The Role of Hosting Providers in Web Security

Understanding and Improving Security Incentives and Performance via Analysis of Large-scale Incident Data

Tajalizadehkhoob, Samaneh

DOI
10.4233/uuid:c343a2dd-15d1-4921-9b45-f00ee38177d8

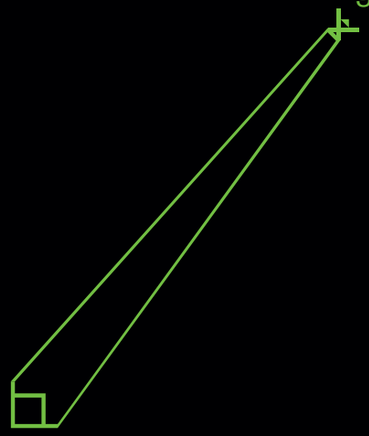
Publication date
2018

Document Version
Final published version

Citation (APA)
Tajalizadehkhoob, S. (2018). *The Role of Hosting Providers in Web Security: Understanding and Improving Security Incentives and Performance via Analysis of Large-scale Incident Data*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:c343a2dd-15d1-4921-9b45-f00ee38177d8

In theory, hosting providers can play an important role in fighting cybercrime and misuse. This is because many online threats, be they high-profile or mundane, use online storage infrastructure maintained by hosting providers at the core of their criminal operations.
However, in practice, we see large differences in the security measures taken by hosting providers. Some providers implement an array of actions to protect their customers. Others lack even the capacity to detect cybercrime, are negligent of cybercrime, or even willfully facilitate it.

This book answers a series of questions that collectively aim to understand the underlying differences in security incentives and policies of hosting providers: How do we define a hosting provider? How are they distributed? To what extent do their individual properties or security measures affect the volume of incident in their networks?
We expect this book to provide useful insights for hosting providers about the effectiveness of their security policies and to serve as a an input for development of evidence-based policies by the government.



TU Delft

The Role of Hosting Providers in Web Security

Samaneh Tajalizadehkhoob

<h1 class="title">

The Role of Hosting Providers in Web Security:

Understanding and Improving Security Incentives and Performance via Analysis of Large-scale Incident Data

</h1>

<h2 class="author">

Samaneh Tajalizadehkhoob

</h2>

The Role of Hosting Providers in Web Security

Understanding and Improving Security
Incentives and Performance
via Analysis of Large-scale Incident Data

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
Maandag 5 februari 2018 om 10.00 uur

door

Samaneh TAJALIZADEHKHOOB

Master of Science in Engineering and Policy Analysis
geboren te Teheran, Iran

This dissertation has been approved by the promotores:
Prof.dr. M.J.G van Eeten
Prof.dr. M.L.P. Groenleer

Composition of the doctoral committee:

Rector Magnificus	Chairman
Prof.dr. M.J.G van Eeten	Promoter, TU Delft
Prof.dr. M.L.P. Groenleer	Promoter, Tilburg University

Independent members:

Prof.dr.ir. J. van den Berg	Faculty of TPM, TU Delft
Prof.dr. P.H. Hartel	Faculty of EEMCS, TU Delft
Prof.dr.ir. H.J. Bos	Vrije Universiteit Amsterdam
Prof.dr. R. Anderson	University of Cambridge
Dr. P. Vixie	Farsight Security

This research has been funded by NWO (grant nr. 12.003/628.001.003), the National Cyber Security Center (NCSC) and SIDN, the .NL Registry.

Cover design: Amir Hossein Farahani.
Printed in the Netherlands by Gildeprint.

Distributed by Delft University of Technology, Faculty of Technology, Policy and Management, Jaffalaan 5, 2628BX Delft, the Netherlands.

ISBN 978-94-6366-007-5

Keywords: cybersecurity, hosting provider, metrics, incentives, shared hosting, patching, vulnerability scan, data analysis, statistical models, machine learning, blacklist data.

Dedicated to Ziba, Parviz, and Anton

"The important work of moving the world forward does not wait to be done by perfect men." [George Eliot, 1858]

Acknowledgments

Albeit my name is standing alone on the front cover of this dissertation, I am not its only contributor. Rather, there are a number of people behind this piece of work who I would like to acknowledge and thank here.

While I am quite sure my words can hardly capture the role that my first promoter, Michel van Eeten, played in my academic and personal life, I would still like to try. Michel has been my mentor and supervisor, since my master thesis. During this period, his constant courage and unflagging support of one sort or another has played a significant role not only in achieving the result of this dissertation but also in shaping the person I am today. He helped me learning how to be more confident and defend my point of view as an independent researcher. He has spent limitless hours brainstorming ideas with me, reading my rather rough manuscripts, and clarifying my arguments throughout our submissions. Michel, thank you for your enthusiasm, guidance and your friendship throughout this process.

I would like to thank Martijn Groenleer, my second promoter who has provided invaluable outsiders perspectives on my research when most needed. Thank you for all the insights on the policy and governance side of my research and thank you for all the hours you have spent on providing many valuable suggestions that improved this manuscript.

I am thankful for having had Maciej Korczyński as my daily-supervisor. He has played a significant role in my academic development. Maciej thank you for setting me off on further roads; you rock. With the same token, a special thank to Rainer Böhme and Tyler Moore for their significant contribution to my learning process. I have enjoyed the opportunity to watch and learn from their knowledge and experience. Their frequent insights and patience with me has been invaluable.

I like to express my gratitude for Carlos, whose constant hands-on scientific help and support, especially in the early years of my PhD project helped me a lot to get the ball of the PhD rolling. Many of my other colleagues have shared a lot with me, have helped me and inspired me in ways that I can't possibly

elaborate on in these lines over the past years: Ardalan, Arman, Behnam Taebi, Femke, Kaveri, Giovane, Qasim, Michael, Orcun, Rene, Rolf, and the members of TBM PhD council. Thank you all. Finally, I would like to thank SIDN and Farsight Security for their useful inputs and continues support for our project over the past four years.

Given that pursuing a PhD consumes a great deal of time from PhD candidates, the little time that is left outside of the university is all one has to fill supported, loved, and inspired. Having moved to another country makes it even harder, particularly because one has to build everything from scratch. I however, have been extremely lucky for having friends who have supported me one way or another during this journey: Ario, Armin, Behzad, Farshad, Ghazaleh, Hamed, Kianoush, Leila, Mahtab, Nami, Pantea, Pejman, Samar, Saba, Sepideh, Siamak, and Shahob. Thank you all. Delaram, Dena, Nadjla, and Ghazaleh, my partners in crime, we have laughed/cried this journey together so far. Thank you for supporting me in difficult and happy moments, for listening to all of my frustrations, and inspiring me everyday towards wanting to be better.

I would like to thank my partner Anton, for his unrelenting encouragement. Put simply, there is no one who believes in me more. Thank you for making me more than I am. I am also thankful for all the people in your life who welcomed me into theirs. Being far from my family has been a lot easier thanks to the love and support I have been receiving from my extended family Azadeh, Mansour, Els, Kees, Maarten, and Victoria here in the Netherlands. Els, I am grateful for all the effort you have put in translating the summary of this dissertation to Dutch.

To my life-coaches, my parents, brother, and sister: because I owe it all to you and the compromises you have made over these years.

Contents

CHAPTER 1

Introduction

1.1 Background

1.1.1 Internet security

In early 2017, Google reported a spear phishing scam in which victims received an email that appeared to be from someone they knew. Opening a link in the email led to a fraudulent website, hosted by a legitimate hosting provider, identical to Google's log-in and permissions page. This harvested all of the log-in details entered by victims, in addition to installing malware on their devices [1].

At about the same time, China Digital Times (CDT) employees received an email from someone purporting to be a UC Berkeley student. The email contained a link to a fake CDT website, designed to redirect users to a WordPress log-in phishing page. The page was used both to harvest employees' personal information and to distribute NetWire malware [2].

Just months later, a network of compromised Internet-of-Things (IoT) devices launched the largest denial-of-service attack ever recorded. Hackers used a variant of the 'Mirai' malware to compromise the home routers of German Internet Service Provider (ISP) Deutsche Telekom. More than 900,000 customers suffered outages as a result. The command-and-control (C&C) servers used to control the Mirai botnet were hosted at 23Media GmbH, a legitimate hosting company [3].

As the cases above demonstrate, Internet infrastructure, in addition to facilitating communication and data sharing for users around the world, also serves as a platform for fraud and misuse. Cybercriminals exploit the global web infrastructure for personal and financial gain. They devise ways to compromise servers and web domains via technical vulnerabilities in systems or human mistakes. Phishing, stealing online banking information, and malware distribution are but a few examples. These malicious practices not only harm individuals,

but also generate wider economic impacts, hurting society as whole.

More than 86,000 vulnerabilities has been reported between 2000 and 2017 [4] and different counter measures have been employed by now. Security companies and researchers dedicated a significant amount of research on identifying and mitigating vulnerabilities present in servers of Internet hosts. However attackers are always one step ahead and discover new vulnerabilities to exploit.

Research on mitigating cybersecurity problems has also focused on the role of end-users (the victims), criminals, or even law enforcement. However, countermeasures addressing end-users directly, such as user awareness-raising and information campaigns, have proven to have limited effectiveness. Users remain the weakest link and hence a major factor in security breaches [5].

Therefore, research has shifted focus to the role of Internet intermediaries in reducing cybercrime [6]. Internet intermediaries play a growing role in shaping the online economy, according to national and international organizations such as the Organization for Economic Co-operation and Development (OECD) and the European Union Agency for Network and Information Security (ENISA) [7, 8]. Examples of these intermediaries are ISPs, social network operators, payment service providers, and hosting providers.

1.1.2 The role of hosting providers in web security

The criminal activities introduced above have one thing in common: they all utilized hosting-provider operated infrastructure, such as servers and websites, to perform the online attacks.

Hosting providers are a key Internet intermediary. These companies "offer end users the ability to create their web presence on hardware they do not actually own[1]" [9]. They provide and facilitate infrastructure for storing and hosting online content. `Go daddy`, `Leaseweb`, and `OVH hosting` are a few well-known hosting providers.

Hosting providers *can* play an important role in fighting cybercrime and misuse [9]. This is because many online threats, be they high-profile or mundane, use hosting infrastructure at the core of their criminal operation. Think of selling stolen credit cards, publishing materials showing child sexual abuse, running C&C servers for botnets, and phishing for personal information. All these crimes use online storage space maintained and offered by hosting providers. Sometimes existing legitimate websites are compromised for illicit purposes, or new websites may be registered solely for criminal gain.

