20 to 12 day delay period, with its median being just over 0 days. Finally, in November 2017, the largest peak appears at around 14 days and is caused by 1 outlying domain name (consisting of 140 URLs) which made up 26% of the dataset. The frequency distribution for November can be seen in Figure 6.4d.

Figure 6.7 shows a boxplot of tweeted GSB blacklisted social engineering and malware URL delays in October and November 2017. The first row shows the distribution of social engineering URLs in October 2017, before the top 7 domains were removed whilst the second row shows the same timeframe but with the top 7 domains removed. Row three shows GSB blacklisted social engineering URLs in November, row four shows malware URLs in October and row five shows malware URLs in November 2017. This shows that, for social engineering URLs, the median delay time was around 7 days in October and just over 0 days in November. For malware URLs the median delay was around 11 days in October and around 8.5 days in November.

These graphs show that GSB appears to be quicker at detecting social engineering websites than malware websites. One reason for this may be that the criteria for the social engineering flag may include a wider net. Therefore, as we saw with the Alexa top 100 domain, some high traffic websites may become blacklisted when they fall into this net. Whereas flagging a website in the malware blacklist requires Google to be certain the

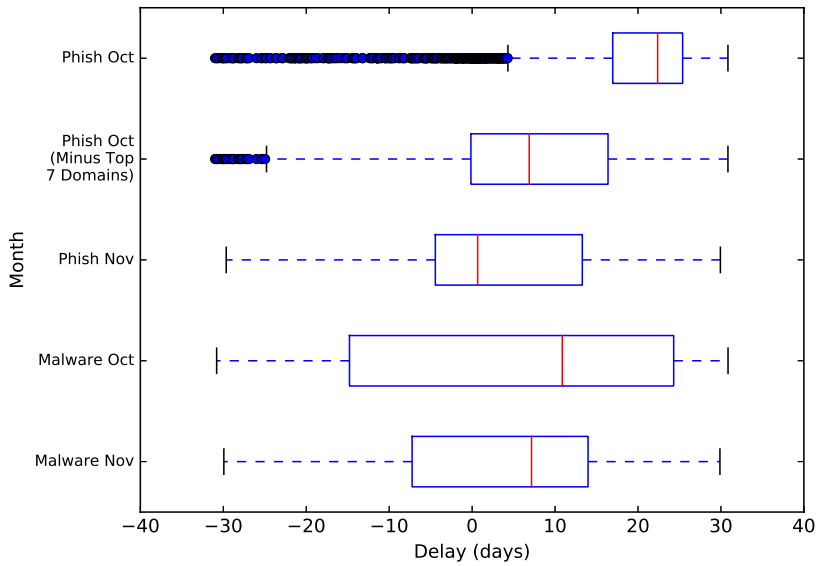


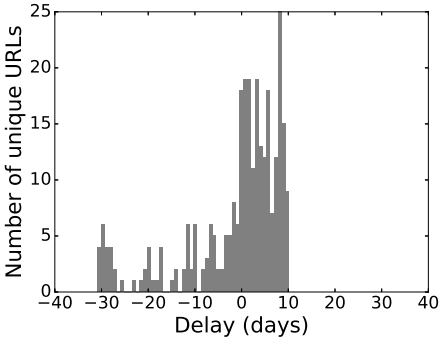
Figure 6.7: Delay from first tweet to first appearing in GSB blacklist – social engineering and malware, October and November 2017.

website is harming – or attempting to harm – the user's computer in some way. This potentially stricter classification may take more time to confirm and may explain why malware is slower to detect than social engineering in our results.

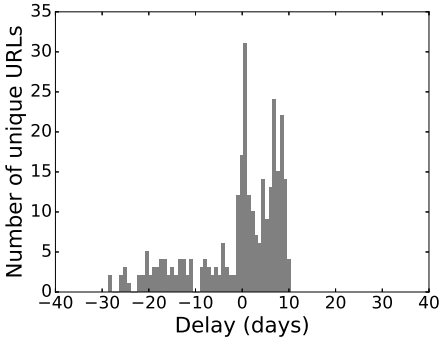
KEY FINDINGS

One of the key takeaways from these experiments is that Twitter allow considerably more URLs to be tweeted ***after*** appearing in the GSB blacklist, compared to the 2010 study [110]. As previously mentioned, this may be because Twitter is relying more on web browsers' built-in protection from malware and phishing URLs. However, one of the biggest weaknesses to this approach is that the built-in blacklists used by web browsers take time to update and this creates an attack space.

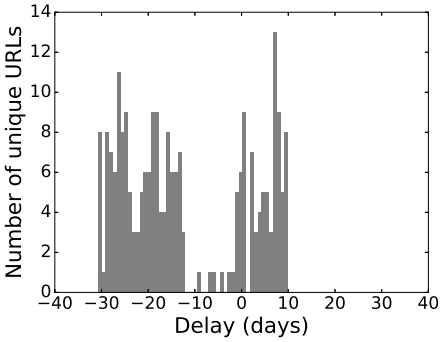
The results in this section also show there is a significant delay – 20 to 30 days in some cases – before URLs are blacklisted. We also see where a combined total of 131,116,820 twitter users are exposed to 2,487 unique blacklisted URLs. This means Twitter users are exposed to these dangerous attacks for a substantial amount of time.



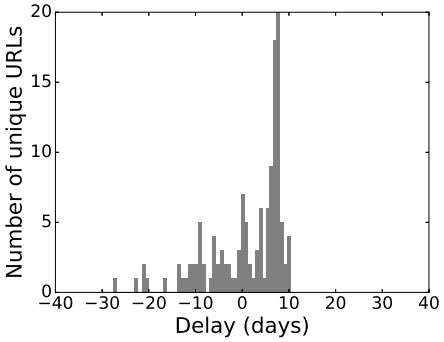
(a) Social Engineering Oct 2017



(b) Social Engineering Nov 2017



(c) Malware Oct 2017



(d) Malware Nov 2017

Figure 6.8: Delay from first tweet to first appearing in GSB blacklist – using Twitter Search API to determine URL first tweet date – social engineering and malware, October and November 2017.

Even though the experiments in this subsection do not identify absolute earliest time of tweet, our delay measurement will always be an underestimate. Therefore the real situation, in terms of Twitter users being exposed to dangerous URLs due to blacklist delay times, is much worse. This methodological weakness is addressed in the next subsection.

6.5.4 Blacklist Delays – Twitter Search API

The experiments in this subsection aim to further improve the accuracy of the experiments carried out in the previous subsection. We do this by making use of Twitter's Search API to determine the original tweet date for blacklisted URLs. Also, because we use Twitter's 1% feed of tweets there may be instances where a URL appears outside of our Twitter Stream. By using Twitter's Search API we can determine when a given URL was tweeted. The measurements taken in this experiment are the same as in

the previous subsection, that is the delay between a blacklisted URL first being tweeted and first appearing in the GSB blacklist within 1 month before of after tweet date, only in this section each URL is searched for on Twitter and the timestamp of that search result is used for the delay calculation. Using this method, in October 2017, 295 social engineering and 230 malware URLs are recorded; their delays can be seen in Figures 6.8a and 6.8c. In November 2017, 284 social engineering and 131 malware URLs are recorded and can be seen in Figures 6.8b and 6.8d.

It is important to note that there are significantly fewer URLs in this part of the dataset. This is because Twitter states that its Search API is not a complete search, therefore some URLs we try to determine original tweet timestamps for cannot be found. In this case these URLs are dropped from the dataset. A clear pattern that emerges in all 4 of these graphs is that there are no URLs with a delay from first Tweet to first blacklist of more than 10 days. This is because Twitter's Search API is limited to 7-10 days; any URLs the system searches for that appeared in the GSB blacklist more than 7-10 days after being tweeted will not show up in a Twitter search if the URL has not been tweeted again since. This limits these graphs, since they show a reduced picture of delays between URLs being tweeted and appearing in the GSB blacklist. However, as seen in the previous two sections, there are still high numbers of URLs already in the GSB blacklist at time of tweet.

KEY FINDINGS

Measurements in the previous subsection do not show the worst-case scenario in terms of delay from first tweet to appearing in blacklist because URLs may have been previously tweeted. However, results in this section, whilst showing fewer URLs in the dataset, do show the worst case scenario for delay from first tweet to blacklist membership. This adds additional evidence that Twitter are not blocking all GSB URLs and may to be relying on other, possibly third-party techniques, to protect its users against attacks.

There is also a significant number of URLs that take between 0 and 10 days to appear in the GSB blacklist – meaning users are exposed to social engineering and malware attacks during these delay periods.

6.5.5 Blacklisted URL Clicks

To explore the impact of tweets that contain blacklisted URLs, we lookup Bitly URLs that either directly appear in or are embedded in the redirection

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

Table 6.6: Total number of tweets containing Bitly URLs, unique Bitly URLs, percentage of all URLs for each category and timeframe, and total Bitly clicks for tweets containing GSB blacklisted phishing and malware URLs, in our dataset, during October and November 2017.

chain that leads to the GSB blacklist, in our dataset. Bitly [43] is a URL shortening service that also provide public analytics for URL clicks, referrers, and location, via an API. By extracting Bitly links from our dataset of tweeted URLs that subsequently appear in the GSB blacklist, we can then use the Bitly API to lookup how many clicks each URL received.

Table 6.6 shows, from our dataset of tweeted URLs that subsequently appeared in the GSB blacklist, the total number of unique Bitly URLs, percentage of all URLs in this category and timeframe, total number of tweets containing Bitly URLs for this category and timeframe, and total number of Bitly URL clicks, during October and November 2017. In October, there were 376 unique Bitly URLs that were either flagged themselves or part of a redirection chain that was in the GSB blacklist as social engineering. These 376 Bitly URLs make up 11% of the 3,273 total social engineering URLs detected in that month in our dataset. The total number of clicks for this 11% is 991,012.

To investigate the impact of tweeting a blacklisted URL to a twitter account with a high number of followers, we extracted a blacklisted URL, from our dataset, that uses Bitly. The blacklisted Bitly URL was tweeted by an account with 3.7 million followers on October 24 and flagged as social engineering in GSB on November 11. The URL received 276 clicks during the week of October 22 2017, of which 270 came from Twitter. 176 of these clicks came from the USA, 19 from Canada, 12 from the UK and the remaining 34 from elsewhere. This URL did not receive any more clicks after the week of October 22 at which point it appears to have been blocked by Bitly. This example shows that a single tweet, from a high follower account, posting a dangerous URL, can receive a high number of global clicks – therefore exposing a large amount of twitter users to the attack. It also shows that GSB took 18 days to add the URL to its blacklist, while Bitly

appears to have blocked the URL much sooner. In this scenario, Twitter appears to have outsourced its filter to Bitly – relying on Bitly to protect Twitter's own users.

KEY FINDINGS

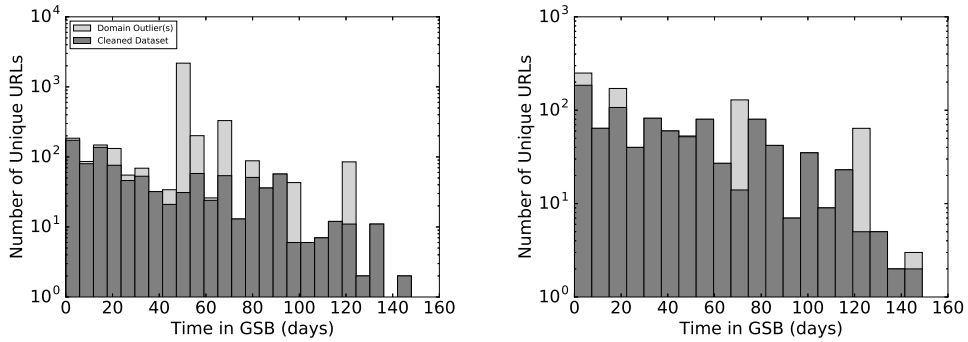
These results show that, in one month alone, 1,052,152 clicks were exposed to dangerous malware and social engineering attacks due to Twitter not blocking harmful URLs. These click metrics represent 11% of our dataset, which is, approximately, 1% of all global Tweets on twitter – giving a sense of the scale and impact caused by Twitter allowing blacklisted URLs to appear on their social network.

6.5.6 Posting Blacklisted URLs to Twitter

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

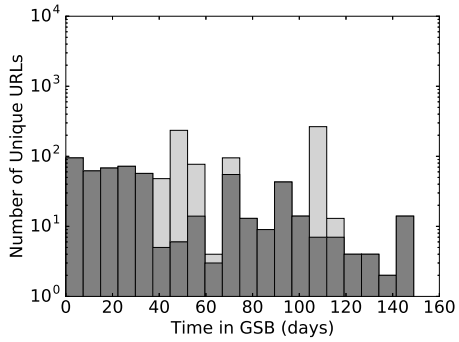
KEY FINDINGS

The outcome of this experiment shows that Twitter appears to be blocking more URLs on the PhishTank and OpenPhish blacklists compared to GSB. Providing further evidence that Twitter is not using the GSB blacklist – therefore exposing users to dangerous URLs.

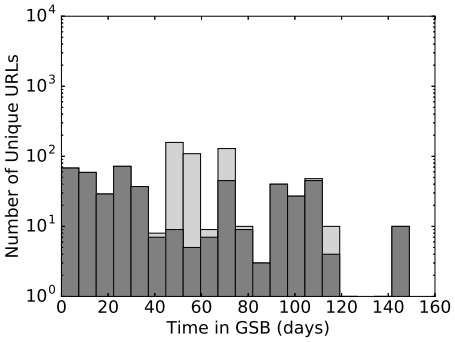


(a) Social Engineering Oct

(b) Social Engineering Nov



(c) Malware Oct



(d) Malware Nov

Figure 6.9: Unique social engineering & malware URLs duration in GSB – first tweeted October, November 2017.

6.5.7 URL Time in GSB

This section explores the duration of time that unique URLs remained in the GSB blacklist for. Each experiment takes all unique URLs that were first tweeted in a given month, then, if a URL is not in GSB at time of tweet, the duration in GSB is calculated as when the system first detects the URL in GSB to when the system last saw the same URL in GSB. If a URL is already in GSB at time of tweet then the GSB library URL hash prefix timestamp is used as time first blacklisted and the time our system last saw the URL in GSB as the final timestamp. The difference between these timestamps is used to calculate total time in GSB for each URL. These duration periods are then plotted on histograms to show the frequency of different duration in GSB for all URLs.

Figures 6.9a and 6.9b show the duration of time that social engineering URLs spent in the GSB blacklist in October and November 2017 and Figures 6.9c and 6.9d show the duration of time malware URLs appeared in the GSB blacklist in October and November 2017. All four of these graphs

have a logarithmic scale on the *y* axis so both high and low numbers are illustrated clearly.

KEY FINDINGS

One of the main conclusions from these graphs is that there is a general downward trend. This shows that, over time, the number of URLs in the GSB blacklist is reducing – meaning that URLs are removed from the blacklist, presumably once they are no-longer perceived as a threat.

Our experiment ran for 150 days and there were over 1,000 URLs remaining in the blacklist, for each category, at the end of the experiment – meaning that many URLs remained in the GSB blacklist for at least 150 days. Some of these URLs may still be dangerous, however, there may be false positives in this blacklist which would mean these URLs are, unnecessarily, being blocked. Exploring long-term false positives in GSB is something we may explore in future work.

6.6 Conclusion

This study examined how effective URL blacklists are in protecting Twitter users against phishing and malware attacks. We analysed over 182 million URL-containing public tweets collected from Twitter's Stream API, over a 2 month period, and compared these URLs against 3 popular social engineering, phishing, and malware blacklists. Our main discovery was that, although the majority of phishing and malware URLs are detected by the GSB blacklist (which is used by popular web browsers) within 6 hours of being tweeted, there are still a large number of URLs that take at least 20 days to appear in GSB. We discovered 4,930 tweets containing URLs leading to social engineering websites that took between 18 and 30 days to appear in the blacklist. Between them, these 4,930 tweets had been tweeted to over 131 million Twitter users. We also discovered 1,126 tweets containing 376 blacklisted Bitly URLs that had received a combined total of 991,012 clicks. These URLs represented 11% of the total blacklisted social engineering URLs in that month. The fact that the GSB blacklist can take weeks to detect dangerous URLs poses serious security risks to Twitter users: tweets containing blacklisted URLs are sent to large numbers of followers and receive a significant amount of clicks, thereby exposing users to dangerous websites. Conversely, and surprisingly to us, there are large numbers of URLs being tweeted that have already been blacklisted by

GSB. This strongly indicates that Twitter is not using the GSB blacklist to block malicious tweets at the time of tweeting, contrary to what was once reported to be the case [200]. In summary, whilst blacklists are reasonably effective at protecting Twitter users from phishing and malware attacks, there is still an unprotected space that leaves Twitter users vulnerable.

7

Web Browser Phishing Detection

Browsing for Phish: An Analysis of Web Browser Phishing Detection Technology

OUTLINE

This chapter investigates how effective popular web browsers are at detecting phishing websites. Our aim is to explore whether or not Twitter can rely on web browsers to protect users against phishing attacks.

One of our key findings in the previous chapter was that large numbers of blacklisted URLs are publicly tweeted on Twitter. We also discovered that the GSB blacklist can be slow to detect these URLs – which may create an opportunity for attackers. Can web browsers provide effective defence against such attacks?

This chapter is an edited version of our CDT summer project entitled: *Browsing for Phish: An Analysis of Web Browser Phishing Detection Technology (2015)*.

7.1 Introduction

The aim of our study is to carry out an investigation into the effectiveness of web browsers' built-in phishing detection technology – to explore how effective their anti-phishing defences are – and to assess how well users are being protected from phishing attempts carried out by attackers.

Many of today's popular web browsers (specifically: Chrome, Safari, Firefox, Opera, and Vivaldi – see Section 2.6.1) have the GSB blacklist built

in. This is designed so that users are protected from phishing and malware attacks. Our methodology to measure web browser phishing detection effectiveness involves using a **different** data set of blacklisted phishing URLs from GSB. Our data set of testing URLs comprises the PT and OP blacklists to determine web browser detection rates for URLs that are **blacklisted** and **not-blacklisted** by GSB at time of test. This methodology allows us to investigate web browser phishing detection rates for both known and unknown phishing websites – therefore determining web browsers' heuristic and blacklsited phishing detection rates.

Our study involves creating an automated testing environment to assess many of today's popular web browsers, across a variety of operating systems, to determine how effective these browsers are at detecting phishing attacks. Various phishing detection technologies, such as blacklists and dynamic page analysis, are examined for their effectiveness.

Since most of the popular web browsers come pre-installed with anti-phishing technology built in, the core focus of our study is to analyse how effective these browsers are at detecting phishing websites. The only way for a victim to view a phishing website is through an internet browser, therefore it makes sense that this window into the world wide web provides built-in phishing detection. In theory, this should give the user a better chance at being protected from phishing website attacks.

To determine which web browsers should be included in our study, usage statistics can be helpful to assess which browsers are the most commonly used to browse the web. As we saw in Section 1.9.3: *Web Browser Usage Statistics*, the most popular web browsers in January 2020 were: Chrome, Safari, Firefox, Internet Explorer (Edge), and Opera.

As previously mentioned, the overall aim of this study is to carry out an investigation into the effectiveness of web browser phishing detection technology. The original aim of our study was to focus specifically on browser plug-ins and extensions that enhance the user's protection against phishing websites. However, soon into the initial phase of our study (assessing which tools to test) it became apparent that many of these browser plug-ins are becoming less popular. The main reason for this is that web browsers' built-in phishing detection has made such plug-ins redundant. So that steered our study into a new direction: how effective are the most popular web browsers (running across a variety of operating systems) at detecting phishing websites? The main aims of our study are:

1. Establish an automated testing environment in order to:
    a) Run the most popular web browsers (Chrome, Safari, Firefox, and Internet Explorer) under the most popular operating systems (Windows, OSX, and Linux Ubuntu)
    b) Execute automated phishing detection tests on above web browsers

2. Determine phishing detection capabilities of most popular web browsers against both known (backlisted) and unknown (non-blacklisted) phishing websites
3. Reverse engineer web browser phishing detection methodology
4. Can phishing detection be circumvented – If so how?
5. Assess effectiveness of phishing warnings produced by web browsers

7.2 Related Literature Summary

As we previously explored (in Section 3.3: *Threat Intelligence & Blacklists*), numerous studies have investigated the effectiveness of web browser phishing detection. Most notably: Zhang *et al.* (2006) [311] and AV Comparatives (2012) [23]. A key limitation of the 2006 [311] study was that it focused on web browser extensions and not on the "vanilla" web browsers themselves (i.e. without any add-ons). The main limitation of the 2012 [23] study was that, whilst it tested popular web browsers, the research was limited to the Windows operating system.

In our experiments we will test the phishing detection effectiveness of web popular browsers. We improve existing research by not using any additional extensions or plug-ins with the web browsers. The web browsers will be "off the shelf" installations. We will further improve existing research by testing the web browsers across multiple operating systems. Existing work does not identify the heuristic phishing detection rates of web browsers (i.e. phishing websites that are not in the GSB blacklist). We address this limitation by characterising URLs as either blacklisted or not-blacklisted by GSB at time of test.

Our experiment to evaluate the effectiveness of web browser warnings in preventing users from visiting phishing websites will build on existing studies (summarised in Table 2.5). We repeat the existing studies to see if there have been any changes in the web browser warnings. We also combine metrics from multiple existing studies for the first time to evaluate the web browser warnings' effectiveness.

7.3 Overview of Experiments

A key part of this study is to carry out automated testing on a number of different web browsers across the three main operating systems. This automated testing needs to appear natural to the computer and replicate the same interactions that a user of the computer would undertake. We use the Web Browser Testing Suite (WBTS) to carry out our experiments. The WBTS system design overview is described in Section 4.1.3, the methodology is

outlined in 4.7.3, and the implementation details are in 4.8.9. The main experiments that we carry out are described below.

Our first experiment is designed to test the phishing detection rates of 5 popular web browsers (Firefox, Chrome, Chromium, Safari, and Internet Explorer) across the most popular desktop operating systems (Microsoft's Windows, Apple's Mac, and Ubuntu for Linux). For this experiment, we categorise the phishing URLs to be tested into: blacklisted (at time of test), non-blacklisted (at time of test), and combined. We then compare the detection rates of web browsers across multiple operating systems.

Our second experiment tests how much time it takes for a website to become blacklisted.

Our third experiment analyses Chromiums source code to reverse engineer its phishing detection methodology. This is used to verify whether Chrome can detect phishing websites that are not blacklisted.

Our final experiment analyses warnings that the popular web browsers display when a user attempts to visit a phishing website. Our analysis is used to determine the effectiveness of warnings that are displayed to users.

7.4 Results

In total, 398 unique URLs were tested during various different experiments against the web browsers: Internet Explorer, Chrome, Chromium, Firefox, and Safari – during August 2015. The main results of these experiments are shown below

7.4.1 Blacklisted Phishing Websites

The results in Table 7.1 show the total number of phishing websites that were detected that were known to be in the Google Safe Browsing Blacklist. 207 phishing URLs were tested in this experiment. Some of the variations in detection rates across operating systems are likely to be caused by a delay in the web browser updating its own local copy of the blacklist.

7.4.2 Non-Blacklisted Phishing Websites

The results in Table 7.2 show the total number of detected phishing websites that were not listed in the Google Safe Browsing blacklist. In total 191 phishing websites were tested, (however, during the experiments this number fell to 171 since these websites either went offline or were added to the blacklist). It is interesting to note that many of the phishing websites that bypassed the non-blacklist detection methods were compromised WordPress websites.

Browser	Detection Rate
Internet Explorer (on Windows)	98%
Chrome (on Windows)	94%
Firefox (on Windows)	95%
Safari (on Mac Yosemite)	99%
Chrome (on Mac Yosemite)	100%
Chrome (on Ubuntu)	94%
Chromium (on Ubuntu)	95%
Firefox (on Ubuntu)	81%

Table 7.1: Blacklisted phishing website detection rates of popular web browsers.

Browser	Detection Rate
Internet Explorer (on Windows)	78%
Chrome (on Windows)	38%
Firefox (on Windows)	71%
Safari (on Mac Yosemite)	58%
Chrome (on Mac Yosemite)	74%
Chrome (on Ubuntu)	72%
Chromium (on Ubuntu)	68%
Firefox (on Ubuntu)	74%

Table 7.2: Non-blacklisted phishing website detection rates of popular web browsers.

These results show that, when presented with a phishing website that is not in the Google Safe Browsing database, the web browsers were able to detect 38% to 78% of phishing websites.

7.4.3 Total Detection Rate

Table 7.3 shows our results for the detection rates of web browsers on both blacklisted and non-blacklisted phishing websites, across 3 popular operating systems (OS). For comparison, Table 7.4 shows the results from the AV Comparatives (2012) [23] study – which was limited to the Windows OS. We address the single operating system (Windows) limitation of the 2012 study by adding 2 popular operating systems: Mac and Ubuntu. By comparing our results with the 2012 study, we see that it can be useful to categorise results into blacklisted and non-blacklisted at time of test. Since

Browser	Detection Rate
Internet Explorer (on Windows)	85%
Chrome (on Windows)	66%
Firefox (on Windows)	83%
Safari (on Mac Yosemite)	79%
Chrome (on Mac Yosemite)	87%
Chrome (on Ubuntu)	83%
Chromium (on Ubuntu)	82%
Firefox (on Ubuntu)	78%

Table 7.3: Total combined (blacklisted and non-blacklisted) phishing website detection rates of popular web browsers across multiple OSs.

Browser	Detection Rate
Opera	94.2%
Internet Explorer	82.0%
Chrome	72.4%
Safari	65.6%
Firefox	54.8%

Table 7.4: AV Comparatives (2012) [23]: web browser phishing detection test results – limited to Windows OS.

the combination of both can cause the results to be misleading. The 2012 results show that Chrome saw a 72.4% detection rate. In our results, we see Chrome's detection rates are: 66%, 87%, and 83% across the operating systems: Windows, Mac, and Ubuntu, respectively. The 2012 study tested Chrome on the Windows operating system only. Therefore it is interesting to note the lower detection rate in our results compared to 2012.

KEY FINDINGS

Our results in this section show that the web browsers we tested were able to detect 81% to 100% of known (*i.e.* blacklisted) websites but only 38% to 78% of unknown (*i.e.* non-blacklisted) websites. This provides evidence that the web browsers could miss up to 19% of known – and up to 62% of unknown – phishing websites.