[1] We will reflect on this definition in more detail in chapter 2.

Depending on the hosting type and distribution of administrative rights, hosting providers may be responsible for assisting their customers maintain the security of the infrastructure they rent. In theory, a web hosting provider can provide critical proactive and reactive security support. For example, providers can act *proactively* by regularly patching their systems and applications. Or they can be *reactive*, taking down websites when they discover them to be compromised or when third parties notify them of malicious activity. Hosting providers that offer domain name registration in addition to hosting can influence domain registration processes. Specifically, they can suspend a domain if it is used for malicious purposes.

In practice, however, thousands of providers are associated with enabling online crime on a daily basis, wittingly or unwittingly. Providers are relatively free to determine how much to invest in their own security practices. We therefore see large differences in the security measures taken by hosting providers. Some providers take an array of actions to protect their customers. Others lack the capacity to detect cybercrime, are *negligent* of cybercrime, or even *facilitate* it.

Canali et al. found that some hosting providers were unable to detect basic attacks against their networks [10]. A major reason was the difficulty providers faced in adopting effective security practices in highly price-competitive markets. The so-called *bulletproof* hosting providers are an example of those that are negligent of or facilitate cybercrime. They are known for their leniency in the face of malicious content in their network [11, 12]. Often, however, it is difficult to distinguish between providers that deliberately facilitate malicious activities and those that are incapable of detecting abuse. Proving that a provider is unwilling to detect abuse, rather than unable, is even more difficult to do.

Given the magnitude of the cybersecurity threats we see every day, it is clear that the hosting market is not performing well in terms of cybersecurity. It is therefore legitimate to inquire into what hosting providers are already doing, what they could do better, and what others could do to incentivize them to achieve higher levels of security.

1.1.3 The economics of security in the hosting market

Ensuring and improving security in the hosting market has been a major challenge so far. But why? What characteristics of hosting providers contribute to insecurity in this market?

We start by addressing two general characteristics that are rather similar across all Internet intermediaries: *negative externalities*, and *information asym-*

metry that could cause *misaligned incentives*. These explain to some extent why cybersecurity problems have not yet diminished, despite the technical solutions available.

The literature on the economics of information security presents cybersecurity as an issue of misaligned incentives among the key actors involved, though technical issues are also recognized as playing a role [13]. Thus, actors with the technical knowledge who can influence security lack the economic incentive to do so. Conversely, those tasked to deal with Internet security, such as traditional law enforcement, may lack the required technical knowledge (although they are catching up quickly).

In addition, those in charge of protecting a system may not bear consequences if it fails [14, 15]. This is a classic example of negative externalities: the cost of a security failure by the owner of a machine or service ends up with third parties [16]. For instance, an individual who connects an insecure PC to the Internet does not face the full economic cost of that action; similar to an individual who produces air pollution by driving a diesel car [14].

There is also information asymmetry in the market for cybersecurity. This leads to a situation like what Akerlof called the 'market for lemons'. That is, buyers of a second-hand car cannot distinguish between a high-quality car, termed 'a peach', and a low-quality car, termed 'a lemon', whilst sellers do know the difference [17]. Buyers are therefore only willing to pay a fixed price (a median price between the 'lemon' and 'peach'), and sellers only sell when they have a 'lemon'. Otherwise they leave the market, which eventually reduces the overall willingness-to-pay of buyers [17]. Likewise in the market for security, buyers are unwilling to pay a premium for more secure services, so sellers are unwilling to offer them [18, 14]. In such a market, a major governance challenge is to improve the incentive of key actors (sellers) to invest in cybersecurity.

A number of properties set hosting providers apart from other intermediaries. (i) Hosting providers are spread over more than 150 jurisdictions. Most of these jurisdictions have few or no formal regulations in place imposing security requirements or obligations on this market [9, 19, 20]. (ii) Renting hosting services is not geographically bound to the country where the infrastructure is located. Such services can be rented anywhere in the world. Technically, many of these services are highly substitutable as well. Thus, hosting services can easily move their infrastructure from one country to another. The fact that hosting companies are so 'footloose' makes the security challenge more onerous. Stimulating improvements in security can therefore be more complex for hosting services than for other Internet intermediaries, such as broadband ISPs, which are geographically bound to physical networks. (iii) Multiple actors are

involved in the hosting space, with both providers and customers occupying various 'layers'. In addition to offering hosting services directly to end-users, some hosting providers lease hosting services to other retailers (resellers), which then sell them on to their own customers [9]. (iv) Security provision responsibilities are rather unclear in hosting services. Both providers and customers have agency. Authority and responsibility for security thus shifts between hosting providers and customers, depending on the hosting service type on offer.

Beyond these known properties of hosting providers, there are many areas in which we are still in the dark. Given what is known, it is clear that understanding and improving the security of the thousands of providers, across the multitude of jurisdictions, is a complex undertaking requiring action from actors beyond the hosting providers alone.

In addition, the negative impacts of insecurity in the hosting services market affect not only providers, but also users, the economy, and society as a whole. Security in the hosting market therefore constitutes a *collective action* problem. In other words, multiple actors would benefit from a solution to this problem, though it is implausible that any individual actor could provide a solution alone, due to all the associated properties and costs.

1.1.4 Security as a governance challenge

The literature on traditional governance identifies four canonical modes, or ways of steering and collaboration, through which complex problems can be addressed [21]: market governance, hierarchical governance, network governance, and community governance [22, 23]. Could any of these offer effective ways of improving hosting provider security in the face of the current underperformance of the hosting market itself?

Market governance hinges on efficiency in resource use and competition between enterprises [24]. Judging from security outcomes (abuse incidents), security levels are currently rather low in the hosting market. This can be termed as a market failure. That is, the market has failed to supply sufficient security with the hosting services offered. Providers lack incentives when it comes to security provision, despite their critical position in cyberspace. After all, security measures are costly. Moreover, information about security is asymmetric. Customers are less savvy than providers about the security levels particular hosting services offer. This reduces many providers' willingness to implement security measures to safeguard their networks. Additionally, the information asymmetry present in this market makes it difficult even for providers to reliably assess the effect of their security policies in comparison to their competitors [25]. Finally,

negative externalities are at work, as insecurities on one website may compromise all of the websites hosted on a shared server.

Hierarchical governance is found in traditional top-down rule-making, for example, through laws, legislation, and regulations [21]. If a market fails to function satisfactorily, a government could address the problem by resorting to hierarchal means, such as law enforcement and regulations. However, given the market conditions outlined above, we suspect hierarchical governance to have limited effectiveness in improving hosting service security. For one thing, this market is globally distributed over more than 150 jurisdictions, making governance via regulations considerably more challenging. Many of these jurisdictions barely have any law enforcement in place, let alone a regulatory framework capable of mandating cybersecurity standards. Furthermore, the few government measures implemented up to now have been predominantly reactive. These have mostly comprised countermeasures initiated subsequent to the detection of malicious activity, such as notifications and take-down efforts. Besides, the scale of government efforts has been miniscule in comparison to the scale of cybercriminal activity. As such, the number of cases in which national or international government entities have taken action is dwarfed by the number of incidents.

Network governance is characterized by interdependence and continuous interaction among network members. These interactions reflect shared resources or purposes and are based on mutual trust [24, 21]. Peer pressure is an example of a network governance mechanism. For instance, if hosting providers were pressured by their peers to ensure a certain level of security for their services, overall security levels would be significantly improved. However, such peer pressure is hardly viable beyond a small-scale operation. It would be especially difficult to achieve in a globally distributed market such as that for hosting services. Moreover, most network governance mechanisms are predicated on trust and reciprocal relations. Developing these among the thousands of hosting providers scattered around the globe would be a challenging task indeed.

Community governance is based on communal identity and norms [21]. It is characterized by large groups of actors aiming to overcome a collective action problem. Examples of communities related to the hosting market are the Messaging, Malware and Mobile Anti-Abuse Working Group (M3AAWG) [9] and Stopbadware [19]. However, their output so far has been limited to formulation of best practices, which have not yet been very effective due to the lack of reliable insights regarding the complexity of the hosting market.

In sum, in order to improve security in the hosting market, there is a need for a governance mechanism or combination of governance mechanisms that can

be effective given the known properties of hosting providers and the market they operate in. However, there is still a lot about this market that is unclear. For instance, there exists no comprehensive empirical understanding of how many providers operate worldwide, how their services are distributed, and what hosting types predominate.

Therefore, the first step in tackling hosting providers' security problem is to focus on the hosting market itself. That is, a better understanding is needed of the hosting providers, the current security levels of their infrastructure, and the mechanisms and factors that shape their security decisions and security outcomes.

The current research takes this proposition as its starting point. The following section identifies knowledge gaps regarding hosting providers and their role in the provision of security in cyberspace. This is followed by a further specification of the scope of this research and introduction of the main research questions.

1.2 Research Gaps

Security of hosting providers is the subject of a growing literature. Numerous scientific studies and industry reports have drawn on abuse data to make inferences regarding security at different levels. Some scholars have investigated security and vulnerability at the level of individual network entities, such as domain names and Internet Protocol (IP) addresses [26, 27, 28, 29, 30, 31]. Others have investigated the security of networks [32, 33, 34], organizations [35] or hosting providers [10], where abuses have been located. Industry reports have ranked hosting providers based on the number of abuse incidents in their networks [36, 37].

This previous work provides a foundation for understanding the security of the infrastructure operated by hosting providers, such as websites and IP addresses. Their results allow us to *theorize* that hosting providers could play a role in tackling cybersecurity problems. However, we do not as yet know if and to what extent hosting providers can actually play a role in security provision in the hosting market. This is mainly because we lack key insights and data, beginning with the security problem itself and hosting providers' involvement in it.

First, we lack as yet a comprehensive mapping of the hosting market and the existing security measures, based on empirical data. Such a mapping would provide information about the different hosting services on offer and their char-

acteristics. How is the hosting market structured? What business models exist in the market? How many providers operate worldwide? How are these geographically distributed? What are the current levels of security in this market? What security measures are providers already taking

Likewise, we have no metrics, as yet, for actually measuring hosting providers' security levels. There is no single accepted definition of hosting provider security or method by which to gauge it. Abuse data or 'blacklists' are publicly available registers containing metadata on websites utilized in particular types of attacks, such as phishing or malware. These basically connect malicious activities to technical identifiers – typically IP addresses, domain names, or URLs. Some previous research has used the number of times a provider's name appears in such blacklists as a proxy for their security/insecurity. However, possible biases introduced by such data are unknown; nor do we know how many blacklist mentions actually materialize as cybersecurity incidents. Some blacklists are open to the public, allowing users to add entries they perceive as malicious. This may introduce errors. Research that relies on direct counts from such sources without attempting to reduce possible data biases (removing false positives) or addressing them (approximating the effect of biases) would thus produce unreliable results.

Furthermore, no research has as yet systematically scrutinized the hosting services market. What drives providers' security performance, as measured by abuse incidents? Is the performance of hosting providers more a function of certain inherent structural properties, or of reactive and proactive security efforts? What role do providers play in security provision? Is the security of the websites in a provider's networks influenced only by them, or by webmasters as well? Similarly, little has been done to develop empirical models for quantifying the impact of factors related to public regulation, self-regulation, market characteristics, and other forces on the security performance of hosting providers. For example, how can we quantify the effect of a country's regulatory framework on the security performance of providers in that country.

Finally, we do not know, as yet, what these knowledge gaps mean for governance. Here governance is defined as processes and structures for coordination, steering, and decision-making among the variety of actors involved in tackling the collective problem of providing security in the hosting market.

1.3 Research Aims and Questions

We already have certain information about hosting providers, about the market for hosting services, and about the security levels of the infrastructure they operate, such as websites and IP addresses. This knowledge allows us to theorize that hosting providers could play a role in in tackling cybersecurity problems. However, research on the hosting space and the role of providers in security provision requires deepening in two major directions: (i) improvement of the technical metrics used to measure cybersecurity performance and (ii) illumination of the relationship between cybersecurity incident data, the economics of the hosting provision market, and governance. Herein lies the focus of this dissertation. This research seeks to advance understanding of the structural properties of the hosting providers and the market they operate in, while investigating different methods of measuring the performance of hosting providers in security provision. It builds upon three bodies of research: web security, security economics and Internet governance. The aim is to answer the following research question:

How can the security performance of hosting providers be measured and improved?

This main research question is divided into several areas of inquiry, or sub-questions. These sub-questions are explored in subsequent chapters through five separate studies. The section below introduces these studies and their corresponding sub-questions in more detail.

1.3.1 Study 1: Understanding the basics of the hosting market (Chapter 3)

The first study is an empirical analysis of the hosting market. Various policies, standards, and best practices have emerged to improve hosting security (e.g., [9, 19]). All these, however, grapple with a significant barrier: the incredible complexity and heterogeneity of the hosting market.

Little effort has been put into reliably identifying the economic agents that operate the IP and domain space, such as hosting providers and the organizations behind hosting services. Additionally, we know little about the hosting market and the distribution of different hosting services. Our study is therefore the first to connect technical identifiers such as domain names and IP addresses from empirical data to hosting providers. The aim is to explore the hosting market and the different business models present in this market.

In short, the study aims to answer the following research question:

1. *What are hosting providers and how is the hosting market structured?*

1.3.2 Study 2: Measuring the impact of provider structural properties on abuse (Chapter 4)

The first study, above, establishes a methodology for identifying hosting providers as economic organizations responsible for the security of the IP addresses assigned to them. The next step is to infer and understand the factors that can influence providers' security performance, as indicated by abuse observations. Some previous work has been done on identifying culprit hosting providers or 'bad performers'. Much of it, however, has neglected the impact of influential factors when drawing conclusions from abuse observations.

Our second study addresses this limitation. First it identifies factors that can influence the abuse data generation process. It then goes on to quantify the impact of these factors. The focus is on the following research questions:

2. *How can we analytically disentangle the different factors at work in the data generation process of abuse observations regarding hosting providers?*

3. *What is the impact of providers' structural properties on their security levels, for the case of phishing abuse?*

We propose an analytical model identifying sources of variance in abuse observations, such as factors related to providers' structure and security efforts, attacker behavior, and measurement error. Next, the relative impact of the structural properties of hosting providers, as described by the analytical model, are estimated using quantitative statistical models.

1.3.3 Study 3: Measuring the impact of providers' reactive security efforts on abuse (Chapter 5)

The second study demonstrates that certain inherent structural properties of hosting providers, such as size and business model, explain more than 84% of the variance in phishing counts. We suspect, however, that the impact of these properties will differ for different types of abuse, assuming attackers are sensitive to providers' reactive security efforts. One example of reactive measures taken by providers is the 'uptime' of a malicious domain, determined by how quick providers take down malicious domains reported in their networks. We hypothesize that uptime of abused domains is critical to attackers, especially

in types of abuse where domain names provide the main node for distributing malware or sending commands to other nodes.

This third study tests this hypothesis. Specifically, it answers the following research questions:

4. *To what extent are abuse concentrations determined by the structural properties of providers, for the case study of infrastructure used in malware distribution?*

5. *What is the impact of providers' reactive security efforts? Do attackers prefer providers that take little or no abuse response action?*

Similar to the second study, we use quantitative statistical models to estimate the impact of different independent variables on the count of domains used in malware distribution, as the dependent variable.

1.3.4 Study 4: Understanding attacker behavior (Chapter 6)

The second study shows that in addition to provider properties and security efforts, attacker behavior and preferences impact abuse concentrations. Accordingly, the fourth study focuses on attacker behavior. Via an exploratory analysis, we study attackers' preferences in target selection for financial malware attacks. In addition, the impacts of hosting provider take-down efforts are assessed on attackers' C&C infrastructure.

This fourth study answers the following research question:

6. *What factors influence attackers' preferences in target selection for malware abuse?*

The study draws on Zeus family malware data over a time span of four years. We trace attackers' choices and activity patterns using techniques borrowed from statistics and machine learning.

1.3.5 Study 5: Measuring the impact of providers' proactive security efforts on abuse (Chapter 7)

The analytical model presented in the second study indicates that abuse observations are determined by attacker behavior, the structural properties of defenders, the security efforts of defenders, and measurement error. That same study shows that the structural properties of hosting providers can explain more than 84% of the variance in abuse observations.

The focus of our fifth study is providers' proactive security efforts. Specifically, we assess the impact of different proactive security measures taken by webmasters and hosting providers on abuse in the shared hosting environment.

The following research questions are addressed:

7. *To what extent and in what areas can hosting providers influence the security of websites?*
8. *How do the proactive security efforts of hosting providers influence the prevalence of abuse?*

To estimate the security effort made by hosting providers, this study draws on a diverse set of security and software features collected using a series of measurements. It then distinguishes features that collectively contribute to what providers can influence in terms of security, such as infrastructure security and web application security – as opposed to the group of features that are mostly determined by webmasters, such as security measures for website content. We construct multiple statistical models to estimate the impact of each factor on malware and phishing abuse observations.

1.4 Dissertation Outline

The remainder of this dissertation is organized as follows. Chapter 2 reviews the literature related to the security practices of hosting providers, as the overarching context of this research. Chapters 3 through 7 then present the five studies introduced above. Finally, chapter 8 recaps and summarizes the studies and presents proposals for future research.

Each of the five empirical chapters has been published as a separate peer-reviewed article in a highly ranked outlet with acceptance rates of 25% or lower. Table 1.1 provides an overview of the corresponding scientific articles. I was fortunate to be able to conduct these studies in collaboration with great researchers in the field of cybersecurity, as is reflected in the list of co-authors in Table 1.1. I gratefully acknowledge their contributions in Section 8.3.3, located at the end of this dissertation.

Table 1.1: Overview of dissertation chapters

Chapter	Publication
3	S. Tajalizadehkhoob, M. Korczynski, A. Noroozian, C. Ganan, and M. van Eeten, "Apples, oranges and hosting providers: Heterogeneity and security in the hosting market". In Proceedings of the *IEEE/IFIP Network Operations and* Management *Symposium (NOMS)*, IEEE, 2016.
4	S. Tajalizadehkhoob, R. Bohme, C. Ganan, M. Korczynski , and M. van Eeten, "Rotten Apples or Bad Harvest? What We Are Measuring When We Are Measuring Abuse". https://arxiv.org/abs/1702.01624, Forthcoming in *ACM Transactions on Internet Technology (TOIT)*, ACM, 2017.
5	S. Tajalizadehkhoob, C. Ganan, A. Noroozian, and M. van Eeten, "The Role of Hosting Providers in Fighting Command and Control Infrastructure of Financial Malware". In Proceedings of the *12th ACM ASIA Conference on Computer and Communications Security ACM (ASIACCS)*, ACM 2017.
6	S. Tajalizadehkhoob, H. Asghari, C. Ganan, and M. van Eeten, "Why Them? Extracting intelligence about target selection from Zeus financial malware". In *Workshop on the Economics of Information Security (WEIS)*, 2014.
7	S. Tajalizadehkhoob, T. van Goethem, M. Korczyński, A. Noroozian, R. Böhme, T. Moore, W. Joosen, and M. van Eeten, "Herding Vulnerable Cats: Disentangling Joint Responsibility for Web Security in Shared Hosting" . In Proceedings of the *ACM Conference on Computer and Communications Security (CCS)*, ACM, 2017.

CHAPTER 2

Review of Literature

This chapter presents the scientific state of the art regarding measuring and explaining abuse incidents in hosting provider networks. It reviews studies and concepts from work on web security and security economics. The chapter includes a model (Figure 2.1) that illustrates the conceptual relationships between factors that shape the security performance of hosting providers and attacks. The work of this dissertation is focused primarily on the highlighted parts of the conceptual framework. This chapter first introduces the framework. Next, it reviews and summarizes prior work regarding different parts of the conceptual framework.

2.1 Conceptual Relations

Rehashing the driving forces behind concentrations of abuse in the network of hosting providers requires a deep understanding of the underlying factors at work. Figure 2.1[1], adopted from the earlier work [38], depicts the conceptual relation between such factors.

Abuse incidents cause tangible losses (e.g., money and resources) and intangible losses (e.g., reputation and credibility). Such losses do not only affect hosting providers and their individual customers, but also impact hosting as a sector and society at large. Incidents are principally caused by cyber attacks. Security/vulnerability and exposure act as moderating factors. They do not cause attacks, but influence the degree to which the attacks materialize as incidents. Exposure refers to an array of factors that affect the magnitude with which a providers' infrastructure is exposed to potential attacks. For example,

[1]I gratefully acknowledge the contributions of Rainer Böhme, who had the original idea for the model, and of the participants of the Dagstuhl Seminar 16461 "Assessing ICT Security Risks in Socio-Technical Systems" who helped to further articulate it.

providers with more customers have higher exposure rate than those with fewer.

Security and vulnerability capture the extent to which a resource is protected. This, in turn, is influenced by controls. Controls consist of measures taken by actors to protect a resource. This actor could be the hosting provider herself or the customer, depending on the hosting type. In other words, controls are the efforts put in place by a responsible entity for securing the resource(s).