7.4.4 Comparrison Across Multiple Operating Systems

The Google Chrome web browser was tested across all three operating systems to compare its detection rate. As the results above show, there was quite a variation between these detection rates. The total detection rate were: Windows: 66%, Mac: 87% and Ubuntu: 83%. One of the main differences that stands out from the results was the low detection (38%) rate of non-blacklisted URLs on the Windows operating system.

7.4.5 Time to Blacklist Websites

In this section we look at the results from two experiments which tested a total of 200 phishing URLs to see how long it took for the URLs to appear in the Google Safe Browsing blacklist. In the first experiment, of the 100 tested phishing URLs, only 3 appeared in the blacklist after 48 hours. In the second test, of 100 phishing URLs, 11 were appeared in the blacklist after 48 hours.

7.4.6 Chromium Phishing Detection Analysis

It is interesting to observe that, based on the above results, Chrome doesn't rely entirely on the Safe Browsing blacklist of phishing websites for its phishing detection mechanism. There is clearly further technology under the hood which is capable of detecting phishing websites that have not previously been blacklisted. This section of the results shall attempt to reverse engineer Google Chrome's phishing detection methodology.

The results above show that both the Chrome and Chromium web browsers produce similar phishing detection rates. Therefore it's likely that Chrome uses a similar phishing detection methodology to Chromium. Since Chromium is an open source version of Chrome, reverse engineering Chromium's phishing detection methodology is likely to produce a similar methodology to that of Chrome.

We dissected Chromium's source code [100] to reverse engineer its main phishing detection methodology. Based on our analysis of the code, Chromium's phishing detection algorithm is outlined below:

1. If URL in GSB blacklist: show phishing warning
2. Extract features from browser
    a) has page been visited before (number of visits in history)
    b) number of times URL typed into omnibox
    c) number of times URL reached by clicking a link
    d) number of times visited more than 24 hours ago

3. Extract features from website
    a) URL host features (e.g. is the URL hostname an IP address?)
    b) URL host tokens (e.g. number of hostnames greater than one looks phishy)
    c) URL path token features (extract URL path)
    d) DOM HTML form features
        i. does page have <form> tags?
        ii. does “action” element of form point to different url?
        iii. does page have <input type=“password”> elements?
    e) DOM HTML link features
        i. number of links that point to a different domain
    f) DOM HTML script features
        i. set if more than one <script> element
        ii. set if more than six <script> elements
    g) Other (e.g. number of image sources – “src” – that point do different URL)
    h) Page term features (i.e. words on the page)
    i) SHA-256 hashes of terms so phishers cannot work out what terms are used for phishing classification
4. Phishing classifier:
    a) Produce a feature map of above features and hand to below Scorer, which computes the probability that the page is phishy. If the page is phishy, show phishing warning
5. Scorer
    a) Uses the above features to calculate a phishing probability score based on machine learning approach

Looking at this algorithm, part 4: *Phishing Classifier* takes various features from part 2: *Browser Features* and 3: *On Page Features* to produce a feature map containing all of the extracted features that are met. Part 5: *Scorer* uses a machine learning approach to calculate the probability that the given website is a phishing attack. This machine learning approach is regularly modified in order to provide the most accurate phishing detection. It is noticeable that this detection methodology takes two main factors into account: browser features (data about how often, if at all, this website has been visited recently) and page features (use of password input fields, number of script elements, use of external images et). This approach, along

with these features, is similar to previous studies noted in our Existing Literature Section: 3.6: *Heuristic Phishing Detection*. The use of this algorithm in Chromium (and probably Chrome) would explain why phishing websites that do not exist in the phishing blacklist are still able, in some cases, to be detected. It is interesting to note that no visual similarity comparison technique could be found in the Chromium source code. This is probably because maintaining a database of trusted websites (such as banks, social media, online auctions etc) for comparing visual similarity would likely be time consuming. Although storing their visual similarity signature would not require much space on the clients hard drive.

KEY FINDINGS

Our findings in this section show how Chromium's heuristic phishing detection technique analyses, for a given webpage, various browser and website features, such as: how many times a webpage has been previously visited, whether the webpage's hostname is an IP address, whether the webpage contains a password input, etc. A machine learning classifier then processes these features to calculate a phishing likeliness probability score for that webpage.

In Section 9.4 we will discuss a number of hypothetical techniques that could bypass the heuristic phishing detection, such as: hijacking previously visited websites, misspelt domain names of popular brands (*e.g. ebayy.com*), leveraging JavaScript to re-write webpage contents to bypass initial checks, etc.

7.4.7 Effectiveness of Browser Warnings

All of the web browsers tested during our experiments displayed an *active* warning that blocked the detected phishing website from loading. However, the effort required by the user to circumnavigate these warnings (e.g. click count, time spent on warning, brevity) varies. In Google Chrome (Figure 7.1) the user must click on "Details" (Figure 7.2), at which point the message "if you understand the risks to your security, you may visit this infected site" is displayed. The user is required to click a link with the text "visit this infected site". Additionally, there is only 1 button shown on the warning page, which contains the text "Back to safety", therefore encouraging the users to follow the command.

Microsoft Internet Explorer (Figure 7.3) displays a similar warning message to users: it shows an image of a green tick (positive reinforcement)

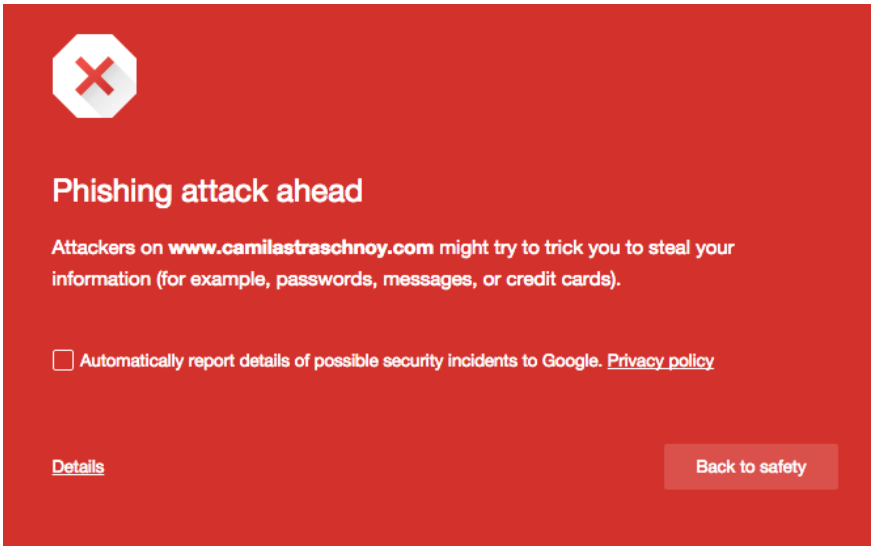


Figure 7.1: Google Chrome Phishing Warning.

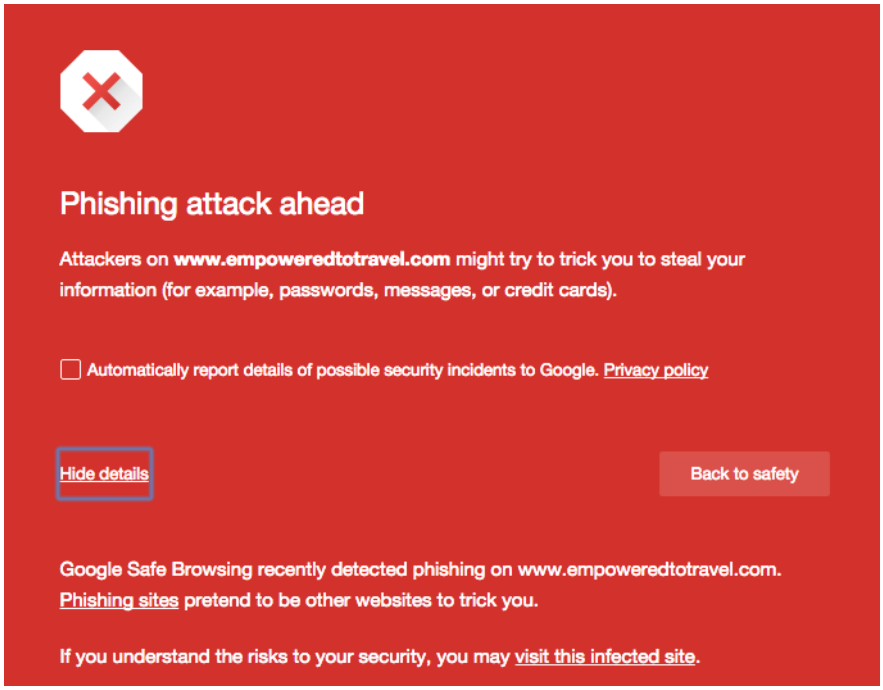


Figure 7.2: Google Chrome Phishing Warning Details.

next to a link with the text “go to my homepage instead” to encourage the user to navigate away from the website. Additionally, an image of a red cross is shown next to a link with the text, in smaller font to the previous link, “disregard and continue (not recommended)” to discourage the user from visiting the phishing website. Although not much effort is required from the user to continue navigating onto the phishing website, the use of

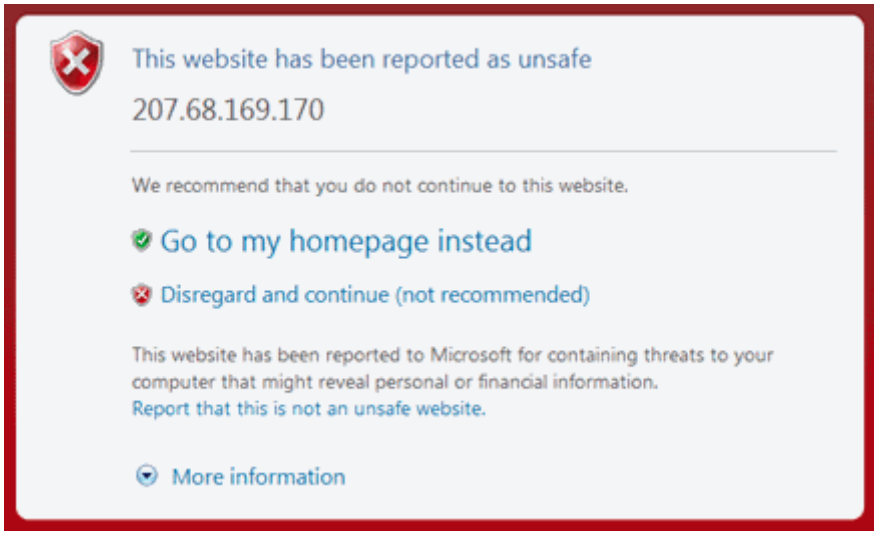


Figure 7.3: Microsoft Internet Explorer Phishing Warning.

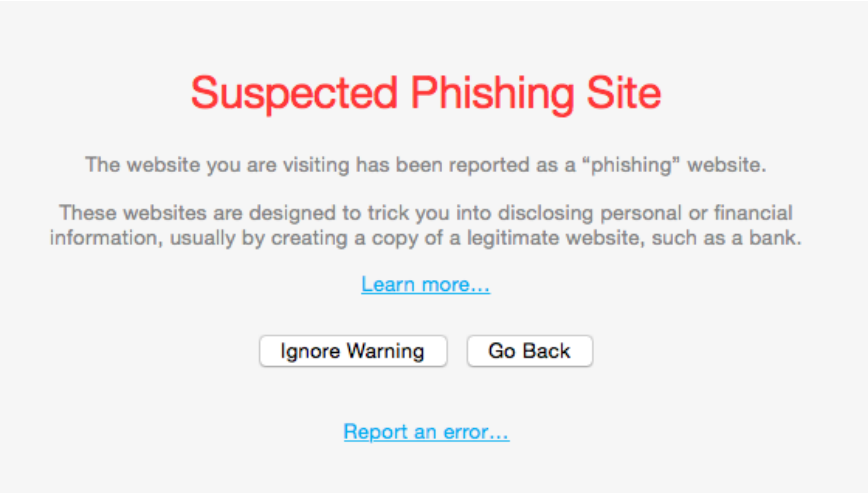


Figure 7.4: Apple Safari Phishing Warning.

effective language and red warning imagery acts as a clear deterrent.

The phishing warning presented to Apple Safari (Figure 7.4) users is possibly the weakest, in terms of encouraging users to navigate away from the phishing website. The main reason is that the first button shown has the text "Ignore warning" followed by a second button which features the text "Go Back". It would require very little effort for a user to click on the "Ignore Warning" button due to its convenient placement and lack of discouraging language or colour.

Finally, Mozilla Firefox (Figure 7.5) presents its users with a phishing warning that requires a little more effort in order to continue navigating onto the phishing website. Only two buttons are displayed to the user,

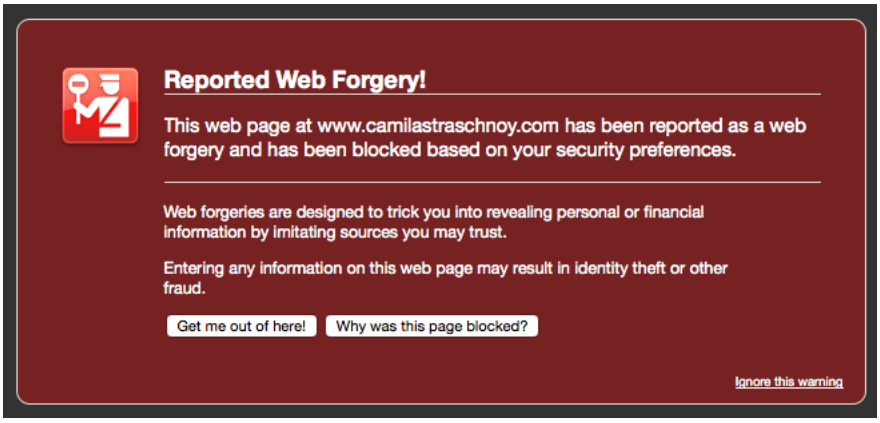


Figure 7.5: Mozilla Firefox Phishing Warning.

	Web Browser			
Metric	**Chrome**	**Firefox**	**Safari**	**IE**
Neutral information	No	No	No	No
System decision	Yes	Yes	Yes	Yes
Passive warning	No	No	No	No
Active warning	Yes	Yes	Yes	Yes
Click count	2	1	1	1
Time spent on warning	Low	Low	Low	Medium
Technical jargon	Low	Low	Low	Low
Reading level	Low	Low	Low	Low
Brevity	High	High / Medium	High	High
Specific risk description	Yes	Yes	Yes	Vague
Illustration	Yes	Yes	No	Yes
Risk level	Yes	Yes	Yes	Yes

Table 7.5: Effectiveness of web browser warnings results.

which contain the text “Get me out of here!” and “Why was this page blocked?”. The latter button takes the user to a Mozilla support page which explains how the browser’s built-in phishing and malware protection work. The user is required to click on the text at the bottom right of the screen, in a smaller font, which features the text “Ignore this warning”. Whilst the placement of this bypass link is effective at discouraging the user to click on it, the use of language and colour is not particularly strong.

Table 7.5 shows the results of our evaluation of the effectiveness of web browser warnings that we tested. We see that none of the web browsers show neutral information or passive warnings. This a good for effectiveness because the presence of these metrics has been shown to reduce warning effectiveness [300, 76]. All web browsers make a system decision and show an active warning that requires the user to take action: continue to visit the phishing website or navigate away. This is good for effectiveness. The Chrome warning required the user to make 2 clicks to visit the phishing website, whereas the rest of the browsers required 1 click. Existing literature shows that, whilst click count does contribute to web browser warning effectiveness, its impact is low [6]. We calculate time spent on warning as the time required to read all text displayed. IE displayed the most amount of text both in the warning and in the buttons, therefore requiring more time to read. The rest of the browsers all required low time to read. This improves effectiveness. All of the web browsers displayed a low level of technical jargon and low reading level. This means that the warnings are accessible to a wide audience range of reading and technical comprehension levels – good for effectiveness. The Firefox warning message was high / medium brevity. This is because the text is quite verbose and repeats itself in places. All other browsers scored high in brevity (i.e. the messages were brief) – good for effectiveness. The specific risk description displayed in the IE warning was somewhat vague because it described phishing websites as “unsafe” and did not mention the specific threat: phishing. All of the other web browsers specified that the warning was about a phishing threat. Safari did not display any illustrations on its warning, whereas all other web browsers did. Illustrations can improve effectiveness. All warnings used the colour red (red improves warning effectiveness): Chrome features a red background with white text, Firefox features a dark red background with white text, Safari features a red title ("Suspected Phishing Site"), and finally IE features a red border. However, it is interesting to note that each browser’s warning uses red in different ways. From Chrome’s full bright-red background, to Safari’s subtle red title. These alternate uses of red may impact their effectiveness – although these fine-grained measures are outside the scope of our study. Overall we see that all 4 of the web browsers we tested show effective warnings to deter users from visiting phishing websites. Chrome scored the highest (i.e. its warnings were the most effective).

It is important to note that there are many other effectiveness metrics that we were unable to measure in this study. Such as: technical knowledge of user, focus level of user (e.g. are they in a distracted state, noisy place), etc. One such metric that we were unable to measure is warning fatigue. Akhawe *et al.* (2013) [6]) state that the effectiveness of web browser warnings will depend on how many similar warnings the user has recently

seen. Warning fatigue is difficult to measure because it is dependent on user context and the situation.

KEY FINDINGS

All of the web browsers we tested in our experiments displayed a warning page to the user once a phishing website had been detected. These warning pages required the user to take a specific action – which was mostly deterred – in order to continue navigating to the potential attack website.

However, varying degrees of effort and attention were required from the user in order to bypass the web browsers' warnings. The warning which required the most amount of effort and attention from the user, in oder to bypass it, was Google Chrome, since the bypass link is initially hidden from the user. In contrast, Apple Safari presents a weaker warning since the bypass button features prominently on the page without much discouragement to click on it. Therefore, we see that Chrome presents the most effective web browser warning to prevents its users from visiting phishing websites.

7.5 Conclusion

This study has successfully created a comprehensive testing environment to assess the effectiveness of web browser anti-phishing technology across a variety of different operating systems. Through this test suite, popular web browsers were presented with a number of phishing websites and their phishing detection rates measured.

Our study was able to conclude, by testing both phishing blacklist and phishing heuristic detection techniques, that most of today's popular web browsers are able to detect, on average, 80% of both blacklisted and non-blacklisted phishing websites.

The phishing detection methodology of the Chromium web browser was analysed, reverse engineered, and described; illustrating how the web browser is able to detect non-blacklisted phishing websites – without requiring a blacklist.

Overall, we see that automated phishing detection, which is built into all of the popular web browsers we tested, can be effective – although the heuristic detection of non-blacklisted phishing websites is typically lower, such as the Google Chrome web browser running in our Windows environment which only detected 38% of non-blacklisted phishing websites.

Our results also show that web browsers display active warning pages that encourage users to avoid visiting detected phishing websites. Although not flawless, some effort and attention is required from users that want to bypass these warnings in order to access the phishing sites – although, in some cases, users could accidentally click through to the phishing websites.

CHAPTER SUMMARY

Results from our previous study (seen in Chapter 6: *Time-of-Post Twitter Study*) showed us that large numbers of blacklisted URLs are tweeted on the social network – and that GSB can be slow to detect these URLs. In this chapter we saw that web browsers do indeed protect users against phishing attacks. However, the effectiveness of this built-in defence mechanism depends on a number of factors, such as whether or not a website is blacklisted and blacklist update delay times.

In this chapter, we also saw that web browsers are reasonably effective at blocking known (*i.e.* blacklisted) phishing attacks. This might suggest that Twitter does not need to protect its own users against phishing attacks – because Twitter can rely on the web browsers as a defence mechanism. However, as we saw in Table 7.2, web browsers are not as effective at detecting fresh (*i.e.* non-blacklisted) phishing websites – detecting only 38% to 78% of these. Also, users are not completely protected against known phishing attacks, since blacklists take time to update – which can create a window of opportunity for attackers. Therefore, if Twitter is relying on web browsers to protect their users against phishing attacks, then Twitter could be leaving its users exposed and vulnerable to these attacks.

8

Time-of-Click Twitter Study

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

OUTLINE

This chapter presents an edited version of our multi-award-winning research paper: BELL, S., AND KOMISARCZUK, P. Measuring the Effectiveness of Twitter's URL Shortener (*t.co*) at Protecting Users from Phishing and Malware Attacks. In *Proceedings of the Australasian Computer Science Week Multiconference* (2020).

In this study we investigate how effective Twitter's URL shortening service (*t.co* [276]) is at protecting users from phishing and malware attacks. We show that over 10,000 unique blacklisted phishing and malware URLs were posted to Twitter during a 2-month timeframe in 2017. This lead to over 1.6 million clicks which came directly from Twitter users – therefore exposing people to potentially harmful cyber attacks. However, existing research does not explore if blacklisted URLs are blocked by Twitter at time of click.

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

Twitter's URL shortener is not particularly effective at filtering phishing and malware URLs – therefore people are still exposed to these cyber attacks on Twitter.

8.1 Introduction

This chapter follows on from our previous study, in Chapter 6: *Time-of-Post Twitter Study*, where we investigated Twitter's use of blacklists along with blacklist delay times. Based on some of the results in that previous study, we want to further explore what impact Twitter's URL shortening service might have on protecting its users against phishing and malware attacks.

In June 2010 Twitter announced [279] that it was working on its own URL shortening service (*t.co*) – this service launched a year later in June 2011[42]. All URLs posted to Twitter (i.e. tweeted) are shortened via Twitter's URL shortening service (*t.co*). If a URL has already been shortened with an existing URL shortener(s) (such as Bitly, Goo.gl, etc) then a redirection chain is formed, starting with *t.co*. This gives Twitter full control over and monitoring of URLs posted to, and clicks leaving, its network.

The aim of our study is to explore how effective Twitter's URL shortening service, (*t.co*) is at protecting Twitter users from phishing and malware attacks. A 2010 study [110] provided evidence to suggest that Twitter is not using GSB effectively to protect users from such attacks. The study showed that 100,000 phishing and malware URLs were tweeted to the social network – therefore endangering users. Some of these blacklisted URLs accumulated 1.6 million clicks from Twitter users – although the total number of URL clicks in the study includes phishing, malware, and scam URLs. These specific results from the 2010 paper were part of a much larger study aiming to characterise spam and analyse features unique to Twitter. As previously mentioned, Twitter'a URL shortening service launched in 2011, therefore the 2010 [110] study was not able to research *t.co* URLs because the study was conducted before Twitter implemented its URL shortener.

In 2017 [40] we investigated the delay times between phishing/malware URLs being tweeted to appearing in blacklists. In this follow-up study to our 2017 work, we build on our existing research to investigate Twitter's URL shortener.

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

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

Our main research questions are:

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

We organise the remainder of this chapter as follows. Section 8.2 explains our methodology, Section 8.3 provides an overview of our experiments. Section 8.4 presents our key measurement results and interpretations, followed by our conclusion in Section 8.5. The results of this study are discussed in Section 9.5: *Time-of-Click Twitter Study*.

8.2 Methodology

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

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

determine which tweeted URLs have been blocked by Twitter. This system is carefully rate limited to ensure we do not flood Twitter's servers.

8.3 Overview of Experiments

The key experiments we carry out in this study are:

1. Measure number of phishing and malware URLs posted to Twitter. Also, to assess impact, measure clicks from Twitter users to these blacklisted URLs (carried out in our previous study, seen in Chapter 6: *Time-of-Post Twitter Study*)
2. Measure Twitter Filtering: does Twitter filter URLs (via its *t.co* URL shortener)? If so, does Twitter block blacklisted URLs?
    - 24-hour measurement period
    - 7-day measurement period

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

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

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

Both experiments involve collecting URL containing tweets, via Twitter's Stream API, then checking these URLs to determine if they are blocked by Twitter and/or blacklisted by GSB, OP or PT. For the 24-hour measurements

this check is carried out once for Twitter filtering and GSB membership. For the 7-day experiment this check is carried out 25 times over the duration of 7 days. The specific methodology for each experiment are described below.

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

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

8.4 Results

8.4.1 Time-of-Post Defence Summary

As we saw from our previous study's results, in Sections 6.5.1: *Twitter Dataset Analysis* and 6.5.5: *Blacklisted URL Clicks* of Chapter 6: *Time-of-Post Twitter Study*, over 10,000 phishing and malware URLs were posted to

Twitter during our 2-month measurement timeframe. More than 45% of these URLs took over 1 month to become blacklisted. This means these blacklisted URLs were publicly available for Twitter users to visit – therefore exposing people to these potentially harmful attacks.. This lead to over 1.6 million clicks which came directly from Twitter users – therefore exposing people to potentially harmful cyber attacks. In the next subsection we assess if Twitter implements additional protection for its users against these attacks. We do this by investigating Twitter's URL shortener (*t.co*) to determine if a filtering system is implemented.

8.4.2 Investigation of Twitter Filtering

Building on from our previous subsection, where we show that phishing and malware URLs are posted to Twitter and receive clicks from Twitter users, this subsection investiagtes if Twitter implements a filtering system to protect its users against blacklisted URLs. This subsection is split into two key measurement experiment: 24-hour and 7-day.

24-hour Measurement Period

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

Table 8.1 shows an overview of the total number of tweeted, blocked by Twitter (via *t.co*), and blacklisted URLs we collected during November, December, and January 2018-19. The table also shows the total number of tweeted URLs that were both blocked by Twitter **and** blacklisted (in one of the 3 blacklists). The total number of tweeted URLs we collected is broken down into total unique *t.co* (shortened) and total unique *unshortened* URLs. All URLs posted to Twitter (i.e. tweeted) are shortened via Twitter's URL shortener, *t.co*. We define an "*unshortened* URL" to be the original tweeted URL that has an associated (i.e. shortened) *t.co* URL. A single tweeted URL can be shortened to one or more different *t.co* URLs – for example: a number of different Twitter users may tweet the same URL which gets shortened to separate *t.co* URLs. Therefore we see a higher number of unique *t.co* URLs in the results table compared to unique *unshortened* URLs. The table also shows how many unique *t.co* URLs were blocked by

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

Table 8.1: Overview showing total number of tweeted, blocked by Twitter (via *t.co*), and blacklisted URLs up to 24-hours after time of tweet, in our dataset. Experiments conducted during Nov, Dec, & Jan 2018-19 (M = million).

Twitter and how many *unshortened* URLs were blocked by Twitter; how many tweeted URLs appeared in the 3 main blacklists (GSB, OP, and PT), along with categorisations for GSB; and finally how many tweeted URLs were blocked by Twitter **and** blacklisted, along with which blacklists they appeared in.

The results in Table 8.1 show that Twitter's defence strategy does indeed involve filtering URLs at time of click via their URL shortener, *t.co*. However, we also see that, of the tweeted URLs which were blacklisted, only 7-18% were also blocked by Twitter: 18% (12 of 67) in November, 14% (16 of 118) in December, and 7% (18 of 251) in January. This shows that, in our dataset, over 80% of blacklisted phishing and malware URLs posted to Twitter are not blocked by the social network – therefore allowing Twitter users to click on these potentially dangerous links.

These results show that Twitter blocked over 14,000 unique, unshortened URLs during our 3-month measurement period – but only 2% of these were blacklisted phishing and malware URLs in GSB, OP, or PT. It is important to note that Twitter's Rules [278] mainly focus on preventing graphic violence, abusive behaviour, violence and physical harm, hateful conduct, imperson-

ation, and spam. Twitter does explicitly state that malware/phishing is not allowed on its platform: "You may not publish or link to malicious content intended to damage or disrupt another person's browser or computer or to compromise a person's privacy". However, this is a small part of its extensive list of rules. Therefore, it is likely that the majority of these URLs blocked by Twitter are for violations of its rules other than phishing/malware. Analysis of the categorisation of URLs blocked by Twitter, that are not phishing/malware, is outside the scope of our study.

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

KEY FINDINGS

Results from our 24-hour measurement experiments show that Twitter's defence strategy does indeed involve filtering URLs at time of click via their URL shortener, *t.co*. However, we also see that, of the tweeted URLs which were blacklisted, only 7-18% were also blocked by Twitter.

During our 3-month measurement study, Twitter blocked over 14,000 unique URLs – but only 2% of these were blacklisted phishing and malware URLs in either the GSB, OP, or PT blacklists.

Our results also show that a larger volume of phishing URLs than malware URLs were posted to Twitter during our measurement study – a similar finding to our 2017 study (Chapter 6: *Time-of-Post Twitter Study*). This may suggest that Twitter continues to be an effective platform for malicious users to carry out social engineering attacks on.

It is important to note that, during these 24-hour measurement experiments, tweeted URLs were checked for GSB blacklist membership once within 10 minutes of tweet and checked for Twitter filtering up to 4 times within 24 hours of tweet. Tweeted URLs were not measured beyond 24 hours. We improve on this methodology in the next subsection.

7-day Measurement Period

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

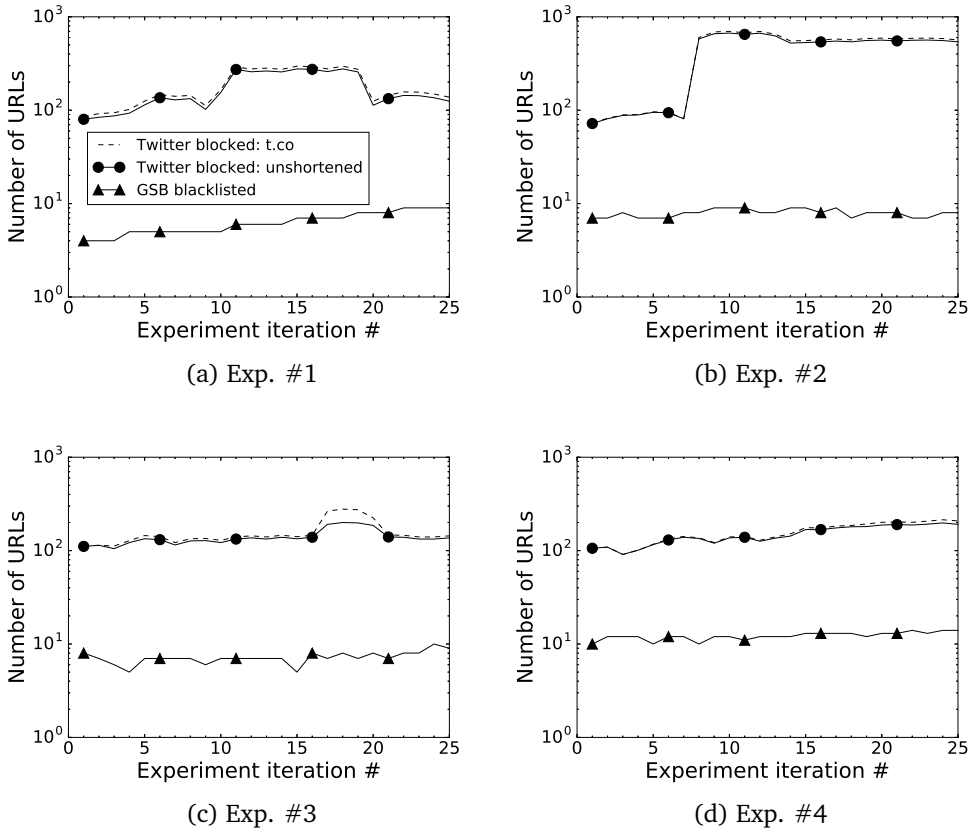
Table 8.2 shows the overview of results for these experiments and

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

Table 8.2: Overview of experiments #1-4 showing total number of tweeted, blocked by Twitter (via *t.co*), and blacklisted URLs, in our dataset. Each experiment measures a 24-hour batch of tweeted URLs over a 7-day period during April & May 2019 (M = million).

are categorised into the 4 experiments that each ran for 7 days. When comparing the results in Tables 8.1 and 8.2 (i.e. comparing the 24-hour to 7-day experiment results), Table 8.1 shows that 0.01-0.02% unique unshortened URLs were blocked by Twitter when measured for 24-hours. In comparison, Table 8.2 shows 0.02-0.06% unique unshortened URLs were blocked by Twitter when measured for 7 days. Table 8.1 shows that 6-13% of tweeted URLs residing in GSB were also blocked by Twitter when measured for 24 hours. In comparison, Table 8.2 shows 8-23% of tweeted URLs residing in GSB were also blocked by Twitter when measured for 7 days. This increase suggests that the number of URLs filtered by Twitter and residing in GSB might increase as the duration of time measured since time of tweet increases. This may be due to a delay between tweeted URLs appearing in GSB and Twitter updating its list of URLs to block via *t.co*. This can be further investigated by analysing the number of tweeted URLs blocked by Twitter and residing in GSB during each of the 25 iterations of the experiments.

Figures 8.1a to 8.1c show the total number of unique *t.co* and unique unshortened URLs blocked by Twitter alongside how many tweeted URLs



(a) Exp. #1 (b) Exp. #2

(c) Exp. #3 (d) Exp. #4

Figure 8.1: Experiments #1-4 showing total number of unique *t.co* and unique unshortened URLs blocked by Twitter alongside total number of tweeted URLs residing in GSB blacklist for each experiments' 25 iterations. Each experiment was measured over a 7-day period during April & May 2019. Shown on a logarithmic scale Y-axis.

appeared in the GSB blacklist. These totals are measured during 25 iterations for each of the 4 experiments shown in Table 8.2. A single iteration takes approximately 7 hours to complete and involves analysing all 3 million tweeted URLs from the most recent 24-hour period prior to that experiment starting. These URLs are checked to determine if they are blocked by twitter and also to determine if they appear in one of the 3 blacklists (GSB, OP, and PT). Each experiment (#1-4) measures a batch of 3 million tweeted URLs for 25 iterations – just over 7 days. Figures 8.1a to 8.1c show that Twitter's system to block URLs posted on its network is dynamic – it is actively blocking and unblocking URLs. At some point during the 7 day measurements, all 4 experiments see a temporary spike in the number of URLs blocked by Twitter. This shows that Twitter is actively monitoring its network for websites that breach its rules [278] and blocking such web-

sites once discovered. We can also deduce that Twitter unblocks websites presumably once it is determined they no longer breach the rules.

In experiment #1 there are 4 tweeted URLs residing in GSB during iteration 1 compared to 9 tweeted URLs residing in GSB during iteration 25 (Figure 8.1a). This shows that the number of tweeted URLs residing in GSB has slightly more than doubled during the 7 day period of this experiment. Whereas experiment #2 started with 7 tweeted URLs residing in GSB and finished, at the end of day 7, with 8 tweeted URLs in GSB. The number of tweeted URLs in GSB throughout all 25 iterations of this experiment stayed relatively consistent with a range from 7 at its lowest to 9 at its highest (Figure 8.1b). A similar pattern can also be seen in experiment #3 which started with 8 tweeted URLs residing in GSB on the first iteration and finished on 10 with a range from 5 to 10. Interestingly, the first 4 iterations of this experiment measured the number of tweeted URLs in GSB as 8, 7, 6, and 5, respectively (Figure 8.1c).

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

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

KEY FINDINGS

When comparing our 24-hour to 7-day experiment results, we see a slight increase in the percentage of URLs that were blocked by Twitter. We also see a slight increase in the percentage of URLs that were both blacklisted by GSB ***and*** blocked by Twitter. This may suggest that the number of URLs filtered by Twitter and residing in GSB might increase as the duration of time measured since time of tweet increases –possibly due to blacklist update delay times.

By analysing the iterations of each 7-day experiment, we see that Twitter's URL filtering (via *t.co*) is dynamic; the set of URLs to block-/unblock is actively updated. The number of GSB blacklisted URLs typically increase slightly over time – although in some cases the volume of GSB blacklisted URLs may temporarily increase before returning to its original volume.

Our results in this section show that fewer phishing and malware URLs were posted to Twitter during our experiments in 2018-19 compared to 2017. This suggests that Twitter may have become more effective at blocking blacklisted URLs at time of Tweet. However, on average, only 12% of blacklisted URLs were blocked by Twitter.

8.5 Conclusion

This study measured the effectiveness of Twitter's URL shortener (*t.co*) at protecting users from phishing and malware attacks. In 2017 we analysed over 182 million URL containing tweets collected from Twitter's Stream API over a 2-month period, and compared these URLs against 3 popular social engineering, phishing, and malware blacklists. Our main findings were that over 10,000 blacklisted phishing and malware URLs were posted to Twitter during this timeframe. This lead to over 1.6 million clicks which came directly from Twitter users – therefore exposing people to potentially harmful cyber attacks. We then investigated if Twitter filters URLs posted to its platform. We discovered a reduction in the number of phishing and malware URLs posted to Twitter in 2018-19 compared to 2017 – suggesting an improvement in Twitter's effectiveness at filtering blacklisted URLs at time of tweet. However, we also discovered that only about 12% of blacklisted phishing and malware URLs – which were not blocked at time of tweet and therefore posted to the platform – were blocked by Twitter in 2018-19. Our results indicate that, despite a reduction in the number of blacklisted URLs

posted to the social network, Twitter's URL shortener is not particularly effective at filtering phishing and malware URLs – therefore people are still exposed to these cyber attacks on Twitter.

9
Discussion

OUTLINE
This chapter answers our research questions (defined in Section 1.3) and discusses our research findings (presented in Chapters 5 to 8). We start with a summary of our aims, followed by a discussion about the soundness of measurements; including the challenges of measuring a constantly evolving landscape. This is followed by discussions about each of our research studies, in Sections 9.2 to 9.5.

We then discuss our effectiveness evaluation framework and consider its metrics. This is followed by discussions on how effectively our measurement study findings address our research questions, and considerations for context (provided by existing literature) and other external factors.

We then discuss our machine learning classifier (previously described in Section 4.6.2), Twitter (including motives and data sharing), ground truth, and ethics. We discuss how our research applies to other social media platforms (such as Facebook, Reddit, 4chan, etc). Finally, we discuss recommendations; including ground truth, safe Twitter use, and measurement study methodology.

Our primary aim is to investigate the effectiveness of Twitter's phishing and malware defence system. We want to determine how well-protected Twitter users are from phishing and malware attacks. Twitter has come under increasing pressure to protect its users from cyberattacks. In 2010, Twitter settled a case with the US Federal Trade Commission (FTC) in which Twitter agreed to strengthen security throughout the platform and to carry out an independently assessed bi-annual information security audit for 10 years [86, 87, 88].

We evaluate the effectiveness of Twitter's phishing and malware defence system by defining a set of metrics and framework (Section 2.16: *Effectiveness*), addressing soundness and limitations of existing work (Chapter 3: *Related Literature*), creating **Phish**alytics – our measurement system (Chapter 4: *Design & Implementation*), then conducting numerous measurement studies (Chapters 5 to 8). Twitter's cybersecurity defence system may rely on other technologies to function. Therefore, our effectiveness evaluation includes blacklists (Chapter 5) web browser detection rates, and web browser warnings (Chapter 7).

As part of our investigation we explore the characteristics of popular phishing and malware blacklists (Chapter 5). We investigate the effectiveness of Twitter's cybercrime defence system at *time-of-tweet* to examine the impact of blacklist delay times on user security (Chapter 6). We examine to what extent Twitter might be relying on web browsers' built-in security, and we analyse the effectiveness of web browsers' security (Chapter 7). Finally, we investigate the effectiveness of Twitter's cybercrime defence system at *time-of-click* and compare findings to Twitter's use of blacklists. We also update and strengthen existing measurement studies to improve our understanding of how the landscape we are measuring has changed over time.

Our empirical observations show that over 10,000 blacklisted phishing and malware URLs were publicly shared (i.e. tweeted) on Twitter (Section 6.5.1). During one month, we discovered 4,930 tweets, containing 2,487 total unique URLs, leading to social engineering websites that had been tweeted to over 131 million Twitter users (Section 6.5.3). Blacklisted phishing and malware URLs – tweeted publicly on the social network – received more than 1.6 million clicks from Twitter users (Section 6.5.5). Our URL click data methodology leverages Bitly URLs, which comprises 11% of our dataset. Therefore our URL-click finding will considerably **underestimate** the actual number of URL clicks. We observed that Twitter's *time-of-click* defence system (provided by Twitter's URL shortener: *t.co*) blocked only 12% of blacklisted phishing and malware URLs (Section 8.4.2).

Our findings suggest that Twitter may no-longer be using the GSB blacklist to protect its users – contradicting prior knowledge [200, 201]. We explored the hypothesis that Twitter may be relying on web browsers – most of which have GSB built in (Section 2.6.1) – to protect users from phishing attacks. This may be an effective strategy to protect users from blacklisted websites: our observations show that 81% to 100% of these websites are blocked by web browsers. However, this strategy is not as effective for unknown (i.e. non-blacklisted) phishing websites: our observations show that only 38% to 78% of these website are blocked by web browsers. Our findings suggest more can be done to strengthen Twitter's phishing and malware defence system. We explore our research questions further in

Section 9.7. We address the soundness of our measurements and findings in this chapter by exploring various aspects that may influence our results; such as perspective, context, and external factors.

Our secondary aim is to address soundness and improve measurement studies, thereby strengthening our measurement studies' contribution to the internet measurement research community. We do this by exploring background concepts (Section 3.1), data feeds (Section 3.2), threat intelligence (Section 3.3), context (Sections 3.4 to 3.8), and ethics (Section 3.9) from existing studies. We also identify and addressing limitations of existing work (Section 3.10). Our exploration covers various methodologies which provide inspiration and justify our research methodology.

We aim to improve the quality, quantity, and analysis of data available to the research community for evaluating the effectiveness of Twitter's phishing and malware defence system. We present the details of our methodology (Section 4.7), infrastructure, and technical implementation (Section 4.8) to improve our studies' reproducibility (as defined in Section 2.1.1). We present our results (Chapters 5 to 8) in a format that can be used and interpreted by our intended audience, including, but not limited to: researchers, policymakers, and technology designers. Finally, we aim to contribute to the current understanding of how to collect and analyse internet measurements – specifically relating to cybercrime on Twitter – and to give insight into how the internet behaves. The full codebase for **Phish**alytics is available on GitHub at:

`https://github.com/sjbell/phishalytics`

In this chapter we aim to analyse the findings from our research studies (presented in Chapters 5 to 8) and discuss the wider implications of our research. We begin with a discussion about measurement studies.

9.1 Soundness of Measurement Studies

Results from our measurement studies, along with our effectiveness evaluation framework, may be of particular interest to researchers, policymakers, technology designers, etc. It it important to discuss the soundness of measurement studies, along with strengths and weaknesses. Measurement studies cannot usually provide a *full* picture of a given area. They can, however, provide a reasonable *snapshot* of an ever-changing and evolving landscape. A snapshot that is of great use to society in areas such as benchmarking future snapshots, improving user security, etc.

If you recall our example in 2.1:

> A highly contagious computer virus is spreading across the internet; causing global havoc. The virus's main transmission vector

> is drive-by-download. You want to determine how prevalent the malware is on websites. A measurement study could be conducted whereby all public-facing websites on the internet are checked for the specific malware strain. By visiting all public-facing websites, and checking them for the malware strain, we can start to understand how many websites are spreading the strain and what impact this may have.

In our example, it would be practically impossible to visit every single public-facing website on the internet. There are likely to be other limitations and implementation challenges that may impact the results. Therefore, with these considerations in mind, we understand that our example measurement study may provide a ***reasonable*** snapshot to help us understand the prevalence of said malware at the time the study was carried out.

Measurement study results should be interpreted alongside wider context, including methodology, existing literature, and unmeasured influences. In Chapter 3 we categorised existing literature to explore how various topics may influence our measurement study results. After analysing the results from our measurement studies, we will review our research questions in Section 9.7, and explore context in Section 9.8.

The strengths of measurement studies include providing accurate and scientific data to help guide future research, implement policies, addressing security weaknesses, and other contributions. Measurement studies can provide society with great knowledge; providing we address soundness, and understand how to interpret results and implement the knowledge.

9.1.1 Measuring a Constantly Evolving Landscape

There are a number of challenges that arise when attempting to conduct a longitudinal measurement study. As previously discussed, one of the biggest challenges is trying to measure a constantly changing and evolving landscape. This can make it difficult to compare results from measurements that were carried out at different times. Even a single measurement – such as a week or month in duration – might contain large changes in the landscape. Identifying these changes within the measurements can be complex and time consuming.

For example: hypothetically, Twitter might be used by mainly musicians in 2004 but by 2014 it could be predominantly politicians that use the social network – which could affect measurement results depending on the research aims and questions.

We attempt to address these challenges by using various techniques, such as comparing historical measurement studies in relation to growth (e.g. comparing the number of Twitter users in a 2010 study compared to

the number of Twitter users in a 2017/19 study), and analysing the ratio and percentage of blacklisted URLs in dataset from different time periods.

Whilst we can attempt to compensate for as many of these challenges as possible, there will always be aspects of measuring a constantly changing and evolving landscape that are outside of our control. Some of these changes could be identified through additional measurement studies or research, such as analysing Twitter demographics and how people use the social platform. However, there are still likely to be – at least to some extent – unknown biases in the results.

In our literature review (Chapter 3) we explored some possible contexts surrounding malicious and phishing URLs on Twitter – such as misinformation and its propagation, networks, URL shorteners, etc. These key areas of context might help to understand and explain the bigger picture of phishing and malware URLs on Twitter and to be aware of how, and in what ways, the landscape that we attempt to measure is constantly evolving.

9.2 Blacklist Analysis Study

The discussion in this section relates to our study from Chapter 5: *Blacklist Analysis Study*. In this study we carried out 4 key experiments:

1. Analysis of blacklists: PT, OP, and GSB, to determine number of URLs in each and how their sizes vary over time
2. Measure how long URLs remain in each blacklist for
3. Measure and analyse blacklisted URLs that are removed from then re-added to the same blacklist; timings between reappearance
4. Comparison of URLs between blacklists and detection times of overlapping URLs

Our results show that the average number of URLs contained within each of the 3 blacklist was: 1,581,351 for GSB; 12,433 for PT; and 3,861 for OP. We see that GSB is by far the largest of the 3 blacklists we analysed. Along with social engineering, GSB also contains URLs categorised as malware, unwanted software, and potentially harmful application. These threat types are organised into different platform types (such as Linux, OSX, Windows etc) and the number of URLs within each threat type and platform type varies. However, the number of URLs under each platform type for social engineering URLs always remains the same – because these attacks rely on human presence and are not platform specific. GSB encrypts all URLs in its blacklist which makes it difficult for us to analyse individual URLs - unless a URL is already known to us.

Across all 3 blacklists, as time increases, fewer URLs remain in the blacklists. This is because, once blacklisted, phishing URLs are often short-lived. We discovered that the OP blacklist removes a significant amount of URLs from its dataset after 5 and 7 days; no URLs remained in OP for more than 21 days. This may potentially limit the effectiveness of OP as a blacklist because users may no longer be prevented from visiting an active phishing website once it has been in the blacklist for over 21 days.

Our results show that none of the 3 blacklists in our study enforce a one-time-only URL policy in their datasets. Therefore all 3 blacklists re-add any URLs to their dataset that become a threat again. This is good for users because they will be protected against reoffending phishing websites. We also see that a large number of URLs reappear in the blacklists within 1 day of removal – suggesting that these URLs were either removed too soon or that they came back online again. Without knowing the response status of these websites we do not know specifically why these websites reappeared in the blacklists.

Our results show that 11,603 unique URLs reside in both the PT and OP blacklists. We also see that OP was faster at detecting over 90% of URLs that eventually resided in both blacklists. However, both blacklists have lead and lag time delays of over 65 days. This may limit the effectiveness if just one of these two blacklists is used. OP deploys an automated approach to phishing detection and this likely explains the faster detection rates seen in our results. Conversely, PT employs a manual, community-driven verification voting system to confirm phishing URLs submitted to its dataset and this may explain its lag behind OP.

9.2.1 Website Analysis

One area which we did not explore in this study is the contents of the blacklisted URLs themselves. Analysing page content may have furthered our research in a number of ways, such as determining how long a blacklisted URL remains a threat for. This might help to explain why URLs are removed from blacklists. This sort of analysis can be very time consuming, so would therefore benefit from automated analysis – allowing us to frequently process large numbers of blacklisted URLs. However, this setup would require additional infrastructure to analyse malicious and phishing pages, such as a honeypot. Also, as previously discussed, accurately and automatically detecting phishing websites can be a complex and challenging task. A simple form of analysis which we did consider was capturing the HTML contents of each blacklisted webpage then comparing these contents However, even this simple form of analysis would still require a deeper level of analysis to determine the context of each blacklisted webpage's content. Also, this simple type of analysis may not reveal enough information or

even potentially produce misleading results.

9.2.2 GSB Hash Collisions

All URLs in GSB are encrypted, which means all of our measurements that determine blacklist add and remove timestamps – based on URL matching – will contain a small number of hash collisions. Although our results contain some noise, we still see patterns in the data which are significant beyond the percentage of hash collisions. In Section 4.6.3: *Hash Collisions in GSB* we produced a statistical model to show that our GSB measurement calculations are accurate to within 0.02%.

9.2.3 PT & OP False Positives

Although we show that the OP blacklist detected over 90% of URLs before PT, we did not analyse false positives in either the PT or OP dataset. It may be that the OP blacklist contains false positives due to its automation whereas the PT blacklist contains fewer false positives because of its human-driven verification process.

9.2.4 Future Work

Future work could explore the overlap between PT/OP and GSB blacklisted URLs. As previously discussed, a number of challenges would need to be addressed in relation to the encryption of URLs in GSB. Whilst a one-off comparison would be relatively simple to complete, a more thorough approach that regularly checks the overlap of these blacklists would likely produce more insightful and meaningful results – such as the response times of GSB compared to PT and OP, characteristics of overlapping URLs, etc.

As previously discussed, analysing website content of blacklisted URLs could be a challenging task. However, if done correctly, and with the right approach, it could reveal interesting findings that provide further insight into our results. Future work could also retrieve the status and contents of blacklisted URLs to improve our understanding of the impact of blacklisting on websites. This could allow us to determine if phishing websites are offline when removed from blacklists, among other research questions.

9.3 Time-of-Post Twitter Study

The discussion in this section relates to our study from Chapter 6: *Time-of-Post Twitter Study*.

9.3.1 Twitter Search API

In our study we leverage Twitter's Search API to determine when a blacklisted URL was first tweeted. However, this API is limited to 7-10 days and is not a complete search. This means that the resulting dataset in this experiment is smaller compared to using the Twitter Stream to determine tweet timestamps. Despite these limitations, this methodology provides a more accurate measurement, compared to the experiment where we do not use Twitter's Search API, as it produces the worst case scenario when calculating delay from first tweet to first appearing in blacklist.

9.3.2 Hijacked Websites

One of the weaknesses in our study is what happens when benign websites become malicious due to compromise, change of ownership, adverts appearing on the website from third parties etc. In theory, these websites will become blacklisted but then removed from the blacklist once/if they remove the malicious content. One way this weakness could be mitigated in future work would be by exploring metadata for websites such as WHOIS and also analysing the websites contents to determine if it was benign. However, whilst these websites may indeed have been benign, they have for some reason started to include malicious content which could harm users. Therefore we believe they are still relevant to our study.

9.3.3 Retroactive Blacklist Membership

One of the potential pitfalls of using retroactive blacklist membership detection to determine unsafe URLs is that Google's Safe Browsing uses path prefix expansion. When given a URL to check for blacklist membership the API will iteratively try broader and broader URLs (e.g., x.y.z/a/b/c, x.y.z/a/, x.y.z, y.z). If the entire host of a website becomes blacklisted then this may be misinterpreted as a historical URL being missed when actually it is a fresh incident. One potential way to mitigate this would be to use a modified version of the GSB library that flags if an entire domain has been blacklisted. This would at least allow the resulting measurements to only include URLs whereby the entire domain has not been blacklisted.

9.3.4 Twitter Filter Analysis

During the months of October and November, there were 9 unique URLs that were first tweeted in these months that appeared in the PhishTank and OpenPhish blacklists at some point. Compared to the number of tweeted URLs that appeared in the GSB blacklist this is a significantly smaller number. The PhishTank blacklist typically contains around 5,000

blacklisted URLs and OpenPhish around 23,000 whereas the GSB blacklist typically contains around 3 million URLs. Therefore, it may be that our October and November measurements found more tweeted URLs in the GSB blacklist simply because it contained more blacklisted URLs. However, it is also possible that Twitter is incorporating the OpenPhish and PhishTank URLs into its filtering process; this would explain why we did not see many in our results. Although, if this were true, we would expect to see some tweeted URLs before they appeared in either of these blacklists.