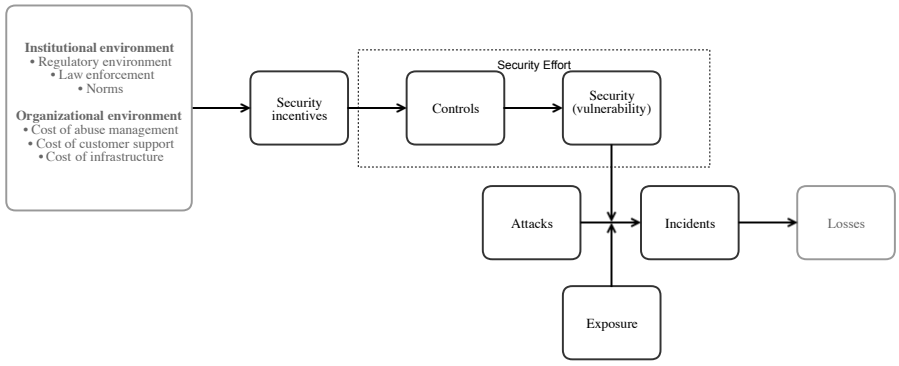


Figure 2.1: Conceptual framework

Decisions regarding which control measures to take are ultimately influenced by security incentives of those in charge of security. These incentives are themselves shaped by a diverse set of factors. The institutional environment of providers is composed of, among other things, social norms, law enforcement and regulatory framework in different jurisdictions and geographical locations. Factors related to the organizational environment, such as cost of abuse management and customer support, can also be influential for security decisions.

In the upcoming sections of this chapter, we review the literature around each of these factors in more detail.

2.2 Hosting Types

According to the Messaging, Malware and Mobile Anti-Abuse Working Group (M3AAWG) best practices for hosting, hosting providers are entities that 'offer end users the ability to create their web presence on hardware they do not actually own' [9].

Hosting services are offered in peculiar types. Depending on the service type, the responsibility in provision of security, abuse handling, and administrative rights can shift from the hosting provider to the customer. So does the distribution of abuse over the providers' network. Prior work has focused on security in different forms of hosting services such as shared and cloud hosting, among others [39, 40]. Here, before discussing the literature on abuse concentration in networks of hosting providers, we first break down the different forms of hosting services and the distribution of responsibilities among providers and customers in maintaining security.

Hosting types can vary from colocation where the provider rents out a physical space for hosting servers owned by customers, to managed hosting which includes a comprehensive support by the hosting provider. Hosting types with higher storage capacity can be used not only for hosting websites (web hosting) but also for data storage and processing purposes. Below, is a short description of distinct hosting types:

- **Shared**: This category of hosting services includes the use of shared resources. In *shared hosting* service, the server and the majority of administrative rights are owned by the hosting provider. This is while the customer has only an end-user access [9]. That is, the customers have control over the client-side applications, in most cases. In shared hosting a physical server and its resources such as RAM and CPU is shared among multiple domains. This is considered entry level hosting, as it requires minimum technical knowledge from its users. It is considered the most affordable hosting plan as well. In *Virtual Private Server (VPS)* a server is divided into a few virtual servers (compartments), where each unit has its own server software set up separately and is capable of functioning independently [41]. Although in VPS the physical server is still shared among multiple customers (less than shared hosting), each website/customer still receives a dedicated portion of those resources such as CPU. *Cloud hosting* is built with the same logic as shared hosting, but with redundancy. Hence, it brings in better performance and uptime, in comparison to traditional shared hosting [9].

- **Dedicated**: In this category of hosting services each website is hosted on a dedicated server and a dedicated IP address. The server is owned by the hosting provider. The customer controls and maintains the server, OS, and software. The customer has full root administrative access to the server and hence is responsible for its security. A milder and more expensive version of a dedicated hosting is *managed* hosting. Although

the user is granted administrative access to the server, the provider is still responsible for server's technical and security support [9].

- **Reseller**: This is where a hosting company provides a dedicated or unmanaged service to a customer (reseller), who can then act as an independent hosting provider and sell services, typically shared hosting, to other customers. These customers can then be either end-users or other resellers [9]. The multi-layer structure of reseller hosting can potentially add more complexity and delay in abuse handling.

- **Colocation**: In colocation only the physical space for hosting the server is provided. Everything else, including the server itself and its maintenance is realized by the customer herself.

Depending on the distribution of authority and responsibility between providers and costumers in each service type explained above, the providers can play a crucial role in the provision of security for their customers. For example, in shared hosting, providers have the most control over the server-side software and hence most responsibility for their server-side resources. However, in dedicated hosting, customers typically have full administrative rights over the dedicated box and hence have to assure its security. Therefore, any research that aims at evaluating the security performance of providers requires to account for the types of hosting service and to identify the entity that is bearing the responsibility and authority for provision of security.

However, given the heterogeneity and several layers of complexity in services such as reseller hosting, establishing the economic entity who is in control of the security and therefore should be held responsible has not yet been very straightforward. Although over the years, various standards and best practices have come forth to improve hosting security [9, 19], none revealed any information about the most basic concepts of the hosting market: How many providers are there? What address space do they manage? How are they distributed in terms of geography, size, types of services?

There exists no comprehensive list of all hosting providers. Prior work on the hosting market often uses BGP data to map IP addresses of abuse incidents to the organizations that own Autonomous Systems (ASes) and equated the latter with hosting providers. They are network administrative entities that control IP routing throughout the Internet [34]. This is problematic due to two main reasons: First, the entity that is routing an IP address is not always the same as the organization that is hosting an IP address. While some organizations operate under several ASes, other organizations share a single AS [42]. Second,

ASes are technical identifiers and cannot be held responsible for security of websites. Hosting providers are organizations who operate the IP space and thus are economic real-world entities in charge of security of their services.

Most of the prior work on predicting or analyzing abuse patterns have been focused on different levels of analysis other than hosting providers: (i) individual resources such and domain names and IP addresses [43, 44, 45, 46, 47, 30], (ii) aggregated network resources such as ASes [48, 34] and TLDs [49], (iii) economic actors such as registries, registrars [50, 51], and organizations [35]. Aside from the work carried out by Canali et al. on the security practices of a small sample of shared hosting providers [10], we are not aware of any work that analyzes abuse patterns across the population of *hosting providers*. Even the basics of the market are not studied yet in any scientific work. This is the first part of the knowledge gap that this dissertation intends to fill in. In chapter 3 of this dissertation, we conduct a comprehensive study on the hosting market and describe its properties based on the empirical data.

2.3 Attacks

The cybercrime problem starts with attacks executed by cybercriminals. Cyber attacks can vary diversely based on the aim and business models of cybercriminals. We do not perform any specific study concerning attack trends during the course of this dissertation. However, reviewing attack trends are still relevant for gaining a better understanding about the measures that hosting providers and webmasters could take to avoid incidents. In the remainder of this section, we review the body of literature around a few examples of attacks carried out by utilizing the hosting infrastructure.

A large and growing body of literature has investigated attack trends using different botnets that utilize hosting infrastructure to host the command and control servers. Andriesse et al. have carried out a detailed analysis on the Gameover Zeus, the peer-to-peer (P2P) Zeus malware variant, and demonstrated its high resilience [52]. Rossow et al. have studied the properties and vulnerabilities of eleven active P2P botnets and assessed their resilience against attacks and showed that some P2P botnet families contain over a million bots [53]. Wang et al. studied GR, an influential Black Hat search engine optimization (SEO) botnet and found several characteristics such as modest size and low churn different from typical e-mail spam botnets [54].

A considerable amount of literature has been published on DDoS attacks [55, 56, 57]. Rossow performed a detailed analysis on distributed reflective denial-of-

service (DRDoS) attacks where attacker sends requests to public servers such as open recursive DNS resolvers and spoof the IP address of a victim. Having used darknet as well as network traffic from large ISPs, he observed both victims and amplifiers and concluded that attackers are already abusing vulnerable protocols other than DNS [58]. Kührer et al. monitored different sources of amplification DDoS attacks. Their results showed that vulnerabilities in the TCP handshake can help attackers to abuse millions of hosts to achieve 20x amplification. They also highlighted networks that allow IP address spoofing as the root cause of amplification attacks. Such networks often lack egress filtering [59]. Santana et al. studied the infrastructure of booter services, services that facilitate DDoS attacks via the provision of of infrastructure-as-a-service to perform attacks. Their results revealed that among the 11 booters analyzed, 10 of them had their infrastructures based on Web-shells scripts and only 1 based on servers [60].

A great deal of previous research into Phishing attacks, where attackers directing users to fraudulent websites which are either hosted on a compromised server or is maliciously registered [61, 62, 63]. Mavrommatis et al. studied popularity of drive-by downloads within a 10 month period and show that over 3 million malicious URLs initiate drive-by downloads and approximately 1.3% of the incoming search queries to Google's search engine returned at least one malicious URL in the results page [64]. Leontiadis et al. investigated search-redirection attacks and found that about one third of all search results are over 7000 compromised websites that redirect the users to a few hundred pharmacy websites [65]. Alrwais et al. looked into *watering holes*, another emerging malware distribution attack where the target of compromise is strategically chosen with the goal of collecting information from a specific group within an organization [66].

Some studies focused on identifying possibilities for attack vectors using hosting infrastructure based on vulnerabilities present in the hosting websites and webservers. A few studies have identified new attack vectors, on the basis of vulnerabilities present in SSLv2 [67] and TLS [68, 69] . They reported that such vulnerabilities are a significant threat against SSL ecosystem. Finally, Nikiforakis et al. identified several attack vectors that can be carried out by exploiting vulnerabilities in configuration of JavaScript code inclusions [70].

Another body of literature worked on the detection of web-based malware campaigns [71, 64, 72, 73, 74]. Borgolte et al. developed the delta-system, a system that is able to identify previously known and unknown malware infection campaigns from changes associated with malicious and benign behavior in websites [74].

2.4 Abuse Incidents

Successful attacks can materialize as abuse indents. Abuse incidents can be measured and collected via methods such as honeypots, spamtraps, and crawlers. Abuse incident observations are then made available mainly for protection and cleaning purposes, in different forms: Blacklists/blocklists are the first example. Blacklists are lists that contain meta data regarding websites or webservers used in various types of online attacks such as DDoS, phishing, malware propagation, and child pornography, among others. Examples of blasklists/blocklists are Anti Phishing Working Group (APWG) [75], PhishTank [76], abuse.ch, The Swiss Security Blog [77], and MalwareDomains [78]. Such lists are normally maintained by third-part security companies or institutions. In some cases they are open for the public to report incidents (e.g., DShield [79]). Prior work has studied limitations of abuse blacklists such as comprehensiveness and independence at length [80, 81, 82]. Having that said, any study that utilizes such data still requires to evaluate the robustness of their results against the potential biases in the blacklist data.