Another possible reason is that Twitter has developed its own heuristics to detect phishing URLs. Therefore, any websites that eventually make it into either the PhishTank or OpenPhish blacklists have already been caught by Twitter's heuristic and prevented from being tweeted. This might explain why we saw fewer matches from these two blacklists compared to GSB. Whilst we could carry experiments to analyse this further, we would essentially be "reverse engineering" Twitter's filtration process. This is something that Twitter knows the answer to and therefore may not be the best use of resources to investigate. It is also hard to devise experiments to test Twitter's filtration methodology without violating Twitter's terms of service, for example: by tweeting known malicious and phishing URLs publicly.

9.3.5 Future Work

Two key areas of future work, which we explored in subsequent studies, involved investigating web browsers built-in phishing protection technology and investigating Twitter's URL shortener, *t.co*, to understand how that impacts malicious URLs posted to Twitter.

Another area for future work involves increasing our ground truth coverage. There are a number of ways in which this might be achieved, such as: leveraging a honeypot to analyse malicious content, increasing the number of blacklists used in the study, producing a machine learning classifier to detect malicious and phishing URLs. Increasing the ground truth coverage in this study might help to detect more URLs (*i.e.* false negatives) – that might have been missed in our original study – and could also help to detect benign URLs that are incorrectly blacklisted (*i.e.* false positives).

9.4 Web Browser Phishing Detection

The discussion in this section relates to our study from Chapter 7: *Web Browser Phishing Detection*. The original aims of this project were:

1. Establish an automated testing environment

   a) Run most popular web browsers (Internet Explorer, Chrome, Safari and Firefox) under the most popular operating systems (Windows, OSX and Linux Ubuntu)
   b) Ability to execute automated phishing detection tests on above web browsers

2. Determine phishing detection capabilities of most popular web browsers against both known (backlisted) and unknown (non blacklisted) phishing websites
3. Reverse engineer web browser phishing detection methodology
4. Can phishing detection be circumvented. If so how?
5. Assess effectiveness of phishing warnings produced by web browsers

9.4.1 Parallel Testing

The automated testing environment was built and able to run the various operating systems and web browsers within it. However, one of the main limitations of the test environment was that only one operating system could run at once. This meant that tests could not run in parallel and many phishing websites may have gone offline or appeared in blacklists during testing. Another limitation of the test environment was the time taken to test a single phishing website and this meant that large numbers of URLs were difficult to test. One way of improving this would be to have a separate virtual machine for each browser being tested. However, the timings used to represent a human would still reduce the total test completion time.

9.4.2 Sample Size

Automated phishing detection tests were carried out on all browsers and results were concluded. One of the main limitations of these tests was the small sample size of phishing websites along with only one source of phishing URLs. A set of results that compared each source of phishing URLs, along with a large number of URLs (e.g. in the thousands) may have produced different detection rates amongst browsers. A greater number of tested phishing websites might also have eliminated the low detection rate seen in Google Chrome under the Windows operating system, or produced more conclusive evidence as to the reason for this lower detection rate.

9.4.3 False Positives & Verification

Another general limitation of tests such as these, that use phishing lists as input, is that some websites might be accidentally flagged by the community as phishing when they are actually legitimate websites. During these tests no

screening was carried out on the phishing websites to determine their true phishiness (although the browser tests were monitored during experiments and all observed websites gave the appearance of a phishing website). This problem could be addressed by taking manual random samples from the phishing websites to check if they are genuine phishing websites.

9.4.4 Circumventing Anti-Phishing Detection

Based on the phishing detection methodology used by Chromium (see 7.4.6: *Chromium Phishing Detection Analysis*, this section briefly outlines a number of theoretical circumvention approaches that could be used to bypass Chromium's phishing detection methods:

- Encourage user to visit phishing website before an actual attack is carried out. This could be achieved by purchasing expired domain names, creating a content scraping website (e.g. by pulling popular news articles from other websites) or by gaining unauthorised access to existing websites. As mentioned at the beginning of this section, many of the phishing websites that were not detected by Chrome, Chromium, Safari and Firefox's non-backlist approach were compromised WordPress sites

- Using a single domain name that is of a similar spelling to a trusted website (e.g. *ebayy.com*) since this would bypass many of the URL structure checks. However, this adds quite an additional cost for phishers since a new domain name will need to be purchased for every phishing campaign and domain misspellings are probably quite rare to find.

- A webpage that uses JavaScript to rewrite the HTML contents and elements of the page could be deployed in order to bypass many of the initial checks. For example: form and script tags could be added dynamically after the page loads, as well as with password input fields. It is not clear if Chromium's phishing detection would detect such alterations after the page has loaded or if a delay in making these alterations to the page (e.g. 5 seconds after the page loads) would affect detection

Whilst some of the above techniques may work in the short term, they only address the limitations of automated phishing detection – and one of the main purposes of automated phishing detection is to address the limitations of blacklists (i.e. update delays). Ultimately, a website that is phishy is likely to be reported and eventually included in a phishing blacklist – at which point circumvention becomes very difficult.

9.4.5 Future Work

The reverse engineering of Chromium's phishing methodology gave an overview of the algorithm and basic features used to make phishing predictions. However, key elements of the algorithm (such as the phishing word list and actual weightings of features) could not be determined by looking at the source code alone. Further experiments could be carried out that attempt to deduce the phishing word list or that try adjusting various phishy features of a webpage in order to deduce the weightings of these features.

Although the phishing detection circumvention techniques were based on Chromium's phishing detection methodology and previous research into automated phishing detection, all proposed solutions were purely theoretical and have not been tested for practicality. Further research into effective phishing detection circumvention methods could be carried out but, as mentioned, these techniques would be short lived once the website is added to a blacklist.

The assessment of phishing warnings produced by web browsers was based on previous studies and no actual testing was carried out on human participants. To produce more conclusive findings, a study could be carried out that involves people attempting to visit phishing websites to see how easily these web browser warnings can be bypassed.

Another limitation of this study is that is only uses phishing URLs. The origins of these URLs have not been explored, although this would merit an entirely different approach of study. Since phishing URLs come from a variety of online sources it would be interesting to study the number of successful phishing attacks compared to phishing URL sources. This would then conclude if certain user groups (such as social media, email, etc) are more likely to be fooled into phishing attacks.

9.5 Time-of-Click Twitter Study

The discussion in this section relates to our study from Chapter 8: *Time-of-Click Twitter Study*. In this study we carried out 2 key experiments:

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

    - 24-hour measurement period
    - 7-day measurement period

Our results from (1) show that, during our 2-month measurement timeframe in 2017, over 10,000 unique phishing and malware URLs were posted to Twitter. The impact of these blacklisted URLs appearing on Twitter was that Twitter users generated over 1.6 million clicks to these potentially harmful websites. Our results from (2) show that, over a 3-month period in 2019, Twitter filtered over 14,000 tweeted URLs via its URL shortener (*t.co*). However, only 2% of these filtered URLs were blacklisted phishing and malware URLs. We see that, during a 24-period, Twitter filtered 6-10% of blacklisted URLs. This increased to 8-30% when measured over a 7-day period. This suggests that Twitter's "time of click" URL filter does improve slightly as time increases, but that a large number of blacklisted phishing and malware websites are not filtered. It is important to note that our 2017 and 2019 datasets were collected using the same sampling strategy.

Our results from these 2 experiments show that, overall, there are fewer phishing and malware URLs being posted to Twitter in 2019 when compared to 2017 and 2010 results. We also see that Twitter adopts both a "time of tweet" and "time of click" defence strategy to protect its users not only from phishing and malware attacks, but also from any user or URL in violation of its set of rules [278]. Twitter is filtering large numbers of tweeted URLs, at time of click, but it appears the majority of these blocked URLs are for violations of its rules other than phishing/malware attacks. Twitter appears to have improved its "time of tweet" defence strategy, since 2010 and 2017, and is perhaps relying on this defence strategy to protect its users from phishing and malware attacks.

9.5.1 Categorising Blocked URLs

In our study we measure the number of *t.co* URLs that have been blocked by Twitter. However, we did not categorise these URLs. Ideally, we wanted to analyse the blocked URLs to determine how many were phishing, malware, etc. Twitter blocks URLs that breach its rules [278], however, this set of rules is quite long and comprehensive – and phishing and malware attacks are only a small part of this comprehensive list of rules. After an initial small, manual, sample of blocked websites, there were concerns about the author's welfare, due to viewing the types of content that had been blocked by Twitter. Therefore the decision was made to not manually inspect blocked websites to determine categorisation.

9.5.2 Inconclusive Hypotheses

One of the main findings of our study is that, since less malware and phishing URLs were tweeted in 2018-19 compared to 2017, Twitter might have become better at catching malicious content. However, this is not

necessarily the case, since it could also be that the trend of malware and phishing decreased, or any number of other reasons. Unfortunately, there is no way to determine which of these hypotheses is correct by looking at the measurements performed in our study due to our very specific measurement focal point. Therefore, as seen in Chapter 3: *Related Literature*, we have tried to explore the context in which phishing and malware URLs might appear on Twitter to improve our understanding of the bigger picture, and where our results fit into the overall picture.

9.5.3 Limitations

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

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

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

9.5.4 Future Work

In future work we would like to explore automated options for the categorisation of blocked URLs. This could include techniques such as leveraging a honeypot to automatically determine if a website is attempting to run or install malicious code on the client machine. Other Another technique might involve automated lexical analysis to determine the content of a webpage to aid categorisation. However, the accuracy of these methods would need to be assessed – which generally goes beyond the scope of this thesis.

9.6 Effectiveness Framework

In this section we discus the results from our measurement studies within the effectiveness framework we defined in Section 2.16.2. We will discuss the results of each metric from our measurement framework.

9.6.1 Time-of-Post

Number of blacklisted phishing/malware URLs posted to Twitter.

In Section 6.5.1 we saw 10,029 unique blacklisted phishing/malware URLs posted to Twitter. In our effectiveness framework, a lower number on this metric contributes to being more effective. In other words: the closer to 0 this number is, the more effective Twitter's phishing and malware defence system may be.

In our results this number is considerably greater than 0 – with a *Very High* effectiveness impact. Therefore, we can say that, for this metric, Twitter's phishing and malware defence system may not be particularly effective.

9.6.2 Time-of-Click

Number of blacklisted phishing/malware URLs blocked by Twitter at time of click.

In Section 8.4.2 we saw that 436 phishing/malware URLs were blacklisted at time of post, but only 48 (11%) were blocked by Twitter at time of click. In our effectiveness framework, the effectiveness contributor for this metric is higher. In other words: the closer to 436 this number is, the more effective Twitter's phishing and malware defence system may be.

In our results, we only see 11% of blacklisted URLs being blocked by twitter – with a *High* effectiveness impact. Therefore, we can say that, for this metric, Twitter's phishing and malware defence system may not be particularly effective.

9.6.3 Blacklist Delay

Delay time from time of tweet to blacklist membership.

In sections 6.5.2, 6.5.3, and 6.5.4 we see delays of of over 20 days in some cases. In the effectiveness framework, the effectiveness contributor is lower. In other words: the closer to 0 this number is, the more effective Twitter's phishing and malware defence system may be.

Our results show large numbers of URLs with blacklist delay times of more than 0 days; this metric has a *Very High* effectiveness impact. Therefore, we can say that, for this metric, Twitter's phishing and malware defence system may not be particularly effective.

9.6.4 Duration in Blacklist

In Section 6.5.7 we see that, as the age of URLs increases, blacklist membership decreases. However, we also see that 1,000 URLs remain in the blacklist for the entire duration of our experiment; 150 days. In Section 5.5.3 we see that the OP blacklist limits the majority of its URLs to a duration of either 5 or 7 days; no URLs remained in OP for more than 21 days. In our framework, the effectiveness contributor for this metric is that it should at least match duration of the URL attack campaign.

In our measurement studies we were unable to attribute specific attack campaigns to blacklist membership durations. Therefore our evaluation of effectiveness is based on the fact that URLs *can* remain in the blacklists for as long as required. We see that the OP blacklist limits membership durations. This metric has a *Very High* effectiveness impact, therefore the effectiveness of OP could be limited due to its duration limit.

9.6.5 User Views

Number of potential Twitter user views (of blacklisted phishing/malware URLs).

In Section 6.5.3 we see that 2,487 unique blacklisted phishing/malware URLs were retweeted by 1,227 Twitter accounts with a combined number of 131,116,820 followers. In the effectiveness framework, the effectiveness contributor is lower. In other words: the closer to 0 this number is, the more effective Twitter's phishing and malware defence system may be.

Our results show the number of potential Twitter user views was significantly high (over 131 million); this metric has a *High* effectiveness impact. Therefore, we can say that, for this metric, Twitter's phishing and malware defence system may not be particularly effective.

9.6.6 Number of Clicks

In Section 6.5.5 we see that blacklisted phishing/malware URLs posted to Twitter received a combined total of 1,052,152 clicks directly from Twitter. In the effectiveness framework, the effectiveness contributor is lower. In other words: the closer to 0 this number is, the more effective Twitter's phishing and malware defence system may be.

Our results show that blacklisted phishing/malware URLs posted to Twitter received a significant volume of clicks from Twitter (over 1 million); this metric has a *Very High* effectiveness impact. Therefore, we can say that, for this metric, Twitter's phishing and malware defence system may not be particularly effective.

9.6.7 Web Browser Phishing Detection: Known URLs

Number of known blacklisted phishing URLs blocked by the web browser.

In Section 7.4.1 we see that 81-100% of known blacklisted phishing URLs are detected by web browsers. In the effectiveness framework, the effectiveness contributor is higher. In other words: the closer to 100% this number is, the more effective Twitter's phishing and malware defence system may be.

Our results show this metric reports 80-100%. However, the metric has a *Low* effectiveness impact because this metric indicates the user has left Twitter; therefore Twitter has failed to protect the user effectively. We now have to rely on the web browser's effectiveness to defend the user against the attack. Therefore we can say that, for this metric, the web browser has contributed to effectiveness – but this will likely be negated by the fact that Twitter failed to protect the user from this attack.

9.6.8 Web Browser Phishing Detection: Unknown URLs

Number of unknown blacklisted phishing/malware URLs blocked by the web browser.

In Section 7.4.2 we see that 38-78% of unknown blacklisted phishing URLs are detected by web browsers. In the effectiveness framework, the effectiveness contributor is higher. In other words: the closer to 100% this number is, the more effective Twitter's phishing and malware defence system may be.

Our results show this metric reports 38-78%. However, this metric has a *Medium* effectiveness impact because this metric indicates the user has left Twitter; therefore Twitter has failed to protect the user effectively. We must now rely on the web browser's effectiveness to defend the user against the attack. Therefore we can say that, for this metric, the web browser has contributed to effectiveness – but this will likely be negated by the fact that Twitter failed to protect the user from this attack.

9.6.9 Accuracy of Ground Truth

E.g. number of false positives / negatives.

We did not directly measure the accuracy of ground truth (i.e. blacklists) in our study. Section 3.3 explored existing literature that attempt to assess the accuracy of the blacklists we use. See saw that blacklists can provide a reliable defence from, and source of ground truth for, cyber threats. However, blacklist delays need to be considered. We also saw that blacklists of the same category can contain a significant overlap which helps to achieve a wide coverage. However, compiling a complete and comprehensive ground truth is a challenging and difficult task. Therefore most organisations will make trade-offs to achieve a good enough ground truth that provides effective protection for their needs.

Our measurement studies use blacklists that specialise in phishing attacks – which is a key threat that we are researching – and provide effective coverage for our requirements. Therefore, based on existing literature, our ground truth should be accurate. In the effectiveness framework, the effectiveness contributor is "high accuracy". In other words: the more accurate the ground truth is, the more effective the ground truth may be; this metric has a *Very High* effectiveness impact. Therefore we can say that, for this metric, based on existing literature, the 3 blacklists we study may provide effective protection against phishing/malware attacks.

Our results in Chapter 6 show that Twitter may not be using these 3 blacklists particularly well. Therefore, this may reduce the effectiveness of Twitter's phishing and malware defence system.

9.6.10 Blacklist Speciality

E.g dedicated phishing blacklist.

In Section 5.5.1 we see that the 3 blacklist we analyse (GSB, PT, and OP) all specialise in either phishing or malware URLs. In our effectiveness framework, the effectiveness contributor is "high speciality". In other words: the more specialised the blacklist is, the more effective the blacklist may be. This metric has a *Very High* effectiveness impact.

However, as with the previous metric, our results in Chapter 6 show that Twitter may not be using these blacklists particularly well, therefore this may reduce the effectiveness of Twitter's phishing and malware defence system.

9.6.11 Blacklist Size

E.g. number of URLs in blacklist.

In Section 5.5.1 we see that the average size of the 3 blacklist we analyse (in terms of number of URLs) are GSB: 1.5 million, PT: 12,433, and OP: 3,861.

In our effectiveness framework, the effectiveness contributor is "high". In other words: the larger the size of the blacklist is, the more effective the blacklist may be. This metric has a *Medium* effectiveness impact. This may help to explain why our results show that GSB detected more blacklisted phishing/malware URLs compared to PT and OP.

As with the previous metric, our results in Chapter 6 show that Twitter may not be using these blacklists particularly well, therefore this may reduce the effectiveness of Twitter's phishing and malware defence system.

9.6.12 Blacklist Comprehensiveness

I.e. comprehensiveness of cover.

Difficult to access and impossible to practically implement 100% [223], [149], [148], [310], [183]). In Section 5.5.2 we see an overview of the categories within GSB, and how many URLs reside in each category. This analysis may gave in indication to the comprehensiveness of GSB. However, based on this data alone we cannot draw a full conclusion as to the comprehensiveness of GSB, PT, or OP. In our effectiveness framework, the effectiveness contributor is "higher comprehensiveness". In other words: the more comprehensive the blacklist is, the more effective the blacklist may be. This metric has a *Medium* effectiveness impact.

However, as with the previous metric, our results in Chapter 6 show that Twitter may not be using these blacklists particularly well, therefore this may reduce the effectiveness of Twitter's phishing and malware defence system.

9.6.13 Blacklist Intersection

Intersection of blacklists.

In Section 5.5.5 we see that 11,603 URLs exist in both the PT and OP blacklists. In our effectiveness framework, the effectiveness contributor is "lower". In other words: the fewer URLs that intersect multiple blacklists, the more effective the blacklist may be. This metric has a *Low* effectiveness impact. This suggests that using both the PT and OP blacklists may not provide full coverage. Effectiveness may be increased by including an additional blacklist, such as GSB.

As with the previous metric, our results in Chapter 6 show that Twitter may not be using these blacklists particularly well, therefore this may reduce the effectiveness of Twitter's phishing and malware defence system.

9.6.14 Blacklist Update Frequency

Including client endpoints (e.g. GSB in web browsers updates at lower frequency to GSB online blacklist).

During our experiments (see Section 4.8.6) we observed that GSB updates approximately every 10 minutes; PT and OP update every hour. In our effectiveness framework, the effectiveness contributor is "higher". In other words: the more frequently the blacklist updates, the more effective the blacklist may be. This metric has a *High* effectiveness impact. This suggests that the GSB blacklist may be more effective than PT and OP because GSB will update 6 times during the time that both OP and PT update only once.

As with the previous metric, our results in Chapter 6 show that Twitter may not be using these blacklists particularly well, therefore this may reduce the effectiveness of Twitter's phishing and malware defence system.

9.6.15 Benchmarking

If possible, use existing studies to benchmark "effectiveness" results (e.g. compare a 2010 study to a 2020 study).

Needs to be in context of Twitter user numbers at time of measurement. For example: 30 million total tweets per day of which 3 million phishing in 2010 (10%) compared to 300 million tweets per day of which 3 million phishing in 2020 (1%) = same volume of phishing tweets but different ratio.

Our results in Section 6.5.2 show that in 2017 a greater number of blacklisted phishing URLs were tweeted compared to a 2010 study. In addition to this, our results in 2017 show fewer numbers of blacklisted malware URLs tweeted compared to a 2010 study. Our 2018-19 results 8.4.2 show that fewer blacklisted phishing and malware URLs were posted to Twitter compared to our 2017 results.

In our effectiveness framework, the effectiveness contributor is "improvement on previous studies' results". In other words: the greater the improvement on previous studies' results, the more effective Twitter's phishing and malware defence system may be. This metric has a *Very Low* impact score. This is because, due to the previously mentioned limitations of measurement studies, a direct comparison between results may not be entirely accurate.

When comparing previous results, we see that the effectiveness of Twitter's **phishing** defence system may have **decreased** from 2010 to 2017. The effectiveness of Twitter's **malware** defence system may have **increased** from 2010 to 2017. Finally, the effectiveness of Twitter's **phishing and**

malware defence system may have **increased** from 2017 to 2018-19.

9.6.16 Evaluation

As we have discussed, our framework provides a guide that contributes towards evaluating the effectiveness of Twitter's phishing and malware defence system through various measurements. However, as previously discussed in Sections 2.1 and 3.8, we must consider soundness, be careful not to make absolute claims, or draw conclusions from dichotomous thinking and measurement fallacies. Results must be understood in context, alongside relevant background topics and existing literature.

We could say that, in the absence of additional information, the results obtained in chapters 5 to 8, when evaluated inline with our measurement framework, suggest that Twitter may not be particularly effective at protecting its users from phishing and malware attacks. However, our measurements are taken from specific observation points – therefore other factors may contribute towards Twitter's effectiveness that we did not measure during our studies (e.g. real world events).

In Chapter 3 we summarised existing literature that may provide context to our studies (which we explore further in Section 9.8). For example, we saw how phishing and malware attacks may propagate via URL shorteners (Section 3.8.1) or misinformation (see 3.8.2) (e.g. via trending hashtags).

Phishing / malware tweets that propagate via certain trending hashtags could be more effective at bypassing Twitter's phishing and malware defence system – e.g due to commercial pressures on said hashtags linked to Twitter's advertising platform. Or a respected and trusted Twitter account may become hijacked, resulting in malicious tweets sent to unsuspecting users; bypassing Twitter's defences that rely on account reputation.

These are hypothetical examples to illustrate the myriad of potential events that *may* occur outside the observation points of our measurement studies. Therefore, our effectiveness framework should be used as an approximate guide.

When evaluating effectiveness it is also important to consider the risk appetite (see Section 2.11 for definition) of individual Twitter users (including organisations and businesses). Our evaluation could conclude that Twitter is *not* effective at protecting users from phishing and malware attacks. However, Twitter users with high risk appetites may interpret evaluation differently, concluding that Twitter *is* effective at protecting its users from phishing and malware attacks.

Our framework features effectiveness *contributors* and *impacts* for the metrics we measure. These could pivot, allowing organisations to customise our framework to reflect their risk appetite. For example: an organisation

with a **high risk appetite** might **reduce** *effectiveness impacts* to between *Medium* and *Very Low*. Whereas an organisation with a **low risk appetite** might **increase** *effectiveness impacts* to between *Medium* and *Very High*.

In summary, we address the soundness of our measurement study conclusions. Our results must be interpreted alongside context to avoid dichotomous thinking and oversimplification. Evaluating the effectiveness of Twitter's phishing and malware defence system is complex. The risk appetite, and risk awareness, of Twitter users should also be considered when evaluating effectiveness. Our framework and analysis contributes towards evaluating effectiveness but cannot provide a complete picture on its own.

9.7 Research Questions

Our research questions, that we defined in Section 1.3, are:

1. How effective is Twitter at protecting its users from phishing and malware URL attacks?
2. How effective are blacklists at helping Twitter to protect its users from phishing and malware attacks?
3. What do popular phishing and malware blacklists consist of? Uptake, dropout, typical lifetimes, and overlap of URLs in these blacklists
4. What is the lifetime of a URL in a phishing/malware blacklist?
5. Are blacklists affected by delays – and, if so, what impact do blacklist delays have on user security?
6. How effective is Twitter's URL shortener, *t.co*, at protecting Twitter users from phishing and malware attacks?
7. What is the comparison between Twitter's use of blacklists and Twitter's URL blocking (via *t.co*) at protecting users from phishing and malware attacks?
8. To explore the impact of phishing and malware attacks on Twitter: how many Twitter users click on publicly tweeted phishing or malware URLs that have been blacklisted?

Based on our findings in Chapters 5 to 8 we shall now address each research question (RQ) in the following subsections.

9.7.1 RQ 1

How effective is Twitter at protecting its users from phishing and malware URL attacks?

Our results (Section 6.5) show that Twitter does protect its users against some phishing and malware attacks. However, we observed over 10,000 blacklisted phishing and malware URLs that had been publicly shared on the social network (Section 6.5.1). Additionally, during one month, we discovered 4,930 tweets containing URLs leading to social engineering websites that had been tweeted to over 131 million Twitter users (Section 6.5.3). Therefore potentially exposing Twitter users to dangerous attacks.

Our findings may be influenced by many factors, such as hijacked websites (Section 9.3.2) and retroactive blacklist membership (Section 9.3.3). Therefore, our results are not absolute (i.e. not independent; require context), but would suggest that Twitter may not be effective at protecting its users from phishing and malware URL attacks – and, therefore, could do more to protect its users against cyber attacks.

As we discussed in Section 9.6.16, when evaluating effectiveness, it is important to consider the risk appetite, and risk awareness, of individual Twitter users (including organisations and businesses). Determining how effectively Twitter protects its users from phishing and malware attacks may very well depend on the risk appetite of those users.

9.7.2 RQ 2

How effective are blacklists at helping Twitter to protect its users from phishing and malware attacks?

Twitter does not appear to be effectively leveraging the GSB blacklist to protect users against phishing and malware threats (Section 6.5.1). Contextually, Twitter may be relying on web browsers built-in security to protect users from phishing attacks. However, our results show that whilst web browsers detect at least 81% of know (i.e. blacklisted) phishing websites (Section 7.4.1), web browsers only detect 38%-78% of unknown (i.e. non-blacklisted) phishing websites (Section 7.4.2). Therefore, in the worst case scenario, Twitter users could be exposed to 62% of phishing attacks.

Whilst our study observed blacklisted URLs posted to Twitter, our ground truth was limited to 3 blacklists (GSB, OP, and PT). Therefore, there may be phishing and malware URLs posted to Twitter that we did not detect. We are also unable to measure the number of phishing and malware URLs that Twitter blocked at time of tweet – only those that Twitter does not block. Twitter may have a sophisticated detection system that detected – and blocked – significantly more than the 10,000 URLs we observe that bypassed Twitter's defence systems (Section 9.3.4). Therefore, considering these contextual implications, our results could suggest that blacklists may not be effective at helping Twitter to protect its users from phishing and malware attacks.