Anti-virus companies such as Sucuri, McAfee, and Norton integrate observations data as an input into their products with the goal of improving the protection and detection quality. Google safe browsing utilizes incident observations as well. It offers a browser plug-in to help end-users in detection of malicious content. Incident report is another mean via which abuse incidents are made available. Incidents reports can also be private or publicly available. Examples are VERIS Community Database (VCDB) [83] and Data Breach Investigations Reports (DBIR) [84].

There is a large number of published papers [63, 6, 85, 48, 43, 47, 86, 87, 88] that studied attack concentrations and patterns in attack targets using abuse blacklist data. Some of these studies are carried out at the level of individual technical entities such as domain names/IPs [63, 89]. Others investigated concentrations for technical identifiers of network entities [85, 6, 48, 80] or real world economic entities/organizations who operate the networks [50, 51, 90, 35].

Among these are also studies that focused on the relationship between control measures or vulnerabilities and abuse. Vasek et al. studied the odds of domain names getting compromised via phishing or malware attacks when they have certain CMS installed or when they are hosted on a shared server [44]. Zhang et al. [90] and Liu et al. [35] looked into the relation between a number of mismanagement security symptoms in the networks of organizations and abuse incidents from incident reports. Although it is very important to anticipate

these relationships, any further remedy for improving security in those networks requires knowing the control points or the key actors who can in practice influence the security.

To that end, there is hardly any paper that investigates the relationship between vulnerabilities/control measures and the amount of abuse in the network of hosting providers. In addition, there exists no study so far that has focused on empirically identifying areas of control, where each of webmaster or provider can influence security of websites. This is one of the most important areas where this thesis is aiming to further investigate.

2.5 Security and Vulnerability

Attacks can be influenced by certain vulnerabilities in the networks of providers, some of which are already known and others which are unknown, zero-day vulnerabilities. In recent years, there has been an increasing amount of literature by both industry and academia on measuring security of websites and web servers or detecting specific vulnerabilities that can lead to compromise. In this section, we discuss some example studies with the aim of providing an insight into this branch of work and its relation to our work.

The literature on this topic has revealed the emergence of several vulnerabilities. Industry has mainly been active in publishing reports on website security statistics [91, 92, 93, 94] ranging from general demographics to specific vulnerabilities. Alarifi et al. evaluated the security of popular Arabic websites via using known website scanners namely, Sucuri SiteCheck, McAfee SiteAdvisor, Google Safe Browsing, Norton, and AVG website. They observed that the majority of the scanned websites contain malicious contents which were proportional to website vulnerabilities , as unpatched software increases the risk of being vulnerable to compromise [95].

Other studies looked into more specific vulnerabilities. Kals et al. [96] and Lekies et al. [97] focused on automatic discovery of vulnerable websites and found several instances with exploitable SQL injection and Cross-Site Scripting (XSS) vulnerabilities. Nikiforakis et al. carried out a large-scale analysis of remote JavaScript inclusions in websites. They also propose a Quality of Maintenance metric that captures the security of web applications running on websites with remotely included the Java-Script library. Their QoM metric asses website's security in terms of availability, cookies, anti-XSS and anti-clickjacking, cache control, SSL/TLS implementation, and outdated web servers. This metric is used to study the trust relationships between websites and JavaScript inclu-

sions. Using the metric they found that a substantial number of high-profile websites that include JavaScript code from external sources are vulnerable to compromise [70]. Van Acker et al. examined login-page security of several websites and found many login pages vulnerable to password leakage and eavesdropping attacks. Nevertheless, they observed a few login pages with advanced security measures regarding the aforementioned vulnerabilities [98]. Doupe et al. develops a state-aware black-box scanner in which they evaluate vulnerabilities in a number of applications including WordPress CMS and PHP [99]. Vasek et al. carried out a case-control study where she measures presence of outdated CMS and web server software, among others [44]. Two studies investigated security of shared hosting servers, where a server is shared between different websites. Both studies demonstrated that lack of enforced session isolation leaves shared web hosts vulnerable to compromise [31, 39].

All the above discussed studies have been successful at revealing one or more vulnerabilities present on web applications, websites, or servers. None of these studies however empirically quantified how and to what extent such vulnerabilities lead to abuse incidents, in the networks of hosting providers. Such analysis is crucial in understanding why certain abuse incidents occur, where are they located, who can influence them and how can they be further mitigated. This is another area where the focus of this dissertation is placed.

2.6 Exposure

Vulnerabilities are one of the factors than can cause an attack to materialize as an incident. *Exposure* is the other factor that can influence occurrence of cybersecurity incidents. Traditional crime Routine Activity Theory highlights exposure as one of the five factors that determine the likelihood for an individual in becoming victim of a crime [100]. Exposure is how accessible potential targets are to potential attackers. The more exposed a target is, the higher the chance of a crime being materialized. The same holds in cyberspace Ḟor example, the more websites hosted in a network of a provider, the higher the exposure rate of that provider, and the chance of being compromised. Given that, any study that aims at identifying underlying reasons behind concentrations of abuse across networks of providers needs to take the effect of exposure into account.

Among the studies that looked into the abuse concentrations in networks of ASes or organizations, some utilized size of a network to normalize abuse counts [34, 33, 6, 101, 51, 102]. Others studied alternative factors such as domain age and Alexa popularity [103, 104, 26].

A very crucial part of reliably identifying culprit hosting providers is to understand what factors drive the concentration of abuse in their networks. The mere act of counting abuse data points in blacklist data and aggregating them to providers does not yield to a reliable benchmark due to the effect of exposure: the inherent or structural properties of a provider that can increase the probability of attracting more attacks regardless of the effort a provider puts in security. In this dissertation, we aim to map the factors at work in the data generation process of abuse observation and estimate their impact on abuse concentrations of hosting providers.

2.7 Controls

Controls are measures taken by responsible actors to ensure and improve the security of a resource and increase its protection against possible attacks. These measures are the results of security efforts put in place by several actors. Website administrators (from now on we call them *webmasters*), software vendors, and hosting providers are some of the most important examples of such actors.

We already know that most of the malicious content on web is not hosted on servers owned by attackers [105] but is rather either (i) malicious content hosted on compromised servers that are exploited due to particular vulnerabilities found by attackers via automatic scans [106], (ii) embedded malicious code in a third-party web application or (iii) maliciously registered domain names used for a purpose of attack (free or paid registration). Given these scenarios, it is very important that hosting providers and webmasters undertake the required control measures to maintain a desirable level of security for their webservers and websites.

Regarding specific control measures, Weichselbaum et al. studied the adoption of Content Security Policy (CSP) – a web platform mechanism that is designed to mitigate XSS attacks – and discovered that more than 90% of websites use a policy with significant flaws in CSP deployment which makes it bypassable by attackers [107]. Pen et al. proposed CSPAutoGen, an application that facilitates CSP adoption by enabling CSP option in real-time without server modifications [108].

Van Goethem et al. developed a general web scanner with the purpose of discovering various features that are *indicators* of website's "security consciousness". These are often certain controls that webmasters or hosting providers have/have not taken. Their findings suggest that many of the investigated websites contain vulnerabilities and weaknesses while most of the control measures

are sparsely distributed. In addition, websites' popularity did not show any relation to the presence of weaknesses and vulnerabilities, despite the common assumption that popular websites are more likely to have better security measures in place [26]. Zhang et al. and Liu et al. analyzed network mismanagement symptoms such as open recursive DNS resolvers, untrusted HTTPS certificates, lack of Egress Filtering – both at the level of IP addresses and autonomous systems (ASes) – and observed prevalent failures in implementing common security practices [90, 35]. Similar to Van Goethem et al., they argued that although most of the symptoms are not directly vulnerabilities, their presence might (ii) expose more attack vectors, or (i) indicate security unconsciousness of the network administrators. Finally, their work highlights a need for future work on measurement of additional security indicators as latent variables that are not directly causing compromise, at the level of hosting providers. To improve the existing defensive mechanisms of networks, they also recommend to put more focus on defender's properties with such symptoms rather than attacker's strategies.

In this dissertation, we aim to understand and ultimately improve the hosting security, by investigating both attacker's and defender's properties. Rather than focusing on specific best practices or vulnerabilities, we are interested to understand what type of providers' characteristics and attacker's preferences influence the amount of abuse in their networks. Control measures are among many other such characteristics. Inherent properties or exposure variables such as network size, geographical distributions, law enforcement, and business models of providers can be equally important in shaping the security outcomes.

2.8 Security Incentives

Control measures are often put in place if people who are responsible for provision of security have enough incentives to invest in it [109]. Internet security is the outcome of decisions of several autonomous actors in different markets all around the world [13]. Information insecurity has as much to do with misaligned incentives of the key actors involved as it has to do with technical vulnerabilities [13]. Incentives of hosting providers to invest in security are influenced by 'information asymmetry' in the market. That is, the buyers of hosting services (ordinary customers), cannot distinguish a more secure hosting service from a less secure one [17]. Even the regulators and hosting providers themselves do not have a clear idea about their position in the market in terms of security.

In addition, security measures are costly and user-unfriendly. Hosting providers

would only be willing to take security measures if they have the proper incentives for it. Such incentives are not only influenced by costs of security measures and information asymmetry in the market. Factors regarding institutional environment (e.g., market structure, regulatory environment, law enforcement) and organizational environment (e.g., cost of abuse management, cost of customer support) among others, influence them as well. A clear example is the difference in security levels due to difference in regulatory environment within distinct countries.

There is a relatively small body of literature that is concerned with the institutional or organizational factors influencing incentives and hence security outcomes. Subrahmanian et al. study the factors that can explain geographical variation in malware concentrations. Their results suggest high malware concentrations and malware binary downloads in countries with low GDP per capita [110]. Contray to Subrahmanian, Mezzour et al. found a relationship between countries' wealth (GDP), technological sophistication (ICT development) and attack concentrations (exposure) [111]. They also found a relation between the countries where attacks are hosted and the joined effect of widespread corruption and computing resources.

Garg et al. found out that the bulk of spam is hosted within a small number of countries. Their results also indicate a positive significant correlation between Internet penetration and spam concentrations [112]. In another study [113], Garg et al. examine the relationship between participation in e-crime tasks, such as Captcha solving in the Mechanical Turk crowd-sourced market, and countries' socio-economic characteristics. Their results indicate that low participation in e-crime tasks is significantly correlated with better rule of law, more governmental transparency, and less corruption [113].