9.7.3 RQ 3 & 4

What do popular phishing and malware blacklists consist of? Uptake, dropout, typical lifetimes, and overlap of URLs in these blacklists.

What is the lifetime of a URL in a phishing/malware blacklist?

Our analysis of popular phishing and malware blacklists (GSB, OP, and PT) observed an average of 1.6 million URLs in the GSB blacklist, over 12 thousand URLs in PT, and nearly 4 thousand URLs in OP (Section 5.5.1). The OP blacklist appears to enforce strict limits on how long URLs can remain in its dataset for (Section 5.5.3). The PT and OP blacklists see a 12% overlap of URLs; OP detected over 90% of these URLs before PT (Section 5.5.5). On average, URLs remain in GSB for 10 days, PT for 2 days, and OP for 5 days. However, we also see that some URLs remain in blacklists for the entire duration of our measurement experiments (Section 5.5.3).

Our findings may be influenced by hash collisions in the GSB blacklist; whereby more than 1 URL may share the same hash prefix (Section 9.2.2). This consideration would impact the GSB figures in our study. Contextually, our study did not explore the contents of blacklisted websites, nor false positives. Existing research (Section 3.3) addresses the accuracy of the blacklist we study, and there is scope for future work (Section 9.2.4). Wider implications of our research may contribute towards future work by improving our understanding of the blacklists we study. This may help future research, and organisations, for example: when choosing a suitable blacklist (See 9.17: *Recommendations*).

9.7.4 RQ 5

Are blacklists affected by delays – and, if so, what impact do blacklist delays have on user security?

Blacklist delays can have a significant impact on user security. Although the majority of URLs in our study were detected by the GSB blacklist within 6 hours of being tweeted, a significant number of URLs took over 20 days to appear in GSB (Sections 6.5.4 to 6.5.2). Blacklist delays can also affect web browsers' ability to defend against such attacks; reducing detection rates from between 81% and 100% (Section 7.4.1) down to between 38% and 78% (Section 7.4.2).

Our results show that blacklists are indeed affected by delays. Contextually, numerous factors may influence our results, such as hijacked websites (Section 9.3.2); whereby a benign websites later becomes hijacked, impacting our delay measurement. The impact of blacklist delays on user

security include creating an attack space for cybercriminals. The impact of delays could be mitigated by incorporating a machine learning classifier to detect phishing and malware websites before they are blacklisted. However, factors such as accuracy and commercial liabilities should be considered.

9.7.5 RQ 6 & 7

How effective is Twitter's URL shortener, t.co, at protecting Twitter users from phishing and malware attacks?

What is the comparison between Twitter's use of blacklists and Twitter's URL blocking (via t.co) at protecting users from phishing and malware attacks?

Our results show that Twitter does block certain URLs at time of click (via its URL shortener, *t.co*), therefore providing additional protection to Twitter users. However, we observed that only 12% of blacklisted phishing and malware URLs were blocked by Twitter at time of click (Section 8.4.2) – therefore potentially exposing people to these threats.

Our results suggest that Twitter's URL shortener (t.co) is not particularly effective at protecting Twitter users from phishing and malware attacks at time of click. However, as we have already seen with our previous research questions, context is important. Our ground truth for classifying phishing and malware URLs is the 3 blacklists (GSB, OP, and PT); our measurements do not include tweeted phishing or malware URLs that are not blacklisted. Also, we did not consider the contents of blacklisted website in our measurements – which could improve website threat classification.

Wider implications of our results suggest that Twitter users should be extremely cautious when clicking tweeted links to external websites. Even trusted Twitter account can become hijacked and send dangerous URLs to unsuspecting Twitter users. Twitter users should ensure that the devices they access Twitter on (laptops, phones, tablets, etc) include adequate protection against phishing and malware attacks that match the level of risk they are comfortable with (See 9.17: *Recommendations*).

9.7.6 RQ 8

To explore the impact of phishing and malware attacks on Twitter: how many Twitter users click on publicly tweeted phishing or malware URLs that have been blacklisted?

Our study observed blacklisted phishing and malware URLs – tweeted publicly on the social network – received more than 1.6 million clicks from Twitter users (Section 6.5.5). This finding highlights the implication of

ineffective phishing and malware protection: people exposed to potentially harmful cyber attacks.

As with many of our previous research question outcomes, it is important to understand context and wider implications of this finding. Our results may be influenced by many factors, such as website hijackings, Twitter user account hijackings (Section 9.3.2) or retroactive blacklist membership (Section 9.3.3). For example: our methodology to determine phishing and malware click data on Twitter consists of retrieving Bitly click data for blacklisted URLs. Therefore, if a benign website receives a certain volume of clicks from Twitter, but later becomes blacklisted, then clicks during the benign timeframe may be included. We try to mitigate this as much as possible by specifying timeframes to the Bitly API that match URL blacklisting timestamps.

Another consideration of our findings is what additional protection Twitter users may, or may not, have after they click a blacklisted URL – therefore leaving Twitter's platform. In previous research questions we investigated Twitter's phishing and malware defence system at time of click (via Twitter's URL shortener, t.co) and we reviewed phishing defence systems of web browsers. As we previously mentioned in Section 2.14, users may have additional protection from anti-virus software, web browser plug-ins, network-level defences, etc. These considerations all address the same question: what additional technologies are in place to address weaknesses in Twitter's defence systems. In other words: how can we protect users from phishing and malware attacks that Twitter missed?

9.7.7 Evaluation

There are various aspects to consider when assessing how well we have answered our research questions. As previously discussed, it is important to consider the difficulty of measuring a constantly changing landscape. Our results in one timeframe may be affected by different aspects to those in another. Additionally, the risk appetite of Twitter users may influence the evaluation of how effectively Twitter protects said users from phishing and malware attacks.

Another consideration is that our measurements offer a very specific *window* into phishing and malware activity on Twitter. We have not explored the wider context of these malicious attacks on Twitter, such as their origins, victims, or analysed the attacks themselves. Our ground truth is consists of 3 blacklists that mostly specialise in phishing attacks. Therefore our research questions focus on these types of cyber attacks.

With these considerations in mind it does create scope for future work – and existing literature – to create a more complete picture of our results. We explored some key existing literature in Chapter 3: *Related Literature*.

Our findings improve our understanding of how effective Twitter's phishing and malware defence system is. Our results show that Twitter does block *some* known blacklisted URLs at time of tweet and *some* known blacklisted URLs at time of click. We saw how Twitter's URL shortener (*t.co*) *can* protect Twitter users from dangerous URLs at time of click – and how this defence mechanism compares to blacklists.

However, we observed over 10,000 phishing and malware URLs – publicly broadcast to over 131 million Twitter users – received over 1.6 million clicks directly from Twitter users. Suggesting there is considerable room for improvement in Twitter's defence system. More must be done to protect Twitter users from phishing and malware attacks.

9.8 Context

In Section 1.9 we explored context of our research topic; such as legislation, phishing and malware attack statistics, and web browser usage figures. In this section we built on that foundation and provide context to our research findings by exploring existing literature (from Chapter 3).

Exploring context can address soundness, enhance our perspective, strengthen our results, and improve our understanding of wider implications. We illustrate this by exploring context from 3 topics: URL shorteners (Section 9.8.1), information credibility (Section 9.8.2), and networks (Section 9.8.3). Our exploration of these 3 topics of context is by no means exhaustive. Other contexts that could influence our findings include real-world events such as major sporting events, global pandemics, and political events. These real-world events may cause a surge in related Tweets and hashtags, which is likely to attract malicious actors. This may influence our measurement results if we did not consider the impact of such events.

9.8.1 URL Shorteners

Antoniades *et al.* (2011) [16] concluded that phishers use the Bitly URL shortener not only for reducing space but also to hide their identity. The implications for the context provided by [16] are that the click figures from our measurement studies could be biased. This is because [16] suggest a higher proportion of phishing URLs posted to Twitter are shortened via Bitly, and we use Bitly as our click data ground truth. Therefore we may be using a biased ground truth. However, the study was conducted [16] in 2011 and has not been repeated since. Twitter introduced its own URL shortener after the [16] study, in 2011, which may potentially reduce reliance and trust in Bitly. Although our studies did not measure user trust in *t.co* or *bit.ly*, it is possible that user trust may have shifted.

Nikiforakis *et al.* (2014) [211] investigated the ecosystem of ad-based URL shortening services from a security and privacy perspective. The study discovered that ad-based URL shorteners are susceptible to a number of security vulnerabilities, such as tabnabbing and link hijacking which can lead to drive-by-downloads, browser-exploits, scams, phishing attacks; and privacy concerns through the use of sequential (i.e. predictable) URLs and URL leaking through HTTP Referrer headers.

Many of the vulnerabilities and privacy concerns identified in [211] may not be detected by blacklists or web browsers' heuristic detection. This means our measurement studies might also not have detected the vulnerabilities and privacy concerns identified in [211]. The implications of [211] on our research findings are that our measurement results are likely to *underestimate* the number of phishing and malware attacks on Twitter; therefore reducing the effectiveness Twitter's defence system.

Finally, results from Maggi *et al.* (2013) [171] show that countermeasures to prevent users from shortening phishing and malware URLs are ineffective and trivial to bypass. This provides further justification for our methodology of checking all hops in tweeted URL redirection chains for blacklist membership. Contextually, it shows that Twitter users should be extremely careful when clicking shortened links on Twitter because the defence systems of URL shorteners may not be particularly effective.

9.8.2 Information Credibility

Castillo *et al.* (2011) [53] discovered that over 90% of trending topics in their dataset were not "newsworthy" (i.e. statements about a fact or actual event). Building on this, Gupta *et al.* (2012) [112] discovered that only 17% of tweets about high impact news events contained credible information. Ghosh *et al.* (2013) [96] discovered that tweets from experts are more trustworthy; containing considerably fewer malicious URLs and spam.

Findings from existing studies illustrate the low ratio of credible information on Twitter. Additionally, due its open platform whereby anybody can broadcast a message, Twitter attracts users that want to spread misinformation. Therefore it can be difficult for general Twitter users to distinguish "fake news" from credible information.

Misinformation may exploit emotional responses from Twitter users (e.g. by leveraging specific trending topics). Being in a heightened emotional state can impact rational decision-making [160, 61, 272] thereby potentially leading Twitter users into an attack. Misinformation, whilst being a danger in and of itself, may also contain phishing and malware attacks. If a person's judgement is reduced, they could be more susceptible to an attack.

Results from our measurement studies show large volumes of blacklisted phishing and malware URLs were tweeted publicly. Therefore it is vital that Twitter highlights credible information on its platform – and flags misinformation – to protect users from phishing and malware attacks.

9.8.3 Networks

Yang *et al.* (2012) [305] investigated cyber criminal ecosystems on Twitter; discovering how criminals are socially connected and how URLs are promoted by their large social groups. Ghosh *et al.* (2012) [95] investigated link farming on Twitter; whereby users acquire large numbers of followers to increase their audience, perceived influence, and ranking. Stringhini *et al.* (2013) [263] investigated Twitter follower markets; whereby users of the social network can purchase followers.

The results in [305, 95, 263] highlight tactics used by cybercriminals to masquerade as legitimate and more trustworthy. These existing studies provide context to our measurement studies by showing how Twitter users may be exposed to phishing or malware URLs.

Our results show that over 10,000 blacklisted phishing and malware URLs were tweeted during our study (Section 6.5.1). However, our measurements do not observe *how* those tweeted URLs propagated – or what promotion techniques cybercriminals used. It may be that all 10,000 blacklisted URLs were shared by independent Twitter users; not part of large criminal networks. However, the opposite may also be true. During one month, our measurement study discovered 2,487 blacklisted phishing URLs had been tweeted by 1,227 individual Twitter accounts (in 4,930 tweets) to over 131 million Twitter users (Section 6.5.3). It could be that those 1,227 Twitter accounts were part of criminal networks. However, they could also be independent accounts.

9.9 Machine Learning Classifier to Detect Fresh Phish

As discussed in Section 4.6.2, we designed a machine learning (ML) classifier to detect tweets that contain suspected phishing URLs. The purpose of the ML classifier was to determine blacklist delays to conduct a lifecycle analysis study. Our methodology involved using the ML classifier to automatically detect tweets that contained phishing URLs. These URLs would then be frequently checked for blacklist membership; delay times calculated. The ML classifier produced results that contradicted prior knowledge that blacklisted URLs could not be posted to Twitter. Therefore, the rest of our

time was dedicated to conducting measurement studies to investigate this finding. Because of this our ML work was not taken further.

Future research could expand our ML work by integrating the classifier into **Phish**alytics. The classified "fresh" phishing tweets would mark time zero in a blacklist lifecycle analysis measurement. This would improve the accuracy of our metric for "time of first tweet" for two reasons. Firstly, phishing tweets could be categorised into blacklisted and non-blacklisted at time of tweet. Therefore improving the quality of data that shows how many blacklisted URLs are tweeted. Secondly, our measurement studies used retroactive blacklist membership for phishing detection – which relied on GSB URL hash prefix lookup methodology. The URL hash prefix lookup methodology was prone to hash prefix collisions (see Section 4.6.3) therefore reducing the size of our usable dataset. By leveraging a source of "fresh" phishing URLs (i.e. pre blacklist membership URLs) from a ML classifier, the resulting measurements would contain a more accurate and richer dataset because GSB membership could be detected in real time. Therefore eliminating the GSB hash collision issue.

Future research could also improve the accuracy of the ML classifier. We improved the classifier's overall accuracy from 92.52% [4] to 99% by adding new features and altering the ratio of benign to phishing tweets in the training data. However, although we increased the detection rate of benign classification from 94.41% [4] to 99%, the trade-off was that the detection rate for phishing classification fell from 92.31% [4] to 28% – therefore missing 72% of phishing tweets. However, it is important to consider the application of the ML classifier. In a typical 24-hour period (approximately 3 million tweets; 1 million unique URLs – see GSB observation in Section 4.8.6) there will be approximately 139 phishing tweets and 999,861 benign tweets. Although our classifier will only detect 39 (and miss 100) phishing tweets, it will correctly classify 999,845 (and miss only 16) benign tweets. Whereas [4] will detect 128 (and miss 11) phishing tweets, it will **incorrectly** classify 95,987 benign tweets as phishing. Therefore, although [4] will be more effective at protecting the user against phishing attacks, it comes at the cost of a high false negative (or miss) rate. If the application of the ML classifier's aim is solely to protect the user from phishing attacks, then it succeeds. However, if false positives are going to present a problem (for example, commercial liabilities such as in consumer products) then there is still work to be done in improving the accuracy of the ML classifier.

9.10 Twitter

This section discusses topics about Twitter. We will address Twitter's phishing & malware detection, motives, data sharing, and sample stream size.

9.10.1 Twitter's Phishing & Malware Detection

One of the key takeaways from our research is that Twitter may not be effectively using the blacklists we analysed (GSB, OP, and PT). However, it is possible that Twitter's phishing and malware detection is much worse than the 3 blacklists combined – but it is also possible that Twitter's detection is much better, or that it simply detects different content. Ultimately, we cannot say for sure based on our specific measurement perspective.

9.10.2 Twitter's Motives

One of the key research questions we are trying to answer in this thesis is: how effective is Twitter's phishing and malware defence system at protecting its users. As we have seen from our research studies, there appear to be limitations of Twitter's defence system and potential room for improvement – which may be leaving Twitter users exposed to such attacks.

However, it is important to understand the context of Twitter's motives. Twitter, as with all other responsible social media platforms, is under constant pressure to take action over users that post malicious content to its platform; this often involves the removal of offending accounts. However, Twitter also needs to demonstrate its continued growth.

On the 7th November 2013, Twitter became a public company, listing its shares on the New York Stock Exchange (NYSE). On that day, Twitter's shares opened at $26.00 and closed at $44.90, giving the company a valuation of around $31 billion [35]. On the 5th February 2014, Twitter published its first results as a public company, showing a net loss of $511 million in the fourth quarter of 2013 [73]. Since then, Twitter has been under constant pressure from investors to increase its net worth. On the 26th July 2019 it was reported [135] that Twitter had started to hide its monthly usage numbers and that the number of Twitter users had spent a year in decline [134]. During that year, Twitter removed bots, spam, and other bad user accounts from its platform – which may have partly driven Twitter's move to obscure its user numbers. Bear in mind: one of the key ways in which Twitter achieves financial growth is through advertising revenue – which is directly impacted by its number of users. On the 6th February 2020, Twitter reported fourth quarter revenues of $1.01bn for the first time ever [173], increasing its shares by 8%.

Twitter must strike a balance between publicising its user base growth, taking action against malicious users, and how this impacts its shareholders and net worth. This becomes a fine balancing act for Twitter to keep multiple parties happy at the same time.

9.10.3 Twitter's Data Sharing

Twitter's sharing of its data has a positive impact on the research and academic community. As we have seen, numerous studies have benefited from Twitter's open data. However, Twitter is also very careful with its full data set, charging a premium for access. As previously mentioned, we contacted Twitter directly and were given a quote for accessing their Decahose (10%) Feed (Twitter provides 3 feeds: free, approximately 1%; 10%; and 100%) – but this quote was considerably beyond what academic researchers – and probably most businesses – can afford. This means that Twitter has placed its full, and even 10%, dataset out of reach of the majority of academics. As a result, Twitter could say, in response to any academic study: *we accept that this study is interesting, but the results might not be accurate because the study only has access to 1% of our data*. Therefore giving Twitter something of a layer of protection against research carried out on its data. But this raises an important question: which organisations ***can*** afford access to Twitter's full, 100%, dataset? And, for those organisations that can afford access to Twitter's 100% dataset, in what ways are these organisations using the dataset to make access to it profitable?

January 2021 Update

On the 21st of January 2021, Twitter announced[1] the launch of its *Academic Research* product track; part of their new v2 API. In their announcement, Twitter state:

> *"Our developer platform hasn't always made it easy for researchers to access the data they need, and many have had to rely on their own resourcefulness to find the right information. Despite this, for over a decade, academic researchers have used Twitter data for discoveries and innovations that help make the world a better place."*

Twitter state that their new *Academic Research* product track includes:

[1] https://blog.twitter.com/developer/en_us/topics/tools/2021/enabling-the-future-of-academic-research-with-the-twitter-api.html

- *"Higher levels of access to the Twitter developer platform for free, including a significantly higher monthly Tweet volume cap of 10 million (20x higher than what's available on the Standard product track today)"*
- *"Free access to full-archive search"*
- *"Shortened URLs are fully unwound for easier URL analysis"*

Despite claims of "higher monthly Tweet volume" in their v2 API, Twitter have introduced limitations compared to their previous v1.1 API. For example: our measurement studies had access to Twitter's v1.1 API. We received over 91 million monthly tweets via Twitter's Filter API and over 105 million monthly tweets on Twitter's Sample API (section 6.5.1: *Twitter Dataset Analysis*). That is, Twitter's old v1.1 API produced over 8x more Tweets than Twitter's new *Academic Research* product track – which itself claims to offer 20x more tweets then their new *Standard* track. Suggesting the new v2 API severely restricts tweet data volumes compared to the previous v1.1 API. This raises questions over the true volume of data that Twitter is sharing with academics. However, Twitter's v2 API feature of allowing academics free access to the "full-archive search" is a useful addition that would have improved our research.

Twitter' new Academic Research access requires a more stringent application process:

> *"We require this additional application step to help protect the security and privacy of people who use Twitter and our developer platform. Each application will go through a manual review process to determine whether the described use cases for accessing our Academic Research product track adhere to our Developer Policy, and that applicants meet these three requirements: you are either a master's student, doctoral candidate, post-doc, faculty, or research-focused employee at an academic institution or university; you have a clearly defined research objective, and you have specific plans for how you intend to use, analyze, and share Twitter data from your research; you will use this product track for non-commercial purposes."*

This access requirement raises questions about Twitter's influence over academic research. It also excludes independent researchers and non-profits from accessing Twitter's research data. Will this new application process give Twitter the power to decide what research can and cannot access its platform? Will Twitter be granted full, or even partial, intellectual property (IP) rights to research – or include non-disclosure agreements (NDA)?

The discourse in Twitter's announcement of the *Academic Research* track includes: *"**enabling** the future of academic research with the Twitter API"*,

*"**invest** in the success of the academic research community with tailored solutions that better serve their goals"*, and *"we are excited to enable even more research that can create a positive impact on the world, **and on Twitter**, in the **future**"*. This language shows that, whilst Twitter may indeed want to "enable" academic research, it is important to consider Twitter's motives. Twitter is a for-profit organisation with financial pressures. It must **invest** in its platform to sustain **future** growth and success. Will Twitter allow objective and scientific research that improves society but, perhaps, conflicts with Twitter's motives?

9.10.4 Twitter Sample Stream Size

We use Twitter's Stream API as a source of tweeted URLs in our research. This stream typically contains approximately 1% of all global tweets. This does mean that our measurements and analysis are of this 1% sample – and not the full Twitter dataset. However, Twitter states that their stream API provides a *random* sample of its full data set, therefore this should provide an accurate representation of its full dataset. As we explored in Section 3.2: *Data Feeds*, the accuracy of using Twitter's sampled dataset will depend on the type of research being carried out. Our research does not rely on an interconnection of tweets or groups of users on Twitter. However, we do require the timestamps of when URLs in our dataset are first tweeted in order to calculate delay times. Therefore there could be instances when the first appearance of a tweeted URL occurs outside of our dataset. We compensate as much as possible for this by using techniques such as Twitter's Search API to determine first tweet timestamps.

9.11 Ground Truth

A key aspect to the accuracy of our measurement studies lies in our ground truth. We use this ground truth to determine which URLs in our datasets are phishing or malicious, and therefore produce our metrics. In our research, we do not detect tweets containing phishing or malicious URLs that never make it into GSB, PT or OP. Therefore these 3 blacklists make up our ground truth. In Chapter 4: *Design & Implementation*, we discussed our reasoning behind focusing on these 3 blacklists and concluded that they are 3 popular and specialised blacklists that would be suitable for our study. In Chapter 5: *Blacklist Analysis Study*, our results show that GSB contains significantly more phishing URLs than PT and OP. This might help to explain why GSB features so prominently at detecting tweeted URLs that are blacklisted in our other studies. However, we also hypothesised that Twitter may block all URLs that reside in the OP and PT blacklists – which might also explain why we see such few PT and OP blacklisted URLs in our Twitter feed.

We have also discussed how we could leverage additional techniques to increase our ground truth coverage – such as with a honeypot, machine learning classifier, or using more blacklists. A limitation of our triad of blacklists ground truth is that we do not verify the accuracy of these blacklists. However, these blacklists are provided by reputable organisations. Also, we explored existing studies [60, 310, 191] that analysed the effectiveness and reliability of the community-based voting blacklist, PhishTank.

As we discussed in 3.3: *Threat Intelligence & Blacklists*, experiments carried out for our thesis include 3 popular phishing blacklists (PT, OP, GSB). Whilst every effort was made to include a *complete coverage* of phishing blacklists, a number of realistic considerations, such as financial, contractual, etc, impacted our ability to include such a complete set of phishing blacklists. We believe that the core blacklists we have used in our experiments are significant in their coverage and speciality, therefore should provide accurate results.

9.12 Ethics

In Chapter 3: *Related Literature*, we reviewed existing literature that has explored ethical considerations of cyber security research. In Chapter 4: *Design & Implementation*, we designed our measurement studies and infrastructure to accommodate these ethical considerations. In chapters 5 to 8 we carried out our research in an objective and scientific manner that related to our research questions. As discussed in 4.5: *Ethical Considerations*, we are able to share our measurement data, that we have collected for our studies, with other researchers in a privacy-aware manner.

9.13 Comparing Twitter and GSB Policies

In our results we see that GSB detects more tweeted URLs than are blocked by Twitter. We mentioned that phishing and malware are only a small part of Twitter's comprehensive list of rules. In Table 9.1 we compare Twitter and GSB's policies on phishing and malware. We see that Twitter's policy covers malware, phishing, and unwanted software all in one statement. Whereas GSB approaches each attack separately. GSB also explicitly states what action Google Chrome will take if a dangerous website is found – a warning within the browser will be shown. Whereas Twitter do not state what action they will take if a dangerous website is discovered.

This comparison shows, as we would expect, that GSB's core focus is to detect attack websites and protect its users from them. Whereas Twitter states – as part of its broad policy that needs to address many rules that govern what users cannot do on its platform – that users may not

Policy	**Twitter**[278]	**GSB**[107, 101, 105]
Phishing	*"You may not publish or link to malicious content intended to damage or disrupt another person's browser or computer or to compromise a person's privacy."*	*"Since 2005, Safe Browsing has protected users across the web from Social Engineering attacks... If Google detects that [a] website contains social engineering content, the Chrome browser may display a "Deceptive site ahead" warning when visitors view [the] site."*
Malware	Same as above	*"Since 2006, Safe Browsing has warned users when they attempt to navigate to sites that might be malicious... Google checks websites to see whether they host software or downloadable executables that negatively affect the user experience."*
Unwanted Software	Same as above	*"In 2014, we added protection against a broad category of harmful technology that we now call "Unwanted Software": for example, programs disguised as helpful downloads that actually make unexpected changes to your computer like switching your homepage or other browser settings to ones you don't want... "*

Table 9.1: Comparison of Twitter and GSB policies for both phishing and malware.

publish malicious content. In some ways it could be beneficial for Twitter to outsource the detection of dangerous websites to GSB, since GSB specialises in detecting that sort of content. However, our results suggests that Twitter is not preventing its users from visiting all websites that reside in GSB. It could also be that, as we see in the policy wordings, GSB wants to clearly promote its interest in detecting these attacks – since GSB wants users to trust it as a protection provider. Whereas Twitter does not need to promote its detection efforts, since their policy simply states that users may not publish malicious content. Twitter may still deploy detection techniques that we have been unable to measure in this thesis.

9.14 Design & Implementation

Theoretically, if we could access Twitter's Firehose (100%) feed, it would be interesting to repeat our experiments and compare the results. There may very well be small patterns starting to emerge in our tweets dataset that might develop into clear patterns, or they might also even out and not actually be patterns at all. It would also be interesting to repeat our study with additional blacklists – especially malware blacklists. Our ground

truth of malware detection was GSB, which featured phishing URLs more predominantly. This meant we saw fewer tweeted malware URLs in our dataset. Also, as previously discussed, incorporating techniques such as machine learning, honeypots, etc, could help to increase that ground truth coverage.

In terms of implementation, it is important to plan ahead with a longitudinal measurement study. During our measurements there was no planned IT maintenance and we experienced relatively little down time. Sometimes the virtual machine we were running our experiments on would slow down due to heavy use of other virtual machines on the system. But overall this was not too much of a problem.

9.15 Test Driven Development (TDD)

Overall, the TDD process worked well and allowed us to test certain features on smaller sets of data and iron out any problems before deployment. The TDD process also allowed us to be flexible with our research and change direction as we discovered new information or tried different techniques.

The process of finding and debugging errors was, at times, quite frustrating and often time consuming due to the impact it had on our running experiments. As previously discussed in Section 4.6, changes to our data sources (such as API changes) would sometimes impact our measurement studies. Where required, we adapted to these changes in various ways, such as altering our methodology.

9.16 Application to Other Social Media Platforms

In this thesis we investigate the effectiveness of Twitter's phishing and malware defence system. We expand the definition of *effectiveness* in this context, provide an effectiveness evaluation framework, design a measurement infrastructure, and improve measurement studies by addressing soundness and limitations of existing work. Our work can also be applied to other, suitable, online social networks (OSNs). Our research methodology requires URLs (that have been shared on the OSN) and a source of ground truth (e.g. blacklists) to measure effectiveness. Therefore, OSNs that are suitable for our measurements should be reasonably text based. That is, OSNs that allow its users to submit and share text; including URLs. Examples of such OSNs include Facebook, Reddit, forums, IRC, Tumblr, 4chan, etc.

Another important consideration when applying our research to other OSNs is how user-contributed text is publicly shared. For example: on Twitter, all user accounts are set to public by default. This means all users' tweets are shared publicly and included in Twitter's data stream API. Whereas on Facebook, all user profiles are set to *share with friends only* by default. This means that researchers can only access Facebook users' posts if they are friends with the users.

Another important consideration when applying our research to other OSNs is the data sharing policies and data feeds of the OSNs. We leverage Twitter for our research because Twitter's data sharing policy is generous. For example: all tweets are public and users agree to share submitted content (i.e. their tweets) with Twitter and third parties (unless a user explicitly sets their account to private). Therefore, Twitter's data-sharing policy provides researchers with a rich set of data that is suitable for empirical studies. Researchers can access Twitter's data through Twitter's comprehensive API that is well documented[2], features various endpoints[3], and includes numerous tools and libraries across multiple programming languages[4]. These aforementioned points improve the accessibility and efficiency of working with Twitter's data, therefore resulting in improved access to data for researchers.

9.17 Recommendations

Based on our findings, we make the following recommendations to policymakers, technology designers, researchers, and other interested parties. Any organisation wanting to defend their system against phishing and malware attacks by using a blacklist(s) should carefully consider the available blacklist providers. Due to its size, GSB *may* provide a certain level of coverage against these attacks. However, the effectiveness of any such protection will depend on the requirements of the organisation; aspects such as risk appetite must be considered.

Additional protection can be provided by specialist phishing blacklists such as PT and OP. It is important to note that there are usually delay times for attack URLs to appear in blacklists. Therefore, blacklist services such as OP, which leverage automated phishing detection, may detect phishing attack URLs faster than traditional blacklist services. However, the trade-off in using a faster detection service is that the duration, and volume, of attack URLs in the blacklist may be limited. Therefore, attack URLs should always be cross-checked with other blacklists.

[2] https://developer.twitter.com/en/docs

[3] https://developer.twitter.com/en/docs/api-reference-index

[4] https://developer.twitter.com/en/docs/twitter-api/tools-and-libraries

The impact of blacklist delays on user security should also be considered. This applies to all internet users and social media users, not just those on Twitter. Blacklists do not offer 100% protection due to many factors, such as delays. Blacklists take time to update, thereby creating an attack space that cybercriminals can exploit. We observed that heuristic phishing detection of web browsers only catches 38% to 78% of phishing attacks – exposing users to 62% of phishing attacks in the worst case. Therefore internet users should always be aware that they may not be protected against online threats.

Anyone using Twitter should ensure they have adequate phishing and malware protection installed on their device. Any warnings produced by the web browser, or any other protection service, should be adhered to and websites flagged as dangerous should not be visited. This protection is especially important because, as we saw in Chapter 6 and 8, significant numbers of phishing and malware URLs are not blocked by Twitter either at time of tweet or time of click – therefore exposing Twitter users to potential threats. As a direct result, we saw that, during one month alone, phishing and malware attacks on Twitter received more than 1 million clicks. Twitter users should not click any links on Twitter from anyone they do not know or trust. Any organisation using Twitter should ensure its risk appetite aligns with its phishing and malware defence strategy; organisations with lower risk appetites should enforce higher levels of phishing and malware protection as part of its risk management strategy.

Researchers aiming to leverage Twitter's data feed should consider the suitability of the sample size. Twitter's v1.1 *filtered stream* provides a sample of approximately 1% of all global (depending on defined parameters) tweets [195]. This size may introduce bias in studies that rely on interconnected tweets (i.e. analysing crowds in context) for accuracy. Researchers with suitable financial budgets may consider Twitter's 10% and 100% streams to improve accuracy. In addition to this, Twitter claim to have increased their data feed sizes in its newly launched *Academic Research* product tract, part of its v2 API. This also includes access to Tweet history. However, monthly tweet volumes in the v2 API may have reduced compared to the v1.1 API (see Section 9.10.3).

Empirical studies aiming to measure the effectiveness of Twitter's (or any other social media platform's) phishing and malware defence system – or, more generally, cybercrime defence system – should evaluate the accuracy and coverage of their ground truth. Our ground truth comprised 3 blacklists (GSB, PT, and OP). Any non-blacklisted dangerous URLs are not measured in our studies; potentially affecting the soundness of our measurements. The accuracy of ground truth can be evaluated by exploring existing literature (see Section 3.3: Threat Intelligence & Blacklists), checking for false positives / negatives (e.g. through random sampling, machine

learning classification, etc), examining reputation of data provider, etc. The coverage of ground truth can be improved by increasing the number of threat intelligence sources (such as blacklists, honeypots, machine learning classifier(s), etc).

Measurement studies should clearly describe their methodology and technical implementation details. This improves reproducibility of the study. It may be useful to repeat a specific measurement study in future; providing a benchmark to determine how the landscape has changed (e.g. in terms of threats, technologies, demographics, etc). Measurement studies should share their resulting data and make it easily available. This allows future research to reproduce the measurements and also build on and improve the data.

Finally, it is important to address the soundness of measurements and conclusions. Be careful not to make absolute claims (i.e. not independent; require context) or draw conclusions from dichotomous thinking and measurement fallacies. Measurement study results may require context, alongside relevant background topics and existing literature, for sound interpretation.

Chapter Summary

In this chapter we answered our research questions by evaluating the effectiveness of Twitter's phishing and malware defence system at *time-of-tweet* and *time-of-click*. Our findings suggest more can be done to strengthen Twitter's cybersecurity defence system and improve user security.

We addressed soundness, discussed our measurement study results within our effectiveness framework, and assessed our research findings. Evaluating the effectiveness of Twitter's cybercrime defence system is complex and non-trivial. Our framework and analysis contributes towards evaluating effectiveness but cannot provide a complete picture on its own.

We discussed how our research findings could be influenced by numerous factors and contexts; such as URL shorteners, information credibility, and cybercriminal networks. Our measurements offer a focused *window* into phishing and malware activity on Twitter as a *snapshot* that can be used for benchmarking.

We discussed Twitter's potential financial motives, such as pressure to grow its user-base – despite conflicts with community safety (e.g. by removing malicious accounts). We also discussed Twitter's data sharing practices; raising questions about how organisations finance access to Twitter's full data feeds.

We addressed limitations of our research, such as using 3 blacklists (GSB, OP, and PT) as our ground truth and explored how future work could increase ground truth coverage (e.g. by leveraging honeypots, machine learning, and additional blacklists).

We explored how our research methodology can be applied to other online social networks (OSNs) such as Facebook, Reddit, and 4chan. We made recommendations on blacklist usage, implications of blacklist delays on user security, and the importance of Twitter users' risk awareness. Finally, we highlighted the importance of sharing measurement study methodology, technical implementation details, and resulting datasets with the research community.

10 Conclusion

In this thesis we investigated the effectiveness of Twitter's phishing and malware defence system. We defined a set of effectiveness evaluation metrics and framework, created ***Phish****alytics*; our measurement system, then conducted numerous measurement studies. Our measurements studies improve the quality, quantity, and analysis of data available to the internet measurement research community and address our core research questions:

1. How effective is Twitter at protecting its users from phishing and malware URL attacks?
2. How effective are blacklists at helping Twitter to protect its users from phishing and malware attacks?
3. What do popular phishing and malware blacklists consist of? Uptake, dropout, typical lifetimes, and overlap of URLs in these blacklists
4. What is the lifetime of a URL in a phishing/malware blacklist?
5. Are blacklists affected by delays – and, if so, what impact do blacklist delays have on user security?
6. How effective is Twitter's URL shortener, *t.co*, at protecting Twitter users from phishing and malware attacks?
7. What is the comparison between Twitter's use of blacklists and Twitter's URL blocking (via *t.co*) at protecting users from phishing and malware attacks?
8. To explore the impact of phishing and malware attacks on Twitter: how many Twitter users click on publicly tweeted phishing or malware URLs that have been blacklisted?

Table 10.1 summarises our contributions; linked to our research questions.

Contribution	RQ
Expand the definition of *effectiveness* – for measuring Twitter's cybercrime defence system – to include new and improved metrics; addressing soundness and limitations of existing work.	All
Define an effectiveness evaluation framework.	All
Design and implement ***Phish**alytics*: novel measurement infrastructure to conduct longitudinal measurement studies.	All
Measurement data (1.5TB); available to help future research.	All
Methodology to improve current understanding of how to collect and analyse internet measurements.	All
Phishalytics technical implementation details and full codebase [37] to aid reproducibility.	All
Novel characterisation and analysis of 3 popular phishing blacklists; including comprehensiveness and typical URL uptake, dropout, lifetime, and overlap [38].	3, 4
Novel, fine-grained and in-depth study into the effectiveness of blacklists at protecting Twitter users from phishing and malware attacks; *time-of-tweet* defence study [40].	1, 2, 5, 8
Novel evidence to suggest Twitter may no longer be using the GSB blacklist to protect users.	1, 2
Contemporary analysis of Twitter's phishing and malware defence system at time of tweet.	1, 2, 5, 8
Improve internet measurements by including URL redirection chain(s); addressing soundness and methodological limitations of existing literature.	1, 2, 5
Improve internet measurements by introducing specialised blacklists: GSB, PT, and OP; addressing soundness and methodological limitations of existing literature.	2, 3, 4, 5
Improve measurement accuracy of tweeted URL blacklist delay times; addressing soundness and methodological limitations of existing work.	1, 2, 5, 8
Novel implementation of historical tweet context in methodology to increase accuracy of time of first tweet.	1, 2, 5
Improve measurement accuracy of blacklisted URL clicks by defining timeframe and referrer; addressing methodological limitations of existing literature.	8
Update and improve the accuracy of existing research into the effectiveness of web browser phishing detection; test suite comprising multiple operating systems; categorisation of URL blacklist status at time of test – to determine heuristic detection; web browser warning effectiveness framework.	1
Novel study into the effectiveness of Twitter's phishing and malware defence system at *time-of-click*; Twitter's URL shortener (t.co) [39].	6, 7

Table 10.1: Thesis contributions linked to research question (RQ) numbers.

Blacklist Analysis Study (Chapter 5)
We investigated 3 key phishing blacklists: Google Safe Browsing (GSB), OpenPhish (OP), and PhishTank (PT). In this longitudinal measurement study we analysed the uptake, dropout, typical lifetimes, and considered the overlap of URLs in these blacklists. During our 75-day measurement period we determined that GSB contained an average of 1,581,351 URLs, compared to 12,433 in PT and 3,861 in OP. We contribute the first such study of these 3 blacklists.

Our measurements revealed that the OP blacklist removed a significant volume of URLs from its dataset after a duration of 5 and 7 days; no URLs remained in OP for more than 21 days. Therefore potentially limiting OP's effectiveness at protecting users from phishing attacks. We saw that, across all 3 blacklists, as time increased, fewer URLs remained blacklisted – phishing URLs are often short-lived.

We determined that none of the 3 blacklists enforced a one-time-only URL policy in their dataset; URLs reappeared in the blacklists if they continued or re-emerged as a threat. This is good for users because they will be protected against reoffending phishing websites. We also showed that a significant number of URLs reappear in all 3 blacklists within 1 day of removal – suggesting that these URLs were either removed too soon or that they came back online.

Finally, we compared the PT and OP blacklists and discovered that 11,603 unique URLs resided in both of these blacklists – a 12% overlap. Despite its smaller average size, OP detected over 90% of these overlapping URLs before PT did.

Time-of-Post Twitter Study (Chapter 6)
We examined how effective URL blacklists are at protecting Twitter users against phishing and malware attacks. In this longitudinal measurement study we analysed over 182 million URL-containing public tweets collected from Twitter's Stream API, over a 2 month period, and compared these URLs against 3 popular social engineering, phishing, and malware blacklists (GSB, PT, and OP).

Our main discovery was that, although the majority of phishing and malware URLs are detected by the GSB blacklist (which is used by popular web browsers) within 6 hours of being tweeted, there are still a large number of URLs that take at least 20 days to appear in GSB.

We discovered 4,930 tweets containing URLs leading to social engineering websites that took between 18 and 30 days to appear in the blacklist. Between them, these 4,930 tweets had been tweeted to over 131 million Twitter users. We also discovered 1,126 tweets containing 376 blacklisted *Bitly* URLs that had received a combined total of 991,012 clicks. These URLs represented 11% of the total blacklisted social engineering URLs in

that month.

The fact that the GSB blacklist can take weeks to detect dangerous URLs poses serious security risks to Twitter users: tweets containing blacklisted URLs are sent to large numbers of followers and receive a significant amount of clicks, thereby exposing users to dangerous websites.

Conversely, and surprisingly to us, there are large numbers of URLs being tweeted that have already been blacklisted by GSB. This may suggest that Twitter is not using the GSB blacklist to block malicious tweets at the time of tweeting, contrary to what was once reported to be the case [200].

In summary, whilst blacklists are reasonably effective at protecting Twitter users from phishing and malware attacks, there is still an unprotected space that leaves Twitter users vulnerable. It could be that Twitter relies on web browser's built-in defence systems to protect users against phishing and malware attacks.

Web Browser Phishing Detection Study (Chapter 7)

We created a comprehensive testing environment to assess the effectiveness of web browsers' anti-phishing technology – across a variety of different operating systems. We then used this testing environment to measure how effective popular web browsers are at detecting phishing websites.

By testing popular web browsers against both known (i.e. blacklisted) and unknown (i.e. non-blacklisted) phishing websites, our results showed that most of today's popular web browsers are able to detect over 80% of known phishing websites. This might suggest that Twitter does not need to protect its own users against phishing attacks – because Twitter can rely on the web browsers to act as a defence mechanism. However, the web browsers we tested were not as effective at detecting unknown (*i.e.* non-blacklisted) phishing websites – detecting only 38% to 78%.

We also analysed the heuristic phishing detection methodology of the Chromium web browser to improve our understanding of how it works and its effectiveness. We posed a number of theoretical phishing detection circumvention techniques, along with their potential weaknesses.

It is important to note that users are not completely protected against known phishing attacks, since blacklists take time to update – which can create a window of opportunity for attackers. Therefore, if Twitter is relying on web browsers to protect their users against phishing websites, then Twitter could be leaving its users exposed and vulnerable to some of these attacks.

Time-of-Click Twitter Study (Chapter 8)
We investigated the effectiveness of Twitter's URL shortener (*t.co*) at protecting users from phishing and malware attacks. In this longitudinal measurement study we follow-up from one of our key findings in Chapter 6: that over 10,000 blacklisted phishing and malware URLs were posted to Twitter during our 2-month measurement study, which lead to over 1.6 million clicks that came directly from Twitter users – therefore exposing people to potentially harmful cyber attacks.

We investigate if Twitter blocks any URLs posted to its platform at time of click – therefore potentially protecting its users against the 10,000+ blacklisted URLs that we detected in the twitter dataset in Chapter 6. We discovered a reduction in the number of phishing and malware URLs posted to Twitter in 2018-19 compared to 2017 – suggesting a potential improvement in Twitter's effectiveness at filtering blacklisted URLs at time of tweet. However, we also discovered that only about 12% of blacklisted phishing and malware URLs – which were not blocked at time of tweet and therefore posted to the platform – were blocked by Twitter in 2018-19 at time of click.

Our results indicate that, despite a reduction in the number of blacklisted URLs posted to the social network, Twitter's URL shortener is not particularly effective at filtering phishing and malware URLs – therefore people are still exposed to these cyber attacks on Twitter.

Discussion Summary
Key points we discussed in Chapter 9 include the difficulties of conducting longitudinal measurement studies in an ever-changing landscape and the challenges associated with comparing results from different measurement timeframes.

We addressed limitations of datasets we use, such as Twitter and other APIs. We mitigate limitations by meticulously planning and researching our methodology. We also discussed our ground truth, which consists of 3 popular blacklists, and explored potential areas for future work.

We discussed the soundness of measurements and conclusions; drawing absolute conclusions (i.e. not independent; require context) from our measurement results may lead to dichotomous thinking and oversimplification. Evaluating the effectiveness of Twitter's phishing and malware defence system is complex. Our framework and analysis contributes towards evaluating effectiveness but cannot provide a complete picture on its own.

We explored how existing literature can provide context to our measurement study results and what implications these may have. We also discussed Twitter's motives; how it is under increasing financial pressure to grow the platform and its user-base. Yet, simultaneously, keep its community safe

and remove malicious accounts.

We made recommendations on blacklist usage, implications of blacklist delays on user security, and the importance of Twitter users' risk awareness. Twitter users should ensure their risk appetite aligns with their phishing and malware defence strategy; lower risk appetites should enforce higher levels of phishing and malware protection.

Finally, we highlighted the importance of sharing measurement study methodology, technical implementation details, and resulting datasets with the research community. We aim to improve risk awareness and better enable policymakers, researchers, and technology designers to strengthen online user security.

RESEARCH SUMMARY

Addressing our 8 research questions (RQ):

RQ 1: Our research results show that Twitter does protect its users against some phishing and malware attacks. However, we observed over 10,000 blacklisted phishing and malware URLs that had been publicly shared on the social network. Therefore, suggesting that Twitter could do more to protect its users against cyber attacks.

RQ 2: Twitter does not appear to be effectively leveraging the GSB blacklist to protect users against phishing and malware threats. Twitter may be relying on web browsers' built-in security to defend against such attacks. However, web browsers are not perfect at detecting phishing attacks – missing up to 62% of non-blacklisted URLs.

RQ 3 & 4: Our analysis of popular phishing and malware blacklists (GSB, OP, and PT) observed an average of 1.6 million URLs in the GSB blacklist, over 12 thousand URLs in PT, and nearly 4 thousand URLs in OP. The OP blacklist appears to enforce strict limits on how long URLs can remain in its dataset for. The PT and OP blacklists see a 12% overlap of URLs; OP detected over 90% of these URLs before PT. On average, URLs remain in GSB for 10 days, PT for 2 days, and OP for 5 days. However, we also see that some URLs remain in blacklists for the entire duration of our measurement experiments.

RQ 5: Blacklist delays can have a significant impact on user security. Although the majority of URLs in our study were detected by the GSB blacklist within 6 hours of being tweeted, a significant number of URLs took over 20 days to appear in GSB. Blacklist delays can also affect web browsers' ability to defend against such attacks.

RQ 6 & 7: Twitter does block certain URLs at time of click (via its URL shortener, *t.co*), therefore providing additional protection to Twitter users. However, we observed that only 12% of blacklisted phishing and malware URLs were blocked by Twitter at time of click – therefore potentially exposing people to these threats.

RQ 8: Our study observed that blacklisted phishing and malware URLs – that were tweeted publicly on the social network – received more than 1.6 million clicks from Twitter users – therefore exposing people to potentially harmful cyber attacks.

Results from our measurement studies should be interpreted in context; external factors of the landscapes outside our measurements may influence our findings. We aim to improve risk awareness and better enable policy-makers, researchers, and technology designers to strengthen online user security.

We contribute towards the internet measurement community, and improve the reproducibility of our measurement studies, by including our methodology, technical implementation details, and sharing our resulting data (see Chapter 4).

Bibliography

[1] ABDELHAMID, N., AYESH, A., AND THABTAH, F. Phishing detection based associative classification data mining. *Expert Systems with Applications 41*, 13 (2014), 5948–5959.

[2] ABU-NIMEH, S., NAPPA, D., WANG, X., AND NAIR, S. A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit* (2007), ACM, pp. 60–69.

[3] ACM. Artifact Review and Badging Version 1.1. https://www.acm.org/publications/policies/artifact-review-and-badging-current, 2020.

[4] AGGARWAL, A., RAJADESINGAN, A., AND KUMARAGURU, P. Phishari: Automatic realtime phishing detection on twitter. In *2012 eCrime Researchers Summit* (2012), IEEE, pp. 1–12.

[5] AGGARWAL, C. C. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases* (2005), VLDB Endowment, pp. 901–909.

[6] AKHAWE, D., AND FELT, A. P. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)* (2013), pp. 257–272.

[7] AKINYELU, A. A. Machine learning and nature inspired based phishing detection: A literature survey. *International Journal on Artificial Intelligence Tools 28*, 05 (2019), 1930002.

[8] ALLCOTT, H., AND GENTZKOW, M. Social media and fake news in the 2016 election. *Journal of economic perspectives 31*, 2 (2017), 211–36.

[9] ALLIX, K., BISSYANDÉ, T. F., KLEIN, J., AND LE TRAON, Y. Are your training datasets yet relevant? In *International Symposium on Engineering Secure Software and Systems* (2015), Springer, pp. 51–67.

[10] ALLLMAN, M., AND PAXSON, V. Issues and etiquette concerning use of shared measurement data. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (2007), ACM, pp. 135–140.

[11] ALLMAN, M., BLANTON, E., AND EDDY, W. A scalable system for sharing internet measurements. In *Proc. PAM* (2002), Citeseer.

[12] ALMOMANI, A., GUPTA, B., ATAWNEH, S., MEULENBERG, A., AND ALMOMANI, E. A survey of phishing email filtering techniques. *IEEE communications surveys & tutorials 15*, 4 (2013), 2070–2090.

[13] ALSHARNOUBY, M., ALACA, F., AND CHIASSON, S. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies 82* (2015), 69–82.

[14] AMAZON. Mechanical Turk. [https://www.mturk.com; accessed Nov-2019].

[15] ANNIE COLBERT. 7 Fake Hurricane Sandy Photos You're Sharing on Social Media. http://mashable.com/2012/10/29/fake-hurricane-sandy-photos/. Accessed: Feb 2020.

[16] ANTONIADES, D., POLAKIS, I., KONTAXIS, G., ATHANASOPOULOS, E., IOANNIDIS, S., MARKATOS, E. P., AND KARAGIANNIS, T. we. b: The web of short URLs. In *Proceedings of the 20th international conference on World Wide Web* (2011), ACM, pp. 715–724.

[17] APWG. Anti-Phishing Working Group. [http://www.antiphishing.org/; accessed Aug-2015].

[18] APWG. Phishing Activity Trends Reports. https://apwg.org/trendsreports/. Accessed: Jan 2020.

[19] APWG. Phishing Activity Trends Report, 2nd Quarter 2019, 2019.

[20] ARDON, S., BAGCHI, A., MAHANTI, A., RUHELA, A., SETH, A., TRIPATHY, R. M., AND TRIUKOSE, S. Spatio-temporal analysis of topic popularity in twitter. *arXiv preprint arXiv:1111.2904* (2011).

[21] ARMSTRONG, T. Twitter – Malware through time. https://securelist.com/twitter-malware-through-time/29775/, 2011.

[22] ASLAM, S. Twitter by the Numbers: Stats, Demographics & Fun Facts. https://www.omnicoreagency.com/twitter-statistics/, 2018.

[23] AV COMPARATIVES. Browser Anti-Phishing-Test December 2012. [http://www.av-comparatives.org/wp-content/uploads/2012/12/avc_phi_browser_201212_en.pdf; accessed Aug-2015].

[24] AV-TEST - THE INDEPENDENT IT-SECURITY INSTITUTE. Total Malware. [https://www.av-test.org/en/statistics/malware/; accessed Feb 2020].

[25] AZEEZ, N. A., ADE, J., MISRA, S., ADEWUMI, A., VAN DER VYVER, C., AND AHUJA, R. Identifying phishing through web content and addressed bar-based features. In *Data Management, Analytics and Innovation*. Springer, 2020, pp. 19–29.

[26] BAJPAI, V., KÜHLEWIND, M., OTT, J., SCHÖNWÄLDER, J., SPEROTTO, A., AND TRAMMELL, B. Challenges with reproducibility. In *Proceedings of the Reproducibility Workshop* (2017), pp. 1–4.

[27] BASNET, R., MUKKAMALA, S., AND SUNG, A. H. Detection of phishing attacks: A machine learning approach. In *Soft Computing Applications in Industry*. Springer, 2008, pp. 373–383.

[28] BASNET, R. B., AND SUNG, A. H. Classifying phishing emails using confidence-weighted linear classifiers. In *International Conference on Information Security and Artificial Intelligence (ISAI)* (2010), pp. 108–112.

[29] BASNET, R. B., AND SUNG, A. H. Mining web to detect phishing URLs. In *2012 11th International Conference on Machine Learning and Applications* (2012), vol. 1, IEEE, pp. 568–573.

[30] BASNET, R. B., SUNG, A. H., AND LIU, Q. Rule-based phishing attack detection. In *Proceedings of the International Conference on Security and Management (SAM)* (2011), The Steering Committee of The World Congress in Computer Science, p. 1.

[31] BASNET, R. B., SUNG, A. H., AND LIU, Q. Feature selection for improved phishing detection. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (2012), Springer, pp. 252–261.

[32] BBC NEWS. Australia fires: Misleading maps and pictures go viral. https://www.bbc.co.uk/news/blogs-trending-51020564. Accessed: Jan 2020.

[33] BBC NEWS. Coronavirus: How a misleading map went global. https://www.bbc.co.uk/news/world-51504512. Accessed: Feb 2020.

[34] BBC NEWS. Gary McKinnon extradition to US blocked by Theresa May. https://www.bbc.co.uk/news/uk-19957138. Accessed: Nov 2019.