The knowledge gap regarding factors influencing security incentives lies more at the level of analysis issue. Most of the existing studies looked into organizational or institutional factors that influence security of *individual* resources such as domain names and IP addresses, found in abuse data. However, little has been done on identifying such factors at the level of *hosting providers*. Having such factors identified is a step towards gaining a better understanding of hosting providers' security incentives and ultimately improving them.

2.9 Conclusions

In this chapter, using the conceptual framework introduced in section 2.1, we introduced the conceptual relations between the underlying factors that influence

providers' abuse concentrations. By surveying the literature regarding each of these factors, we highlighted the ares of knowledge gap, where this thesis is planning to contribute. We started with the formal definition of hosting providers and outlined different hosting types. We have seen that depending on hosting types, provider's responsibility and authority for the provision of security differs.

We further explained that abuse incidents are caused by attacks and this relation is mediated by providers' exposure and security characteristics. Security itself is then influenced by controls. These control measures are only in place if people who safeguard the systems have enough security incentives, and incentives themselves can be influence by several institutional and organizational factors such as regulatory framework, law enforcement and various costs.

In summary, to better understand and improve the role of hosting providers in security, the literature needs to be improved in three main aspects: (i) Understanding the basics of the hosting market, (ii) Quantifying the impact of various exposure, security or incentive related factors on the concentrations of abuse at the level of *hosting providers*, and (iii) Identifying the role of provider and distinguishing it from the role of webmasters, in the provision of security. In short, to identify areas for improving the role of providers in security, we suggest to start from the problem again, and the context in which this problem occurs.

CHAPTER 3

Understanding the Basics of the Hosting Market

All kinds of basic facts about the hosting market are still poorly understood. Questions such as how to identify a hosting provider, how many providers operate in this global market, what type of hosting services they offer and how they are distributed are still unanswered. This chapter providers a comprehensive overview of hosting providers and the market they operate in. The first part of the chapter discusses a method to identify hosting providers at scale from technical identifiers such as domain names and IP addresses, captured from passive DNS and WHOIS data. It then uses these to explore the basics of the market and of providers' business models. The second part of the chapter then tests whether the different business models are correlated with differences in abuse rates across the market.

3.1 Introduction

Hosting providers play a pivotal role in the provisioning of all kinds of Internet-based services, as well as in mitigating the abuse of these services. Criminals purchase or hack services for hosting malware, phishing pages, command and control (C&C) servers, drop zones, dark markets, child pornography and more.

Over the years, various policies, standards and practices have emerged to improve hosting security (e.g., [9, 19]). These initiatives run into a significant barrier: the incredible complexity and heterogeneity of the hosting market. Even the most basic facts are unknown: How many providers are there? What address space do they manage? How are they distributed in terms of geography, size, types of services?

Developing policies and best practices in the absence of this kind of informa-

tion seems unlikely to be effective. We cannot generate reliable security metrics for hosting providers without accounting for their heterogeneity [114]. It makes a big difference whether a best practice is geared towards hosting behemoths like GoDaddy, which operates an infrastructure across 800,000 IP addresses, towards the tiny providers which administer services on a single IP address, or perhaps towards some median point on this scale.

By necessity, security practices will look different across this spectrum. One can speculate that the same holds for security performance. Tiny providers might not be able to achieve the same level of competence as the large providers with their dedicated abuse departments, but perhaps they make up for it by being more agile.

Remarkably, the complexity of the hosting market has barely been studied empirically, least of all in the area of security. Research has typically equated providers with Autonomous Systems [34, 115, 32]. Using routing data to identify providers and attribute security incidents is problematic as a lot of address space that is announced by an AS is not actually assigned to, or administered by, the AS owner.

There are some proprietary approaches to more accurately map the hosting space [116], but the underlying methodology and data are not publicly available. Lists published by sites like `webhosting.info` are of poor quality and lack key properties needed for research. In short: a decent map of the landscape is missing.

In this chapter, we propose a novel measurement approach for capturing the complexity of the hosting market. In Section 3.2, we systematically identify hosting providers through a fine-grained method combining passive DNS data to find hosting infrastructure and WHOIS data to determine address space assignment around that infrastructure. This results in a set of 45,434 hosting providers. Section 3.3 discusses the hosting landscape by exploring different provider characteristics that can be extracted from the data. In Section 3.4, we condense the complexity and heterogeneity of the hosting market by performing cluster analysis on the properties of providers. Finally, we demonstrate the value of these clusters by showing that they are associated with significant differences in the uptimes of phishing sites.

As far as we know, this is the first comprehensive mapping of the hosting provider market. The value of a more accurate mapping of the hosting market consists of i) identification of providers rather than owners of Autonomous Systems; ii) more accurate attribution of security incidents to providers; iii) more accurate comparison and benchmarking of providers, also by normalizing for the size of providers. In the chapter, we demonstrate these contributions by using

the new map in a case study of phishing websites. We make the map available to other researchers upon request.

3.2 Methodology for Identifying Hosting Providers

The Message, Mobile and Malware Anti-Abuse Working Group (M3AAWG), a leading industry association, defines a hosting provider as "any entity which offers end users the ability to create their web presence on hardware they do not actually own" [9]. Hosting providers offer a variety of hosting services. They range from free and shared hosting services with limited resources and administrative privileges for customers, to more expensive services such as dedicated hosting and virtual private servers (VPS) where customers have more control over the computing resources [9]. The role of the provider to safeguard security also changes across these services.

While web presence is just one of the services on offer, we assume that all hosting providers have at least some webhosting in their portfolio. This allows us to use domain names as a way to identify providers. More specifically, we follow several steps to get from domain names to the population of providers (see Figure 3.1):

1. Extract domain names from DNSDB, a passive DNS dataset with an reasonable approximation of all domains in use on the web;
2. Identify the IP addresses where these websites are hosted;
3. Extract from WHOIS the netblocks to which these IP addresses belong and the organizations to which they are assigned;
4. Filter out the organizations that are clearly not hosting providers.

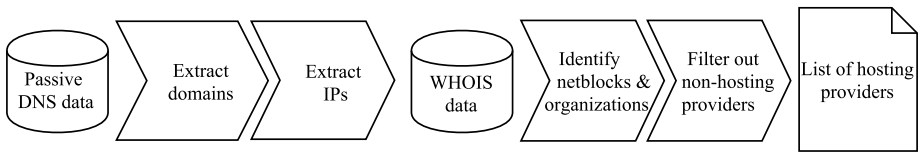


Figure 3.1: Steps towards identifying hosting providers

In the next subsection, we systematically walk the reader through the design decisions taken in each step of this process.

3.2.1 Identifying webhosting infrastructure

We first obtain a list that approximates the population of all domains in DNSDB – a passive DNS database that is generously shared with us by Farsight Security. DNSDB data is widely used by researchers worldwide and to our knowledge, it has the best coverage of the overall domain name space that is available to researchers. It draws on hundreds of sensors worldwide and on the authoritative DNS data that various top-level domain (TLD) zone operators publish [117]. Other studies also confirm that DNSDB has a relatively unbiased and comprehensive view of the overall domain and IP space [118, 38].

From DNSDB we extract all second-level domain names seen between January-June 2015 and the IP addresses that they resolved to. We identify 214,138,467 unique 2nd-level domain names that are mapped to 47,446,082 unique IP addresses.

3.2.2 Identifying organizations and IP ranges

We use WHOIS data provided by Regional Internet Registries (RIR) to map the IP addresses of domains to the netblocks and names of the organizations to which these addresses are assigned.

WHOIS data has its own limitations, most notably the fact that records can be stale, inaccurate and non-standardized [119]. That being said, compared to the routing (BGP) data that most security research uses to associate IP addresses with providers, IP assignment better captures who is responsible for an address range and the services offered there than AS-level routing information. An AS, think of a data center, can announce routes for many different providers using its infrastructure.

We have used MaxMind's Organization database [120], which collates the WHOIS data of RIRs. The organization is identified by MaxMind from different fields of WHOIS databases, such as "descr" or "role" or "organization", depending on the RIR's WHOIS format.

When mapping IP addresses to organization names, an organization might appear multiple times in slightly different versions: `Go Daddy Netherlands B.V.`, `GoDaddy.com, LLC` and `GoDaddy.com Singapore`. The different names may point to the same organization. Sometimes, however, the differences reflect the fact that there are separate entities, for example in different jurisdictions. There currently is no reliable process to distinguish these situations, which is why we chose to not merge organizations with similar names.

Mapping IP addresses to their ranges and organizations results in a list of

161,891 organizations, covering 28,489 unique ASNs. On average, an ASN has address space allocated to around 7 organizations. This underlines just how problematic the current practice is to equate ASes with providers.

3.2.3 Filtering out non-hosting providers

Clearly, not all of the organizations that host domains are hosting providers. When filtering out these cases, one has to balance potentially removing true positives versus keeping in false positives. Since our aim is to capture the complexity of the market, we do not want to lose true positives and apply three filters that conservatively remove false positives.

Filter 1: AS level. In a previous study [121], we have manually categorized 2000 ASes that contributed the most machines to botnet populations seen in sinkholes and spamtraps. Based on different data sources, we assigned ASes to one of the following types: **(*i*)** education, **(*ii*)** government, **(*iii*)** hosting, **(*iv*)** ISP-mobile, **(*v*)** ISP-other, **(*vi*)** ISP-broadband, and **(*vii*)** corporate networks such as banks, hospitals, etc. The first filter removes 6598 organizations (4% of the total set) that are located in the 332 ASes belonging to the categories education, government, and corporate networks.

Filter 2: Organization level. We generated a list of keywords for education, government and corporate networks. For example, the education category consist of the following list of keywords: universi, institut, college, school, akademi, academy, academi, research, teach, education, and science. We matched the keywords with organization names. In case of a match, we excluded the organization.

In this step we removed 39,369 organizations from the 155,293 that remained after the previous filter (25,4%), most of which matched an education keyword.

Filter 3: Number of domains. The third filter looks at the number of domains hosted by the organization. Organizations that host fewer domains than a certain threshold value are considered as "non-hosting". We hypothesize such organizations are not providing hosting services for others but instead they host their own websites.

To find the appropriate threshold, we took a sample of 163 organizations through a stratified sampling method to maintain the population's distribution in terms of the size of their address space, while keeping the sample size amenable to manual inspection. We manually assign "hosting" and "non-hosting" labels to the organizations by checking their names and visiting the corresponding websites, if they exist. The "hosting" label is assigned to all organizations that offer hosting service as a part of their business.

We then perform a sensitivity analysis on the threshold value for the number of domains to filter out non-hosting providers from the total set. For each threshold on the number of domains, we calculate the following parameters:

$$FP\ rate = \frac{FP}{FP + TN}, \qquad TP\ rate = \frac{TP}{TP + FN} \tag{3.1}$$

$$Accuracy = \frac{TP + TN}{FP + TN + TP + FN} \tag{3.2}$$

Where true positive (TP) is when an organization is correctly classified as "hosting", false positive (FP) is when an organization with "non-hosting" label is incorrectly classified as a hosting provider. Similarly, true negative (TN) is when an organization that is labeled as "non-hosting" is correctly classified as "non-hosting", whereas false negative (FN) is when an organization that has "hosting" label is incorrectly classified as "non-hosting".

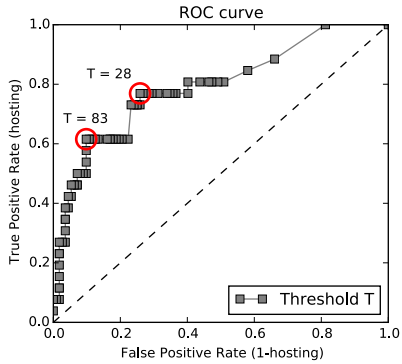


Figure 3.2: ROC curve of different threshold values for the number of domains

The receiver operating characteristic (ROC) curve shows the performance of different thresholds (Figure 3.2). The two thresholds marked with red circles in the ROC curve (T=83 and T=28) are the optimal thresholds for detecting hosting providers according to Equation 3.2. Note that our data is highly skewed and contains a large number of organizations with only a few domains. This leads to substantial noise when detecting hosting providers. At both thresholds, we have already included more than 99% of the total domain space in the data. Therefore the choice is essentially driven by the conservative approach of maximizing the chance of correctly identifying hosting providers and we are less sensitive to include false positives. We select T=28 as the threshold and define a hosting provider an organization that is hosting more than 28 domains.

The filter discards 73,801 organizations from the set of 119,235 providers (62%) – e.g., Family Dental of Chicago (netblock 72.54.46.208/29) and United States Institute of Peace (netblock 64.210.233.0/23).

After applying these filters, we have a population of 45,434 organizations identified as hosting providers.

3.3 Exploring the Hosting Landscape

From the underlying data, we can extract several characteristics of the 45,434 hosting providers, such as the size of their address space, as well as the portion of that space used for webhosting. What can these tell us about the hosting market?

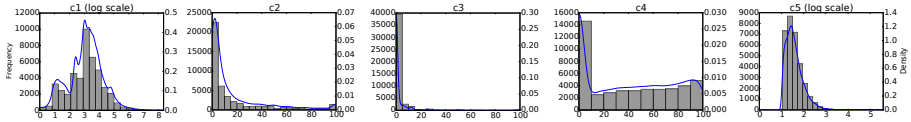


Figure 3.3: Histograms and kernel density estimates for five characteristics of hosting providers

[c1] IP address range size: The first plot in Figure 3.3 displays the distribution of providers in terms of their address space. The distribution goes from around 200 providers with only one IP addresses all the way up to providers with six or seven orders of magnitude larger address space. There we find ISPs like AT&T and Comcast, for whom hosting is not the main service. The distribution is centered around providers with 1,000 to 10,000 addresses (median: 1,517). From an economic perspective, this market shows a surprising lack of consolidation. One would think that economies of scale, in combination with commoditized services that can be globally delivered, would lead to a few large providers dominating the market. This mechanism is clearly visible in cloud services, but not here. It takes 1,210 providers to account for 80% of the address space used for webhosting. How can the many medium-sized providers compete on price with the large ones? How do the tiny providers survive in this market? This finding underlines that we know little about the incentives in this market and the security practices that they give rise to.

[c2] Percentage of IP range used for hosting websites: What percentage of the address space of a provider is used for webhosting? This tells us to what extent webhosting is the core business model or not. The second plot shows the distribution of providers. It shows that for the bulk of them, webhosting

is only a minor part of their infrastructure. Their infrastructure might be used to run game servers, databases, VPN exit nodes, and other services. Some smaller providers use almost all of their address space for webhosting, whereas larger companies such as GoDaddy and OVH are using approximately half of their allocated range for webhosting, but they are all on the higher end of the spectrum.

[c3] Percentage of IP range used for shared hosting: When talking about abuse in hosting services, shared hosting is often flagged as a problem area [26, 122]. One reason is the low profit margins of these services, which seems to be accompanied by poor security, according to a recent study [10]. The third plot shows the percentage of the address space used for shared hosting. We consider an IP address to be used for shared hosting if it serves more than 10 domains. While shared hosting draws a lot of attention in research, most providers actually use only about 10% of their address space for this purpose. Only around 500 providers use more than 50% for shared hosting, while 225 focus exclusively on shared hosting.

[c4] Percentage of domain names on shared hosting: A slightly different take on the importance of shared hosting is to look at its portion of all domains that are hosted by the provider. The fourth plot shows a rather uniform distribution, except for the first group, who offer no shared hosting at all. In other words, for webhosting as a service, shared hosting is provided in all portfolios and has a fluid proportion to other webhosting solutions, like VPS or dedicated hosting.

[c5] Density of domains on shared hosting IP addresses: The average number of domain names on IP addresses used for shared hosting can indicate in what part of the market the provider is competing. Higher density (more domains per server) would indicate more shared resources and competing for lower value customers. The last plot shows that a few hundreds of providers have shared IP addresses with more than thousand domains, on average, while the majority of the providers have 10 to 100 domains per shared IP address.

These individual characteristics give us a sense of the hosting landscape. We can see just how much complexity and heterogeneity is present across providers. There is remarkably little consolidation and many small players shape the landscape as much as the larger providers. Webhosting, the service that has dominated the image of the sector, only plays a limited role for many providers – and shared hosting even more so.

All of these characteristics influence security incentives and practices, especially in combination. Viewing a characteristic isolated from the others can be misleading. For example, when looking at the influence of size of a provider, one cannot simply use address space as a proxy, because it ignores the fact

that the providers with the largest address space are not predominantly hosting providers, so their hosting product groups may actually resemble those of small or medium-sized providers.

To deepen our understanding of the market, we would need to identify how different values for these characteristics occur in combination across the population of providers. We propose to profile the providers by performing cluster analysis on the characteristics. This would condense the complexity into a tractable starting point for further empirical research. Are certain types of providers more effective in securing their infrastructure? Perhaps type is not that relevant. We might find a equally strong and poor security practices within each type. In the remainder of this chapter we first perform cluster analysis on the characteristics and then use those clusters to determine whether they uncover meaningful differences in terms of security, as measured by the uptime of phishing websites in the networks of these providers.

3.4 Categorizing Hosting Providers

We try to profile hosting providers using the set of five characteristics explained in the previous section. To identify providers that have similar business model profiles, we need to use a machine learning technique that allow us to do a similarity analysis without a need for ground truth data. Clustering, is an unsupervised learning technique that would fit within this criterion.

In the following sections, we first identify the appropriate clustering algorithm, carry out the clustering and finally discuss the interpretation of clusters.

3.4.1 Choice of the clustering algorithm

To meaningfully partition the hosting space, we test four clustering algorithms: k-means [123], k-medoids [124], expectation maximization (EM) [125], and hierarchical [126]. We first randomly sample ten thousand hosting providers, we then evaluate the four selected algorithms using five types of cluster validation measures, as described by Brock *et al.* [127]. Table 3.1 reports on the stability metrics (APN: average proportion of non-overlap, ADM: average distance between means, and FM: figure of merit) and internal metrics (connectivity and silhouette width) calculated for four clustering algorithms and different numbers of clusters.

The results shown in Table 3.1 indicate that clustering of hosting providers obtained using hierarchical and k-means algorithms are more stable (smaller

Table 3.1: Stability and internal metrics per clustering algorithm and number of clusters

Clustering Algorithm	Metric	Number of Clusters												
		2	3	4	5	6	7	8	9	10	11	12	13	14
hierarchical	APN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ADM	10,628.70	17,804.84	17,805.16	17,805.16	21,989.46	21,991.56	21,993.01	21,993.08	26,749.91	26,750.06	26,751.00	26,751.00	29,675.66
	FOM	422,786.09	422,807.23	422,828.35	422,849.48	422,870.63	422,891.67	422,912.59	422,933.65	422,954.74	422,975.74	422,996.68	423,017.85	423,039.01
	Connectivity	3.86	9.54	11.54	14.31	18.87	20.87	23.37	27.10	29.42	31.42	35.47	40.32	42.92
	Silhouette	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.98
kmeans	APN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02
	ADM	10,624.97	18,525.27	18,527.03	21,989.46	25,463.27	28,099.38	28,100.34	28,289.60	28,290.13	29,999.74	30,000.07	30,085.53	30,112.69
	FOM	422,786.17	422,807.20	422,828.28	422,849.48	422,870.63	422,891.68	422,912.71	422,933.64	422,954.50	422,975.51	422,996.68	422,904.85	422,925.75
	Connectivity	3.86	11.53	13.53	16.82	18.23	22.09	24.09	29.52	32.02	32.74	34.74	46.86	45.58
	Silhouette	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.98	0.98	0.98
kmedoids	APN	0.00	0.10	0.11	0.14	0.14	0.14	0.15	0.15	0.16	0.16	0.16	0.16	0.16
	ADM	10,622.25	25,458.85	27,179.76	30,109.91	31,577.42	31,581.43	31,350.52	31,580.66	31,662.98	31,790.83	31,663.69	31,668.07	31,708.57
	FOM	422,786.18	422,806.49	422,686.58	422,708.06	422,751.59	422,768.74	422,661.05	422,690.55	422,696.29	422,717.51	422,632.32	422,653.60	422,662.08
	Connectivity	3.86	5.38	13.72	16.33	18.47	29.76	31.76	44.55	47.33	39.30	45.86	57.83	60.45
	Silhouette	1.00	0.99	0.99	0.98	0.87	0.88	0.88	0.88	0.88	0.88	0.73	0.73	0.74
EM	APN	0.18	0.27	0.49	0.38	0.46	0.47	0.52	0.49	0.48	0.50	0.53	0.48	0.51
	ADM	85,712.28	101,145.64	104,900.13	118,009.14	116,255.66	114,734.28	101,111.36	117,688.39	99,655.42	120,131.56	112,749.72	100,706.27	109,422.14
	FOM	422,785.45	422,788.48	422,793.21	422,807.16	422,790.86	422,860.83	422,604.51	422,635.49	422,573.50	422,601.98	422,525.90	422,590.41	422,520.98
	Connectivity	1,400.55	2,589.21	3,022.55	4,338.44	4,692.36	5,689.08	6,378.72	7,345.93	7,572.52	7,349.92	7,742.04	7,034.40	8,122.14
	Silhouette	0.39	-0.10	0.11	-0.33	-0.52	-0.52	-0.51	-0.51	-0.51	-0.46	-0.49	-0.45	-0.45

values of APN and ADM) and compact (lower connectivity and Silhouette width close to 1) comparing to k-medoids and EM algorithms. Given the similarity in evaluation results of k-means and hierarchical algorithms, we choose the former. It is computationally more efficient and it enables the iterative improvements in grouping of the hosting providers. We inspected the stability and internal metrics as a function of a number of clusters (see Table 3.1). Combined with our domain knowledge about the hosting sector, we grouped providers in 10 clusters using k-means.