[35] BBC NEWS. Twitter shares jump 73% in market debut. https://www.bbc.co.uk/news/business-24851054. Accessed: Feb 2020.

[36] BEGUM, A., AND BADUGU, S. A Study of Malicious URL Detection Using Machine Learning and Heuristic Approaches. In *Advances in Decision Sciences, Image Processing, Security and Computer Vision*. Springer, 2020, pp. 587–597.

[37] BELL, S. **Phish**alytics Codebase. https://github.com/sjbell/phishalytics.

[38] BELL, S., AND KOMISARCZUK, P. An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank. In *Proceedings of the Australasian Computer Science Week Multiconference* (2020).

[39] BELL, S., AND KOMISARCZUK, P. Measuring the Effectiveness of Twitter's URL Shortener (*t.co*) at Protecting Users from Phishing and Malware Attacks. In *Proceedings of the Australasian Computer Science Week Multiconference* (2020).

[40] BELL, S., PATERSON, K., AND CAVALLARO, L. Catch Me (On Time) If You Can: Understanding the Effectiveness of Twitter URL Blacklists. *arXiv preprint arXiv:1912.02520* (2019).

[41] BENEVENUTO, F., MAGNO, G., RODRIGUES, T., AND ALMEIDA, V. Detecting spammers on Twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)* (2010), vol. 6, p. 12.

[42] BERGEN, J. Twitter finally adds its own URL shortening service. https://www.geek.com/news/twitter-finally-adds-its-own-url-shortening-service-1388467/, 2011.

[43] BITLY. URL Shortener, Custom Branded URLs, API & Link Management. https://bitly.com. Accessed: Feb 2016.

[44] BLUM, A., WARDMAN, B., SOLORIO, T., AND WARNER, G. Lexical feature based phishing URL detection using online learning. In *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security* (2010), ACM, pp. 54–60.

[45] BONNINGTON, C. Twitter is promoting a "get verified" phishing scam. https://www.dailydot.com/debug/twitter-promoted-phishing-site/, 2018.

[46] BOSHMAF, Y., MUSLUKHOV, I., BEZNOSOV, K., AND RIPEANU, M. Design and analysis of a social botnet. *Computer Networks 57*, 2 (2013), 556–578.

[47] BRITISH COMPUTING SOCIETY. The Chartered Institute for IT. [http://www.bcs.org/; accessed Aug-2015].

[48] BRITISH COMPUTING SOCIETY. Code of Conduct, 2015. [https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf; accessed Aug-2015].

[49] BURNAP, P., JAVED, A., RANA, O. F., AND AWAN, M. S. Real-time classification of malicious URLs on Twitter using machine activity data. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2015), IEEE, pp. 970–977.

[50] BURNAP, P., RANA, O., WILLIAMS, M., HOUSLEY, W., EDWARDS, A., MORGAN, J., SLOAN, L., AND CONEJERO, J. Cosmos: Towards an integrated and scalable service for analysing social media on demand. *International Journal of Parallel, Emergent and Distributed Systems 30*, 2 (2015), 80–100.

[51] CÁCERES, R., DUFFIELD, N. G., HOROWITZ, J., AND TOWSLEY, D. F. Multicast-based inference of network-internal loss characteristics. *IEEE Transactions on Information theory 45*, 7 (1999), 2462–2480.

[52] CANALI, D., AND BALZAROTTI, D. Behind the scenes of online attacks: an analysis of exploitation behaviors on the web. In *20th Annual Network & Distributed System Security Symposium (NDSS 2013)* (2013).

[53] CASTILLO, C., MENDOZA, M., AND POBLETE, B. Information credibility on Twitter. In *Proceedings of the 20th international conference on World Wide Web* (2011), ACM, pp. 675–684.

[54] CHA, M., HADDADI, H., BENEVENUTO, F., AND GUMMADI, K. P. Measuring user influence in Twitter: The million follower fallacy. In *fourth international AAAI conference on weblogs and social media* (2010).

[55] CHHABRA, S., AGGARWAL, A., BENEVENUTO, F., AND KUMARAGURU, P. Phi. sh/$ ocial: The Phishing Landscape Through Short URLs. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference* (2011), ACM, pp. 92–101.

[56] CHOU, N., LEDESMA, R., TERAGUCHI, Y., MITCHELL, J. C., ET AL. Client-side defense against web-based identity theft. In *NDSS* (2004).

[57] CHRISTOPHER KESSLING. Release from Custodial Sentences. https://www.defence-barrister.co.uk/release-from-custodial-sentences. Accessed: Nov 2019.

[58] CHU, Z., GIANVECCHIO, S., WANG, H., AND JAJODIA, S. Detecting automation of Twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing 9*, 6 (2012), 811–824.

[59] CHU, Z., WIDJAJA, I., AND WANG, H. Detecting social spam campaigns on Twitter. In *International Conference on Applied Cryptography and Network Security* (2012), Springer, pp. 455–472.

[60] CLAYTON, R., MOORE, T., AND CHRISTIN, N. Concentrating correctly on cybercrime concentration. In *WEIS* (2015).

[61] CLORE, G. L., AND HUNTSINGER, J. R. How emotions inform judgment and regulate thought. *Trends in cognitive sciences 11*, 9 (2007), 393–399.

[62] CONTAGIO MALWARE DUMP. An Overview of Exploit Packs (Update 25) May 2015. http://contagiodump.blogspot.be/2010/06/overview-of-exploit-packs-update.html. Accessed: Nov 2019.

[63] COVA, M., KRUEGEL, C., AND VIGNA, G. Detection and analysis of drive-by-download attacks and malicious javascript code. In *Proceedings of the 19th international conference on World wide web* (2010), pp. 281–290.

[64] CUI, Q., JOURDAN, G.-V., BOCHMANN, G. V., COUTURIER, R., AND ONUT, I.-V. Tracking phishing attacks over time. In *Proceedings of the 26th International Conference on World Wide Web* (2017), International World Wide Web Conferences Steering Committee, pp. 667–676.

[65] CUNHA, C. R., BESTAVROS, A., AND CROVELLA, M. E. Characteristics of WWW client-based traces. Tech. rep., Boston University Computer Science Department, 1995.

[66] DA CHRONIC. AOHell Documentation. [http://www.aolwatch.org/chronic2.htm; accessed Feb 2020].

[67] DARREN LILLEKER. You're probably more susceptible to misinformation than you think. https://theconversation.com/youre-probably-more-susceptible-to-misinformation-than-you-think-129171. Accessed: Nov 2019.

[68] DE CHOUDHURY, M., COUNTS, S., AND CZERWINSKI, M. Find me the right content! Diversity-based sampling of social media spaces for topic-centric search. In *Fifth International AAAI Conference on Weblogs and Social Media* (2011).

[69] DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological) 39*, 1 (1977), 1–22.

[70] DEPAULO, B. M., LINDSAY, J. J., MALONE, B. E., MUHLENBRUCK, L., CHARLTON, K., AND COOPER, H. Cues to deception. *Psychological bulletin 129*, 1 (2003), 74.

[71] DHAMIJA, R., AND TYGAR, J. D. The battle against phishing: Dynamic security skins. In *Proceedings of the 2005 symposium on Usable privacy and security* (2005), ACM, pp. 77–88.

[72] DHAMIJA, R., TYGAR, J. D., AND HEARST, M. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 581–590.

[73] DOMINIC RUSHE. Twitter posts revenues of $242m but share price plummets as growth stalls. https://www.theguardian.com/technology/2014/feb/05/twitter-revenues-share-price-drops. Accessed: Feb 2020.

[74] EFRON, B. *The jackknife, the bootstrap, and other resampling plans*, vol. 38. Siam, 1982.

[75] EGELMAN, S., BONNEAU, J., CHIASSON, S., DITTRICH, D., AND SCHECHTER, S. It's not stealing if you need it: A panel on the ethics of performing research using public data of illicit origin. In *International Conference on Financial Cryptography and Data Security* (2012), Springer, pp. 124–132.

[76] EGELMAN, S., CRANOR, L. F., AND HONG, J. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), ACM, pp. 1065–1074.

[77] ENGINEERING AND PHYSICAL SCIENCES RESEARCH COUNCIL. Principles. [https://epsrc.ukri.org/about/standards/researchdata/principles/; accessed Feb 2020].

[78] ESET. First Twitter-controlled Android botnet discovered. https://www.welivesecurity.com/2016/08/24/first-twitter-controlled-android-botnet-discovered/, 2016.

[79] FACEBOOK. Threat Exchange. https://developers.facebook.com/programs/threatexchange. Accessed: Nov 2019.

[80] FBI. Cyber Crime. https://www.fbi.gov/investigate/cyber. Accessed: Nov 2019.

[81] FELEGYHAZI, M., KREIBICH, C., AND PAXSON, V. On the potential of proactive domain blacklisting. *LEET 10* (2010), 6–6.

[82] FELT, A. P., AINSLIE, A., REEDER, R. W., CONSOLVO, S., THYAGARAJA, S., BETTES, A., HARRIS, H., AND GRIMES, J. Improving SSL warnings: Comprehension and adherence. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems* (2015), pp. 2893–2902.

[83] FETTE, I., SADEH, N., AND TOMASIC, A. Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 649–656.

[84] FILIPOVICH, A. gglsbl: Python client library for Google Safe Browsing Update API v4. https://github.com/afilipovich/gglsbl/.

[85] FREE SOFTWARE FOUNDATION, INC. GNU Screen: full-screen window manager. [https://www.gnu.org/software/screen/; accessed Feb 2016].

[86] FTC. Twitter Settles Charges that it Failed to Protect Consumers' Personal Information; Company Will Establish Independently Audited Information Security Program. https://www.ftc.gov/news-events/press-releases/2010/06/twitter-settles-charges-it-failed-protect-consumers-personal, 2010.

[87] FTC. FTC Accepts Final Settlement with Twitter for Failure to Safeguard Personal Information. https://www.ftc.gov/news-events/press-releases/2011/03/ftc-accepts-final-settlement-twitter-failure-safeguard-personal-0, 2011.

[88] FTC. Twitter, Inc, Case Timeline. https://www.ftc.gov/enforcement/cases-proceedings/092-3093/twitter-inc-corporation, 2011.

[89] GANDOTRA, E., BANSAL, D., AND SOFAT, S. Malware analysis and classification: A survey. *Journal of Information Security 5*, 02 (2014), 56.

[90] GAO, H., HU, J., WILSON, C., LI, Z., CHEN, Y., AND ZHAO, B. Y. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement* (2010), ACM, pp. 35–47.

[91] GARERA, S., PROVOS, N., CHEW, M., AND RUBIN, A. D. A framework for detection and measurement of phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode* (2007), ACM, pp. 1–8.

[92] GARETH CORFIELD. Guilty of hacking in the UK? Worry not: Stats show prison is unlikely. https://www.theregister.co.uk/2019/05/29/computer_misuse_act_prosecutions_analysis. Accessed: Nov 2019.

[93] GERBET, T., KUMAR, A., AND LAURADOUX, C. A privacy analysis of Google and Yandex safe browsing. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (2016), IEEE, pp. 347–358.

[94] GHOSH, S., SHARMA, N., BENEVENUTO, F., GANGULY, N., AND GUMMADI, K. Cognos: crowdsourcing search for topic experts in microblogs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (2012), ACM, pp. 575–590.

[95] GHOSH, S., VISWANATH, B., KOOTI, F., SHARMA, N. K., KORLAM, G., BENEVENUTO, F., GANGULY, N., AND GUMMADI, K. P. Understanding and combating link farming in the Twitter social network. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 61–70.

[96] GHOSH, S., ZAFAR, M. B., BHATTACHARYA, P., SHARMA, N., GANGULY, N., AND GUMMADI, K. On sampling the wisdom of crowds: random vs. expert sampling of the Twitter stream. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management* (2013), ACM, pp. 1739–1744.

[97] GIGLIETTO, F., AND SELVA, D. Second screen and participation: A content analysis on a full season dataset of tweets. *Journal of Communication 64*, 2 (2014), 260–277.

[98] GONZALEZ, H., NANCE, K., AND NAZARIO, J. Phishing by form: The abuse of form sites. In *2011 6th International Conference on Malicious and Unwanted Software* (2011), IEEE, pp. 95–101.

[99] GONZÁLEZ-BAILÓN, S., WANG, N., RIVERO, A., BORGE-HOLTHOEFER, J., AND MORENO, Y. Assessing the bias in samples of large online networks. *Social Networks 38* (2014), 16–27.

[100] GOOGLE. Chromium Source Code. [https://code.google.com/p/chromium/codesearch#chromium/src/; accessed Aug-2015].

[101] GOOGLE. Malware and Unwanted Software. [https://support.google.com/webmasters/answer/3258249; accessed Nov-2015].

[102] GOOGLE. Safe Browsing. [https://safebrowsing.google.com/; accessed Nov-2015].

[103] GOOGLE. Safe Browsing Transparency Report; API Data Endpoint: Number of sites deemed dangerous. https://transparencyreport.google.com/transparencyreport/api/v3/safebrowsing/sites?dataset=1&series=malware,phishing&start=1148194800000&end=1615926688000. Accessed: March 2021.

[104] GOOGLE. Safe Browsing Transparency Report; API Data Endpoint: Unsafe websites detected per week. https://transparencyreport.google.com/transparencyreport/api/v3/safebrowsing/sites?dataset=0&series=malwareDetected,phishingDetected&start=1148194800000&end=1615926688000. Accessed: March 2021.

[105] GOOGLE. Social Engineering (Phishing and Deceptive Sites). [https://support.google.com/webmasters/answer/6350487/; accessed Nov-2015].

[106] GOOGLE. Transparency Report - Safe Browsing: malware and phishing. https://transparencyreport.google.com/safe-browsing/overview. Accessed: Feb 2020.

[107] GOOGLE. Unwanted Software Policy. [https://www.google.com/about/unwanted-software-policy.html; accessed Nov-2015].

[108] GOOGLE. Safe Browsing protection from even more deceptive attacks. https://security.googleblog.com/2015/11/safe-browsing-protection-from-even-more.html, 2015.

[109] GOVINDAN, R., AND REDDY, A. An analysis of internet inter-domain topology and route stability. In *Proceedings of INFOCOM'97* (1997), vol. 2, IEEE, pp. 850–857.

[110] GRIER, C., THOMAS, K., PAXSON, V., AND ZHANG, M. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security* (2010), ACM, pp. 27–37.

[111] GUARDIAN, THE. How many fake Sandy pictures were really shared on social media? https://www.theguardian.com/news/datablog/2012/nov/06/fake-sandy-pictures-social-media. Accessed: Feb 2020.

[112] GUPTA, A., AND KUMARAGURU, P. Credibility ranking of tweets during high impact events. In *Proceedings of the 1st workshop on privacy and security in online social media* (2012), Acm, p. 2.

[113] GUPTA, A., LAMBA, H., KUMARAGURU, P., AND JOSHI, A. Faking sandy: characterizing and identifying fake images on Twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web* (2013), ACM, pp. 729–736.

[114] GUPTA, B. B., TEWARI, A., JAIN, A. K., AND AGRAWAL, D. P. Fighting against phishing attacks: state of the art and future challenges. *Neural Computing and Applications 28*, 12 (2017), 3629–3654.

[115] GYAWALI, B., SOLORIO, T., MONTES-Y GÓMEZ, M., WARDMAN, B., AND WARNER, G. Evaluating a semisupervised approach to phishing URL identification in a realistic scenario. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference* (2011), ACM, pp. 176–183.

[116] HAN, X., KHEIR, N., AND BALZAROTTI, D. The role of cloud services in malicious software: Trends and insights. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2015), Springer, pp. 187–204.

[117] HAN, X., KHEIR, N., AND BALZAROTTI, D. Phisheye: Live monitoring of sandboxed phishing kits. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), ACM, pp. 1402–1413.

[118] HM GOVERNMENT. Computer Misuse Act 1990. http://www.legislation.gov.uk/ukpga/1990/18/contents. Accessed: Nov 2019.

[119] HM GOVERNMENT. Data Protection Act 1998. [http://www.legislation.gov.uk/ukpga/1998/29/contents; accessed Aug-2013].

[120] HM GOVERNMENT. Policy Paper: National Cyber Security Strategy 2016 to 2021. https://www.gov.uk/government/publications/national-cyber-security-strategy-2016-to-2021. Accessed: Feb 2016.

[121] HM GOVERNMENT. Regulation of Investigatory Powers Act 2000. [http://www.legislation.gov.uk/ukpga/2000/23/contents; accessed Oct-2013].

[122] HM GOVERNMENT - THE CROWN PROSECUTION SERVICE. Cybercrime - prosecution guidance. https://www.cps.gov.uk/legal-guidance/cybercrime-prosecution-guidance. Accessed: Nov 2019.

[123] HM GOVERNMENT - UK PARLIAMENT. Disinformation and "fake news". https://www.parliament.uk/business/committees/committees-a-z/commons-select/digital-culture-media-and-sport-committee/inquiries/parliament-2017/fake-news-17-19/. Accessed: Jan 2020.

[124] HONEY NET. Capture HTC. [https://projects.honeynet.org/capture-hpc; accessed Nov-2019].

[125] IDIKA, N., AND MATHUR, A. P. A survey of malware detection techniques. *Purdue University 48* (2007), 2007–2.

[126] INTERNET ARCHIVE. The Heritrix web crawler project. http://crawler.archive.org/, 2018.

[127] INTERNET SOCIETY, THE. RFC: Computing TCP's Retransmission Timer. https://tools.ietf.org/html/rfc2988. Accessed: Aug 2019.

[128] INTERNET SOCIETY, THE. RFC: Hypertext Transfer Protocol – HTTP/1.1 – Section 10.4: Client Error 4xx. https://tools.ietf.org/html/rfc2616#section-10.4. Accessed: Aug 2019.

[129] INTERNET SOCIETY, THE. RFC: Hypertext Transfer Protocol – HTTP/1.1 – Section 10.5.5: 504 Gateway Timeout. https://tools.ietf.org/html/rfc2616#section-10.5.4. Accessed: Aug 2019.

[130] INTERNETLIVESTATS. Twitter Usage Stats. http://www.internetlivestats.com/twitter-statistics/. Accessed: April 2016.

[131] INTERPOL. Cybercrime. https://www.interpol.int/en/Crimes/Cybercrime. Accessed: Nov 2019.

[132] IRANI, D., WEBB, S., GIFFIN, J., AND PU, C. Evolutionary study of phishing. In *eCrime Researchers Summit* (2008), IEEE, pp. 1–10.

[133] ISO GUIDE 73:2009. Risk management – Vocabulary. Standard, International Organization for Standardization, Geneva, CH, Nov. 2009. https://www.iso.org/standard/44651.html, Accessed: August 2018.

[134] JACOB KASTRENAKES. Twitter keeps losing monthly users, so it's going to stop sharing how many. https://www.theverge.com/2019/2/7/18213567/twitter-to-stop-sharing-mau-as-users-decline-q4-2018-earnings. Accessed: Feb 2020.

[135] JACOB KASTRENAKES. Twitter's growth looks way better now that it's hiding monthly usage numbers. https://www.theverge.com/2019/7/26/8929933/twitter-q2-2019-earnings. Accessed: Feb 2020.

[136] JAGATIC, T. N., JOHNSON, N. A., JAKOBSSON, M., AND MENCZER, F. Social phishing. *Communications of the ACM 50*, 10 (2007), 94–100.

[137] JOHN LEYDEN. McKinnon will not be extradited to the US, says Home Secretary. https://www.theregister.co.uk/2012/10/16/mckinnon_extradition_decision/. Accessed: Nov 2019.

[138] JOSH SOWALSKY. Exporting Topmeta. https://discovertext.com/tag/gnip-powertrack/. Accessed: Feb 2020.

[139] JUNG, J., AND SIT, E. An empirical study of spam traffic and the use of dns black lists. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement* (2004), ACM, pp. 370–375.

[140] JUNG, J., SIT, E., BALAKRISHNAN, H., AND MORRIS, R. DNS performance and the effectiveness of caching. *IEEE/ACM Transactions on networking 10*, 5 (2002), 589–603.

[141] JUNGHERR, A. Twitter use in election campaigns: A systematic literature review. *Journal of information technology & politics 13*, 1 (2016), 72–91.

[142] KANTCHELIAN, A., TSCHANTZ, M. C., AFROZ, S., MILLER, B., SHANKAR, V., BACHWANI, R., JOSEPH, A. D., AND TYGAR, J. D. Better malware ground truth: Techniques for weighting anti-virus vendor labels. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security* (2015), ACM, pp. 45–56.

[143] KAPRAVELOS, A., SHOSHITAISHVILI, Y., COVA, M., KRUEGEL, C., AND VIGNA, G. Revolver: An automated approach to the detection of evasive web-based malware. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)* (2013), pp. 637–652.

[144] KELLY, M. Politicians aren't "entirely" above the rules, Twitter says. https://www.theverge.com/2019/10/15/20916264/twitter-trump-policies-public-figures-interest-moderation-speech, 2019.

[145] KHONJI, M., IRAQI, Y., AND JONES, A. Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials 15*, 4 (2013), 2091–2121.

[146] KHONJI, M., JONES, A., AND IRAQI, Y. A novel Phishing classification based on URL features. In *2011 IEEE GCC conference and exhibition (GCC)* (2011), IEEE, pp. 221–224.

[147] KIRDA, E., AND KRUEGEL, C. Protecting users against phishing attacks with AntiPhish. In *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International* (2005), vol. 1, IEEE, pp. 517–524.

[148] KÜHRER, M., AND HOLZ, T. An empirical analysis of malware blacklists. *PIK-Praxis der Informationsverarbeitung und Kommunikation 35*, 1 (2012), 11–16.

[149] KÜHRER, M., ROSSOW, C., AND HOLZ, T. Paint it black: Evaluating the effectiveness of malware blacklists. In *International Workshop on Recent Advances in Intrusion Detection* (2014), Springer, pp. 1–21.

[150] KUMARAGURU, P. *PhishGuru: A System for Educating Users about Semantic Attacks*. Carnegie Mellon University, 2009.

[151] KUMARAGURU, P., SHENG, S., ACQUISTI, A., CRANOR, L. F., AND HONG, J. Teaching Johnny not to fall for phish. *ACM Transactions on Internet Technology (TOIT) 10*, 2 (2010), 7.

[152] LANDAGE, J., AND WANKHADE, M. Malware and malware detection techniques: A survey. *International Journal of Engineering Research and Technology (IJERT) 2*, 12 (2013), 2278–0181.

[153] LANDIS, J. R., AND KOCH, G. G. The measurement of observer agreement for categorical data. *Biometrics* (1977), 159–174.

[154] LAVASOFT. Ad-Aware. http://www.lavasoftusa.com/software/adaware/, 2018.

[155] LAZER, D. M., BAUM, M. A., BENKLER, Y., BERINSKY, A. J., GREENHILL, K. M., MENCZER, F., METZGER, M. J., NYHAN, B., PENNYCOOK, G., ROTHSCHILD, D., ET AL. The science of fake news. *Science 359*, 6380 (2018), 1094–1096.

[156] LEE, K., CAVERLEE, J., AND WEBB, S. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (2010), ACM, pp. 435–442.

[157] LEE, K., EOFF, B. D., AND CAVERLEE, J. Seven months with the devils: A long-term study of content polluters on Twitter. In *Fifth International AAAI Conference on Weblogs and Social Media* (2011).

[158] LEE, S., AND KIM, J. WarningBird: Detecting Suspicious URLs in Twitter Stream. In *NDSS* (2012), vol. 12, pp. 1–13.

[159] LELAND, W. E., TAQQU, M. S., WILLINGER, W., AND WILSON, D. V. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on networking 2*, 1 (1994), 1–15.

[160] LERNER, J. S., AND KELTNER, D. Beyond valence: Toward a model of emotion-specific influences on judgement and choice. *Cognition & emotion 14*, 4 (2000), 473–493.

[161] LI, F., HO, G., KUAN, E., NIU, Y., BALLARD, L., THOMAS, K., BURSZTEIN, E., AND PAXSON, V. Remedying web hijacking: Notification effectiveness and webmaster comprehension. In *Proceedings of the 25th International Conference on World Wide Web* (2016), International World Wide Web Conferences Steering Committee, pp. 1009–1019.

[162] LI, V. G., DUNN, M., PEARCE, P., MCCOY, D., VOELKER, G. M., AND SAVAGE, S. Reading the tea leaves: A comparative analysis of threat intelligence. In *28th USENIX Security Symposium (USENIX Security 19)* (Santa Clara, CA, Aug. 2019), USENIX Association, pp. 851–867.

[163] LIANG, B., SU, M., YOU, W., SHI, W., AND YANG, G. Cracking classifiers for evasion: a case study on the Google's phishing pages filter. In *Proceedings of the 25th International Conference on World Wide Web* (2016), International World Wide Web Conferences Steering Committee, pp. 345–356.

[164] LIN, J., SNOW, R., AND MORGAN, W. Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (2011), ACM, pp. 422–429.

[165] LISA FAZIO. Out-of-context photos are a powerful low-tech form of misinformation. https://theconversation.com/out-of-context-photos-are-a-powerful-low-tech-form-of-misinformation-129959. Accessed: Feb 2020.

[166] LIU, Y., KLIMAN-SILVER, C., AND MISLOVE, A. The tweets they are a-changin': Evolution of Twitter users and behavior. In *Eighth International AAAI Conference on Weblogs and Social Media* (2014).

[167] LOMBORG, S., AND BECHMANN, A. Using APIs for data collection on social media. *The Information Society 30*, 4 (2014), 256–265.

[168] LONGLEY, P. A., ADNAN, M., AND LANSLEY, G. The geotemporal demographics of Twitter usage. *Environment and Planning A 47*, 2 (2015), 465–484.

[169] LUDL, C., MCALLISTER, S., KIRDA, E., AND KRUEGEL, C. On the effectiveness of techniques to detect phishing sites. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2007), Springer, pp. 20–39.

[170] MA, J., SAUL, L. K., SAVAGE, S., AND VOELKER, G. M. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (2009), ACM, pp. 1245–1254.

[171] MAGGI, F., FROSSI, A., ZANERO, S., STRINGHINI, G., STONE-GROSS, B., KRUEGEL, C., AND VIGNA, G. Two years of short URLs internet measurement: security threats and countermeasures. In *proceedings of the 22nd international conference on World Wide Web* (2013), ACM, pp. 861–872.

[172] MARCHAL, S., SAARI, K., SINGH, N., AND ASOKAN, N. Know your phish: Novel techniques for detecting phishing sites and their targets. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)* (2016), IEEE, pp. 323–333.

[173] MARK SWENEY. Twitter revenue tops $1bn a quarter for first time. https://www.theguardian.com/business/2020/feb/06/twitter-revenue-tops-1bn-a-quarter-for-first-time. Accessed: Feb 2020.

[174] MARKMONITOR. Clarivate Analytics | Brand Protection, Domain Management, Anti Piracy, Anti Fraud. http://www.markmonitor.com.

[175] MARPAUNG, J. A., SAIN, M., AND LEE, H.-J. Survey on malware evasion techniques: State of the art and challenges. In *2012 14th International Conference on Advanced Communication Technology (ICACT)* (2012), IEEE, pp. 744–749.

[176] MARTIN, W., HARMAN, M., JIA, Y., SARRO, F., AND ZHANG, Y. The app sampling problem for app store mining. In *Proceedings of the 12th Working Conference on Mining Software Repositories* (2015), IEEE Press, pp. 123–133.

[177] MATHIOUDAKIS, M., AND KOUDAS, N. TwitterMonitor: trend detection over the Twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (2010), ACM, pp. 1155–1158.

[178] MC LAUGHLIN, G. H. SMOG grading-a new readability formula. *Journal of reading 12*, 8 (1969), 639–646.

[179] MCCORD, M., AND CHUAH, M. Spam detection on Twitter using traditional classifiers. In *international conference on Autonomic and trusted computing* (2011), Springer, pp. 175–186.

[180] McGrath, D. K., and Gupta, M. Behind phishing: An examination of phisher modi operandi. *LEET 8* (2008), 4.

[181] Medvet, E., Kirda, E., and Kruegel, C. Visual-similarity-based phishing detection. In *Proceedings of the 4th international conference on Security and privacy in communication netowrks* (2008), ACM, p. 22.

[182] Melissa Lemieux. Infamous Teenage U.K. Hacker Accused of Computer Fraud by U.S. Authorities. https://www.newsweek.com/infamous-teenage-uk-hacker-accused-computer-fraud-us-authorities-1461182. Accessed: Jan 2020.

[183] Metcalf, L., and Spring, J. M. Blacklist ecosystem analysis: spanning Jan 2012 to Jun 2014. In *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security* (2015), ACM, pp. 13–22.

[184] Mike langberg. AOL acts to thwart hackers. [https://simson.net/clips/1995/95.SJMN.AOL_Hackers.html; accessed Feb 2020].

[185] Ministry of Justice. Outcomes by Offence data tool; 2017. https://www.gov.uk/government/statistics/criminal-justice-system-statistics-quarterly-december-2017, 2017.

[186] Ministry of Justice. Outcomes by Offence data tool. https://www.gov.uk/government/statistics/criminal-justice-system-statistics-quarterly-december-2018, 2018.

[187] Moore, D., Shannon, C., Brown, D. J., Voelker, G. M., and Savage, S. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems (TOCS) 24*, 2 (2006), 115–139.

[188] Moore, T., and Clayton, R. An empirical analysis of the current state of phishing attack and defence. In *WEIS* (2007).

[189] Moore, T., and Clayton, R. Examining the impact of website take-down on phishing. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit* (2007), ACM, pp. 1–13.

[190] Moore, T., and Clayton, R. The consequence of non-cooperation in the fight against phishing. In *2008 eCrime Researchers Summit* (2008), IEEE, pp. 1–14.

[191] Moore, T., and Clayton, R. Evaluating the wisdom of crowds in assessing phishing websites. In *International Conference on Financial Cryptography and Data Security* (2008), Springer, pp. 16–30.

[192] Moore, T., and Clayton, R. Evil searching: Compromise and recompromise of internet hosts for phishing. In *International Conference on Financial Cryptography and Data Security* (2009), Springer, pp. 256–272.

[193] Moore, T., and Clayton, R. How hard can it be to measure phishing? *Mapping and Measuring Cybercrime* (2010).

[194] Moore, T., Clayton, R., and Stern, H. Temporal correlations between spam and phishing websites. In *LEET* (2009).

[195] Morstatter, F., Pfeffer, J., and Liu, H. When is it biased?: Assessing the representativeness of Twitter's streaming API. In *Proceedings of the 23rd international conference on world wide web* (2014), ACM, pp. 555–556.

[196] MORSTATTER, F., PFEFFER, J., LIU, H., AND CARLEY, K. M. Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In *ICWSM* (2013).

[197] MOSHCHUK, A., BRAGIN, T., GRIBBLE, S. D., AND LEVY, H. M. A crawler-based study of spyware in the web. In *NDSS* (2006), vol. 1, p. 2.

[198] MOTOYAMA, M., MCCOY, D., LEVCHENKO, K., SAVAGE, S., AND VOELKER, G. M. Dirty jobs: The role of freelance labor in web service abuse. In *Proceedings of the 20th USENIX conference on Security* (2011), USENIX Association, pp. 14–14.

[199] MSN. Hurricane Sandy Fake Photos. http://now.msn.com/hurricane-sandy-fake-photos.

[200] NARAINE, R. Twitter Now Filtering Malicious URLs. https://archive.f-secure.com/weblog/archives/00001745.html, 2009.

[201] NARAINE, R. Twitter turns to Google for help with malware attacks. http://www.zdnet.com/article/twitter-turns-to-google-for-help-with-malware-attacks/, 2009.

[202] NATIONAL CRIME AGENCY. Cyber crime. https://www.nationalcrimeagency.gov.uk/what-we-do/crime-threats/cyber-crime. Accessed: Nov 2019.

[203] NAZARIO, J. PhishingCorpus. http://monkey.org/~jose/wiki/doku.php?id=PhishingCorpus, 2011.

[204] NDIBWILE, J. D., LUHANGA, E. T., FALL, D., MIYAMOTO, D., BLANC, G., AND KADOBAYASHI, Y. An empirical approach to phishing countermeasures through smart glasses and validation agents. *IEEE Access 7* (2019), 130758–130771.

[205] NETSCAPE. Open directory project. http://www.dmoz.org.

[206] NEUPANE, A., RAHMAN, M. L., SAXENA, N., AND HIRSHFIELD, L. A multi-modal neuro-physiological study of phishing detection and malware warnings. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2015), CCS '15, ACM, pp. 479–491.

[207] NEWMAN, L. H. How Google's Safe Browsing Helped Build a More Secure Web. https://www.wired.com/story/google-safe-browsing-oral-history/, 2018.

[208] NEWMAN, M. E. Power laws, Pareto distributions and Zipf's law. *Contemporary physics 46*, 5 (2005), 323–351.

[209] NGUYEN, L. A. T., TO, B. L., NGUYEN, H. K., AND NGUYEN, M. H. Detecting phishing web sites: A heuristic URL-based approach. In *2013 International Conference on Advanced Technologies for Communications (ATC 2013)* (2013), IEEE, pp. 597–602.

[210] NGUYEN, L. A. T., TO, B. L., NGUYEN, H. K., AND NGUYEN, M. H. A novel approach for phishing detection using URL-based heuristic. In *2014 International Conference on Computing, Management and Telecommunications (ComManTel)* (2014), IEEE, pp. 298–303.

[211] NIKIFORAKIS, N., MAGGI, F., STRINGHINI, G., RAFIQUE, M. Z., JOOSEN, W., KRUEGEL, C., PIESSENS, F., VIGNA, G., AND ZANERO, S. Stranger danger: exploring the ecosystem of ad-based URL shortening services. In *Proceedings of the 23rd international conference on World wide web* (2014), ACM, pp. 51–62.

[212] NILIZADEH, S., LABRÈCHE, F., SEDIGHIAN, A., ZAND, A., FERNANDEZ, J., KRUEGEL, C., STRINGHINI, G., AND VIGNA, G. Poised: Spotting Twitter spam off the beaten paths. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), ACM, pp. 1159–1174.

[213] NIU, Y., HSU, F., AND CHEN, H. iPhish: Phishing Vulnerabilities on Consumer Electronics. In *Proceedings of the 1st Conference on Usability, Psychology, and Security* (Berkeley, CA, USA, 2008), UPSEC'08, USENIX Association, pp. 10:1–10:8.

[214] OLIVER, J., PAJARES, P., KE, C., CHEN, C., AND XIANG, Y. An in-depth analysis of abuse on Twitter. *Trend Micro 225* (2014), 1–22.

[215] OPENPHISH. Phishing Intelligence Blacklist. https://openphish.com/.

[216] ORACLE. MySQL Database. https://www.mysql.com/, 2018.

[217] OSBORNE, M., AND DREDZE, M. Facebook, Twitter and Google Plus for Breaking News: Is There a Winner? In *Eighth International AAAI Conference on Weblogs and Social Media* (2014).

[218] OWEN BOWCOTT. Cybercrime laws need urgent reform to protect UK, says report. https://www.theguardian.com/technology/2020/jan/22/cybercrime-laws-need-urgent-reform-to-protect-uk-says-report. Accessed: Feb 2020.

[219] PAN, Y., AND DING, X. Anomaly based web phishing page detection. In *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)* (2006), IEEE, pp. 381–392.

[220] PARNO, B., KUO, C., AND PERRIG, A. Phoolproof phishing prevention. In *Financial Cryptography* (2006), vol. 4107, Springer, pp. 1–19.

[221] PAXSON, V. *Measurements and analysis of end-to-end Internet dynamics*. PhD thesis, University of California, Berkeley, 1997.

[222] PAXSON, V. Strategies for sound internet measurement. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement* (2004), pp. 263–271.

[223] PENG, P., YANG, L., SONG, L., AND WANG, G. Opening the blackbox of virustotal: Analyzing online phishing scan engines. In *Proceedings of the Internet Measurement Conference* (2019), ACM, pp. 478–485.

[224] PERLROTH, N. Fake Twitter Followers Become Multimillion-Dollar Business. https://bits.blogs.nytimes.com/2013/04/05/fake-twitter-followers-becomes-multimillion-dollar-business/, 2013.

[225] PHANTOMJS. Scriptable Headless Browser. https://phantomjs.org. Accessed: Nov 2019.

[226] PHISHTANK. Phishing Blacklist | Join the fight against phishing. https://www.phishtank.com/.

[227] PHISHTANK. Friends of PhishTank. https://www.phishtank.com/friends.php, 2018.

[228] PITSILLIDIS, A., KANICH, C., VOELKER, G. M., LEVCHENKO, K., AND SAVAGE, S. Taster's choice: a comparative analysis of spam feeds. In *Proceedings of the 2012 Internet Measurement Conference* (2012), ACM, pp. 427–440.

[229] PRAKASH, P., KUMAR, M., KOMPELLA, R. R., AND GUPTA, M. Phishnet: predictive blacklisting to detect phishing attacks. In *2010 Proceedings IEEE INFOCOM* (2010), IEEE, pp. 1–5.

[230] PRIESTMAN, W., ANSTIS, T., SEBIRE, I. G., SRIDHARAN, S., AND SEBIRE, N. J. Phishing in healthcare organisations: threats, mitigation and approaches. *BMJ health & care informatics 26*, 1 (2019).

[231] PRINCE, M. B., DAHL, B. M., HOLLOWAY, L., KELLER, A. M., AND LANGHEINRICH, E. Understanding How Spammers Steal Your E-Mail Address: An Analysis of the First Six Months of Data from Project Honey Pot. In *CEAS* (2005).

[232] PROVOS, N., MAVROMMATIS, P., RAJAB, M. A., AND MONROSE, F. All Your iFRAMEs Point to Us. In *Proceedings of the 17th Conference on Security Symposium* (Berkeley, CA, USA, 2008), SS'08, USENIX Association, pp. 1–15.

[233] PROVOS, N., MCNAMEE, D., MAVROMMATIS, P., WANG, K., MODADUGU, N., ET AL. The ghost in the browser: Analysis of web-based malware. *HotBots 7* (2007), 4–4.

[234] PURKAIT, S. Phishing counter measures and their effectiveness – literature review. *Information Management & Computer Security 20*, 5 (2012), 382–420.

[235] PYTHON SOFTWARE FOUNDATION. Requests: HTTP for Humans. http://docs.python-requests.org/, 2018.

[236] QUINLAN, J. R. *C4. 5: Programs for Machine Learning*. Elsevier, 2014.

[237] RAMACHANDRAN, A., FEAMSTER, N., AND VEMPALA, S. Filtering spam with behavioral blacklisting. In *Proceedings of the 14th ACM conference on Computer and communications security* (2007), pp. 342–351.

[238] RATKIEWICZ, J., CONOVER, M., MEISS, M., GONÇALVES, B., PATIL, S., FLAMMINI, A., AND MENCZER, F. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th international conference companion on World wide web* (2011), ACM, pp. 249–252.

[239] REKOUCHE, K. Early phishing. *arXiv preprint arXiv:1106.4692* (2011).

[240] RODRIGUES, T., BENEVENUTO, F., CHA, M., GUMMADI, K., AND ALMEIDA, V. On word-of-mouth based discovery of the web. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference* (2011), ACM, pp. 381–396.

[241] ROSIELLO, A. P., KIRDA, E., KRUEGEL, C., AND FERRANDI, F. A layout-similarity-based approach for detecting phishing pages. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on* (2007), IEEE, pp. 454–463.

[242] ROYAL HOLLOWAY, UNIVERSITY OF LONDON. Guidelines on Research Governance, Research Ethics and Good Research Practice. [https://intranet.royalholloway.ac.uk/iquad/collegepolicies/documents/pdf/research/codeofgoodresearchpractice.pdf; accessed Feb 2020].

[243] SAEED, I. A., SELAMAT, A., AND ABUAGOUB, A. M. A survey on malware and malware detection systems. *International Journal of Computer Applications 67*, 16 (2013).

[244] SANGLERDSINLAPACHAI, N., AND RUNGSAWANG, A. Using domain top-page similarity feature in machine learning-based web phishing detection. In *2010 Third International Conference on Knowledge Discovery and Data Mining* (2010), IEEE, pp. 187–190.

[245] SAUCEZ, D., AND IANNONE, L. Thoughts and Recommendations from the ACM SIGCOMM 2017 Reproducibility Workshop. *ACM SIGCOMM Computer Communication Review 48*, 1 (2018), 70–74.

[246] SCHEITLE, Q., WÄHLISCH, M., GASSER, O., SCHMIDT, T. C., AND CARLE, G. Towards an ecosystem for reproducible research in computer networking. In *Proceedings of the Reproducibility Workshop* (2017), pp. 5–8.

[247] SCHRYEN, G. The impact that placing email addresses on the internet has on the receipt of spam: An empirical analysis. *computers & security 26*, 5 (2007), 361–372.

[248] SHANNON, C., MOORE, D., KEYS, K., FOMENKOV, M., HUFFAKER, B., AND CLAFFY, K. The internet measurement data catalog. *ACM SIGCOMM Computer Communication Review 35*, 5 (2005), 97–100.

[249] SHENG, S., HOLBROOK, M., KUMARAGURU, P., CRANOR, L. F., AND DOWNS, J. Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 373–382.

[250] SHENG, S., MAGNIEN, B., KUMARAGURU, P., ACQUISTI, A., CRANOR, L. F., HONG, J., AND NUNGE, E. Anti-phishing Phil: the design and evaluation of a game that teaches people not to fall for phish. In *Proceedings of the 3rd symposium on Usable privacy and security* (2007), ACM, pp. 88–99.

[251] SHENG, S., WARDMAN, B., WARNER, G., CRANOR, L. F., HONG, J., AND ZHANG, C. An empirical analysis of phishing blacklists. *Proceedings of Sixth Conference on Email and Anti-Spam (CEAS)* (2009).

[252] SHEPHERD, S. Vulnerability disclosure: How do we define responsible disclosure? *GIAC SEC Practical Repository, SANS Inst 9* (2003).

[253] SHUE, C. A., KALAFUT, A. J., AND GUPTA, M. Exploitable redirects on the web: Identification, prevalence, and defense. In *WOOT* (2008).

[254] SINHA, S., BAILEY, M., AND JAHANIAN, F. Shades of grey: On the effectiveness of reputation-based "blacklists". In *2008 3rd International Conference on Malicious and Unwanted Software (MALWARE)* (2008), IEEE, pp. 57–64.

[255] SQLITE CONSORTIUM. SQLite Database. https://www.sqlite.org/.

[256] STAJANO, F., AND WILSON, P. Understanding scam victims: seven principles for systems security. Tech. rep., University of Cambridge, Computer Laboratory, 2009.

[257] STATCOUNTER. Desktop Browser Market Share Worldwide - Jan 2009 to Jan 2020. [https://gs.statcounter.com/browser-market-share/desktop/worldwide/#monthly-200901-202001; accessed Feb 2020].

[258] STATCOUNTER. Operating System Market Share Worldwide - Jan 2009 to Jan 2020. [https://gs.statcounter.com/os-market-share#monthly-200901-202001; accessed Feb 2020].

[259] STATISTA. Number of monthly active Twitter users worldwide from 1st quarter 2010 to 4th quarter 2017 (in millions). https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/, 2018.

[260] STEVEN POOLE. Before Trump: the real history of fake news. https://www.theguardian.com/books/2019/nov/22/factitious-taradiddle-dictionary-real-history-fake-news. Accessed: Jan 2020.

[261] STRINGHINI, G., EGELE, M., KRUEGEL, C., AND VIGNA, G. Poultry markets: on the underground economy of Twitter followers. *ACM SIGCOMM Computer Communication Review 42*, 4 (2012), 527–532.

[262] STRINGHINI, G., KRUEGEL, C., AND VIGNA, G. Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference* (2010), ACM, pp. 1–9.

[263] STRINGHINI, G., WANG, G., EGELE, M., KRUEGEL, C., VIGNA, G., ZHENG, H., AND ZHAO, B. Y. Follow the green: growth and dynamics in Twitter follower markets. In *Proceedings of the 2013 conference on Internet measurement conference* (2013), ACM, pp. 163–176.

[264] TADEK PIETRASZEK BORBALA BENKO, E. B., AND RISHER, M. Cleaning up after password dumps. https://security.googleblog.com/2014/09/cleaning-up-after-password-dumps.html, 2014.

[265] TALLY, G. Phisherman: A Phishing Data Repository. In *Cybersecurity Applications & Technology Conference for Homeland Security* (2009), IEEE, pp. 155–160.

[266] THOMAS, D. R., PASTRANA, S., HUTCHINGS, A., CLAYTON, R., AND BERESFORD, A. R. Ethical issues in research using datasets of illicit origin. In *Proceedings of the 2017 Internet Measurement Conference* (2017), ACM, pp. 445–462.

[267] THOMAS, K., AMIRA, R., BEN-YOASH, A., FOLGER, O., HARDON, A., BERGER, A., BURSZTEIN, E., AND BAILEY, M. The abuse sharing economy: Understanding the limits of threat exchanges. In *International Symposium on Research in Attacks, Intrusions, and Defenses* (2016), Springer, pp. 143–164.

[268] THOMAS, K., GRIER, C., MA, J., PAXSON, V., AND SONG, D. Design and Evaluation of a Real-Time URL Spam Filtering Service. In *2011 IEEE symposium on security and privacy* (2011), IEEE, pp. 447–462.

[269] THOMAS, K., GRIER, C., SONG, D., AND PAXSON, V. Suspended accounts in retrospect: an analysis of Twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference* (2011), ACM, pp. 243–258.

[270] THOMAS, K., LI, F., ZAND, A., BARRETT, J., RANIERI, J., INVERNIZZI, L., MARKOV, Y., COMANESCU, O., ERANTI, V., MOSCICKI, A., ET AL. Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (2017), ACM, pp. 1421–1434.

[271] THOMAS, K., MCCOY, D., GRIER, C., KOLCZ, A., AND PAXSON, V. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)* (2013), pp. 195–210.

[272] TIEDENS, L. Z., AND LINTON, S. Judgment under emotional certainty and uncertainty: the effects of specific emotions on information processing. *Journal of personality and social psychology 81*, 6 (2001), 973.

[273] TREND MICRO. Site Safety Center. http://global.sitesafety.trendmicro.com/. Accessed: Nov 2019.

[274] TWEEPY. Tweepy: An easy-to-use Python library for accessing the Twitter API. http://www.tweepy.org/.

[275] TWITTER. Giving you more characters to express yourself. https://blog.twitter.com/official/en_us/topics/product/2017/Giving-you-more-characters-to-express-yourself.html. Accessed: Nov 2019.

[276] TWITTER. t.co Links. https://developer.twitter.com/en/docs/basics/tco. Accessed: Nov 2018.

[277] TWITTER. Terms of Service. https://twitter.com/en/tos. Accessed: Feb 2020.

[278] TWITTER. The Twitter Rules. https://twitter.com/rules.

[279] TWITTER. Links and Twitter: Length Shouldn't Matter. https://blog.twitter.com/en_us/a/2010/links-and-twitter-length-shouldn-t-matter.html, 2010. Accessed: Nov 2019.

[280] TWITTER. World Leaders on Twitter: principles & approach. https://blog.twitter.com/official/en_us/topics/company/2019/worldleaders2019.html, 2019.

[281] TWITTER, INC. Investor Relations. https://investor.twitterinc.com/home/default.aspx. Accessed: March 2020.

[282] TWITTERCOUNTER. Twitter Top 100 Most Followers. https://twittercounter.com/pages/100.

[283] UNSPAM TECHNOLOGIES. Project Honey Pot: Help stop spammers before they even get your address. https://www.projecthoneypot.org/.

[284] UPTON, E., AND HALFACREE, G. *Raspberry Pi user guide*. John Wiley & Sons, 2014.

[285] URIBL. Realtime URI blacklist. http://uribl.com/.

[286] VIS, F. A critical reflection on Big Data: Considering APIs, researchers and tools as data makers. *First Monday 18*, 10 (2013).

[287] W3COUNTER. Web Browser Usage Trends. [https://www.w3counter.com/trends; accessed Feb 2020].

[288] WANG, A. H. Don't follow me: Spam detection in Twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on* (2010), IEEE, pp. 1–10.

[289] WANG, D., NAVATHE, S. B., LIU, L., IRANI, D., TAMERSOY, A., AND PU, C. Click traffic analysis of short URL spam on Twitter. In *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing* (2013), IEEE, pp. 250–259.

[290] WANG, G., KONOLIGE, T., WILSON, C., WANG, X., ZHENG, H., AND ZHAO, B. Y. You are how you click: Clickstream analysis for sybil detection. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)* (2013), pp. 241–256.

[291] WANG, G., WILSON, C., ZHAO, X., ZHU, Y., MOHANLAL, M., ZHENG, H., AND ZHAO, B. Y. Serf and turf: crowdturfing for fun and profit. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 679–688.

[292] WEB OF TRUST. MyWOT API. http://www.mywot.com/wiki/API.

[293] WEBB, H., JIROTKA, M., STAHL, B. C., HOUSLEY, W., EDWARDS, A., WILLIAMS, M., PROCTER, R., RANA, O., AND BURNAP, P. The Ethical Challenges of Publishing Twitter Data for Research Dissemination. In *Proceedings of the 2017 ACM on Web Science Conference* (2017), pp. 339–348.

[294] WEBPRONEWS. Google Discusses Its Safe Browsing Record. https://www.webpronews.com/google-discusses-its-safe-browsing-record-2012-06/, 2012.

[295] WEEK, THE. 10 fake photos of Hurricane Sandy. https://theweek.com/articles/470936/10-fake-photos-hurricane-sandy. Accessed: Feb 2020.

[296] WEIN, J. Joewein.de LLC: fighting spam and scams on the Internet. http://www.joewein.net/, 2018.

[297] WENYIN, L., HUANG, G., XIAOYUE, L., MIN, Z., AND DENG, X. Detection of phishing webpages based on visual similarity. In *Special interest tracks and posters of the 14th international conference on World Wide Web* (2005), ACM, pp. 1060–1061.

[298] WHITTAKER, C., RYNER, B., AND NAZIF, M. Large-scale automatic classification of phishing pages. In *NDSS* (2010).

[299] WITTEN, I. H., AND FRANK, E. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[300] WU, M., MILLER, R. C., AND GARFINKEL, S. L. Do security toolbars actually prevent phishing attacks? In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 601–610.

[301] XIANG, G., HONG, J., ROSE, C. P., AND CRANOR, L. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC) 14*, 2 (2011), 21.

[302] XIANG, G., AND HONG, J. I. A hybrid phish detection approach by identity discovery and keywords retrieval. In *Proceedings of the 18th international conference on World wide web* (2009), ACM, pp. 571–580.

[303] YAN, L., AND MCKEOWN, N. Learning networking by reproducing research results. *ACM SIGCOMM Computer Communication Review 47*, 2 (2017), 19–26.

[304] YANG, C., HARKREADER, R., AND GU, G. Empirical evaluation and new design for fighting evolving Twitter spammers. *IEEE Transactions on Information Forensics and Security 8*, 8 (2013), 1280–1293.

[305] YANG, C., HARKREADER, R., ZHANG, J., SHIN, S., AND GU, G. Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on Twitter. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 71–80.

[306] YANG, C., HARKREADER, R. C., AND GU, G. Die Free or Live Hard? Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. In *International Workshop on Recent Advances in Intrusion Detection* (2011), Springer, pp. 318–337.

[307] YARDI, S., ROMERO, D., SCHOENEBECK, G., ET AL. Detecting spam in a Twitter network. *First Monday 15*, 1 (2010).

[308] YU, K., TAIB, R., BUTAVICIUS, M. A., PARSONS, K., AND CHEN, F. Mouse behavior as an index of phishing awareness. In *IFIP Conference on Human-Computer Interaction* (2019), Springer, pp. 539–548.

[309] ZHANG, C. M., AND PAXSON, V. Detecting and analyzing automated activity on Twitter. In *International Conference on Passive and Active Network Measurement* (2011), Springer, pp. 102–111.

[310] ZHANG, J., CHIVUKULA, A., BAILEY, M., KARIR, M., AND LIU, M. Characterization of blacklists and tainted network traffic. In *International Conference on Passive and Active Network Measurement* (2013), Springer, pp. 218–228.

[311] ZHANG, Y., EGELMAN, S., CRANOR, L., AND HONG, J. Phinding phish: Evaluating anti-phishing tools. In *Tech Report: CMU-CyLab-06-018* (2006), ISOC.

[312] ZHANG, Y., HONG, J. I., AND CRANOR, L. F. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 639–648.