3.4.2 Groups of hosting providers

Table 3.2 shows the groups: the size (number of providers), and the mean and standard deviation of each characteristic. Cluster 2 represents a group of the smallest hosting providers that are assigned on average a few to a few dozen IP addresses which are only used for shared hosting–the proportion of provider's domain name space and IP space used for shared hosting ($c3$ and $c4$) is above 97%. Note that the mean density of domains per shared hosting IP address ($c5$) is very high (1720), as is the standard deviation. Both are driven by `GoDaddy, LLC` with 385,757 domains registered to a single IP address (other sources, like `DomainTools.com` report this as well). When we closely investigate a sample of domains hosted on this IP, we come across several parked domains, which to some extent explain the density. Without this provider, the mean density drops to 178 (SD: 222). Providers in this cluster are mainly located in United States (97.2%). They offer a great variety of cheap or even free hosting services. For example, we observed an average of 1983 domains per IP hosted on 2048 IP

addresses of the OpenTLD Web Network TK organization (the `.tk` registry). Most of these providers include free plans with limited web space and data transfer under certain second-level domains. For a monthly fee of few euros a customer may obtain an unlimited number of domains under the most popular gTLDs as well as unlimited storage and bandwidth.

Clusters 4 and 8 contain somewhat larger providers (in terms of address space) such as 1&1 Internet. They offer more diverse services in comparison to the smaller ones. Around 80% of their addresses are used for webhosting, but only a small share of this space is used for shared hosting services (11% and 33%, respectively). The lower density of domains over shared IP addresses ($c5$) may suggest that they offer virtual private hosting as an extension for the hosting services. This type of service is usually unmanaged, i.e., the customer administrates the virtual system and software that runs on the server.

Cluster 10 is similar to clusters 4 and 8 in terms of the mean size of the IP range ($c1$) and the portion of IP space used for webhosting ($c2$), while a much smaller portion of the address space ($c3$) and domain name space ($c4$) is shared hosting. This suggest that providers in this cluster offer more non-shared (and thus expensive) type of services, such as dedicated hosting.

Clusters 3 and 7 are the next class in terms of size, moving from hundreds to thousands of IP addresses ($c1$). A smaller portion of the address space is for webhosting ($c2$)—40% and 33% respectively. The providers in cluster 3 use 7.6% of their IP address and 80% of their domain name space for shared hosting ($c3$ and $c4$). In cluster 7, on the other hand, providers have less shared hosting address space (0.49%) and only 9.85% of all domains are on shared addresses. Again, this suggests that providers in this group such as Go Daddy offer more dedicated or managed hosting services.

Similar conclusions could be drawn from a comparison of hosting providers in clusters 5 and 6. We move, once more, up one class in terms of size of the address space, where the portion of those addresses used for webhosting further diminishes. In contrast to cluster 5, the webhosting of providers in cluster 6 is mostly shared hosting. In comparison to other clusters, cluster 5 has the smallest shared hosting portion of its IP address space and domain name space.

Cluster 1 is similar to cluster 5 in terms of the allocated IP space ($c1$) and the portion of IP space used for webhosting ($c2$) while a bigger portion of domain name space ($c4$) in this cluster is shared hosting.

Cluster 9 with around 17% of the total providers in the data, mostly contains providers with the largest allocated IP space ($c1$) and a small portion of the address space used for webhosting ($c2$) such as Telecom companies. The values of percentage of IP space and domain names space used for shared hosting ($c3$

Table 3.2: Hosting provider groups

		Mean (Standard Deviation)				
Cluster	**Size**	c1	c2	c3	c4	c5
1	7,413 16,81%	48286.64 (263086.31)	4.36 (4.90)	0.15 (0.25)	31.20 (8.16)	25.97 (19.41)
2	250 0.47%	28.44 (205.85)	99.62 (2.46)	97.77 (7.43)	99.68 (1.79)	1720.09 (24387.17)
3	3,771 7.94%	2441.18 (29297.19)	40.03 (10.69)	7.58 (4.51)	80.64 (15.00)	114.57 (402.76)
4	1,748 3.35%	210.35 (2455.62)	81.00 (13.86)	10.92 (4.75)	75.93 (15.72)	117.74 (1354.02)
5	13,367 29.01%	48775.60 (377535.49)	4.77 (4.90)	0.01 (0.03)	1.96 (4.38)	4.08 (10.67)
6	6,657 15.02%	16594.20 (101181.78)	6.83 (6.62)	1.31 (2.26)	85.43 (8.33)	391.45 (3946.55)
7	2,550 5.90%	5948.00 (75045.93)	33.08 (9.98)	0.49 (1.01)	9.85 (14.34)	11.30 (22.71)
8	988 1.88%	459.95 (9557.63)	79.14 (21.34)	32.82 (10.71)	91.78 (9.17)	113.18 (1006.72)
9	7,389 17.07%	307011.67 (4327995.09)	4.95 (5.75)	0.35 (0.54)	57.85 (8.40)	42.40 (46.62)
10	1,301 2.55%	679.87 (6270.40)	79.21 (15.22)	0.98 (1.98)	8.13 (13.89)	8.10 (20.77)

c1: IP address range size
c2: Percentage of IP address range used for hosting websites
c3: Percentage of IP address range used for shared webhosting
c4: Percentage of domain names on shared webhosting
c5: Density of domains on shared webhosting IP addresses

and *c4* respectively) suggest a significant portion of the webhosting in this group is shared hosting.

These results, while crude, allow us to distinguish groups of providers with different profiles, from small companies that offer cheap webhosting on highly dense shared servers from those providers that offer more expensive and flexible services, such as managed and dedicated hosting.

Finally, we analyze the geographical location of the providers in each of the clusters. Most of the providers in clusters with smaller average IP ranges are located in United States while clusters containing providers with larger IP range sizes are evenly distributed across different countries.

We expect that different groups of providers offering various types of hosting services handle domain abuse differently, which is then examined in terms of

uptimes of phishing domains discussed in Section 3.5.

3.5 Case Study: Analysis of Uptime for Phishing Websites

In the previous sections, we grouped the hosting providers into 10 different clusters with different business profiles. In this section, we examine whether these profiles are associated with differences in abuse handling, more specifically, the speed with which phishing websites are taken down.

3.5.1 Phishing data

We analyze data on the uptime of phishing websites from the moment the provider has been notified, which was generously provided to us by Cyscon GmbH [128].

Table 3.3: Summary of phishing data points per cluster

Cluster	Providers	ASes	FQDNs	URLs	IPs	Countries
1	221	229	633	3234	367	63
2	24	6	425	556	241	4
3	453	357	29641	78592	8036	54
4	86	41	689	1418	344	19
5	82	84	210	2521	134	43
6	938	893	10265	30638	4998	84
7	47	48	465	1400	229	21
8	48	19	1130	1634	734	11
9	483	504	4165	13957	1677	77
10	12	12	155	482	98	6

The dataset contains 137,577 phishing URLs associated with 48,224 fully qualified domain names (FQDNs) that were hosted on 17,279 IP addresses in 1,962 ASes located in 114 countries. Each websites is then tagged with the first and last time it is seen online. Note that for the websites that are only seen once, the first seen is the same as the last seen, indicating that they were taken down before the second measurement moment. These are logged as having an uptime of 0 hours. The data contains websites that were first seen between June 4 to August 16, 2015. Many of the targets are known brands such as `Paypal`, `Dropbox`, `Yahoo`, or `Wells Fargo`, `World of Warcraft` and `Battlenet`.

We mapped the phishing data to the different clusters of hosting providers

discussed in section 3.4. Table 3.3 displays the distribution of the data across the different clusters.

3.5.2 Analysis of uptime

An important criteria to evaluate security performance of hosting providers, is how fast they respond to being notified about malicious sites [34]. Uptime has been used in previous security research as a standard metric for studying lifetime of different attack types [65, 129, 130].

We define "uptime" of a phishing website as the number of days between the first and last time the phishing site is observed online and reported by Cyscon. Some of the phishing sites remain online beyond the measurement period, which leaves their uptime unknown. To correctly account for these cases, we analyze uptimes through survival analysis with right-censoring.

The survival function $S(t)$ expresses the probability that a phishing website is online at a specific time during the observation period. It is calculated at time t using the standard Kaplan-Meier estimator without any assumption about the distribution of the underlying data [131].

Figure 3.4 shows survival curves for phishing websites in the different provider clusters. In Table 3.4 we present descriptive statistics on uptimes, based only on sites that had been taken down by the end of our measurement period.

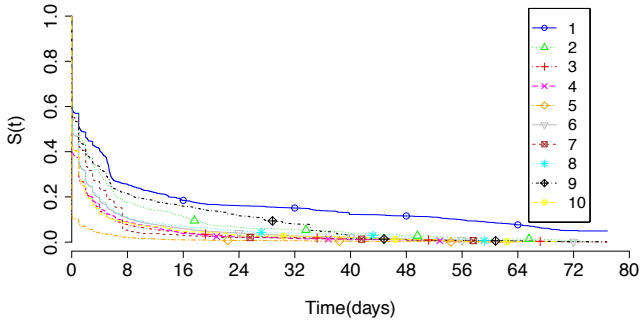


Figure 3.4: Kaplan-Meier estimates per cluster

The differences among the survival curves are highly significant, not only across the population as a whole, but even when performing pair-wise comparisons among all clusters. Figure 3.5 displays the results for log-rank non-parametric tests [132]. Only the blank tiles indicate non-significant differences

Table 3.4: Descriptive statistics of uptimes (hours) per cluster

Cluster	Min	Mean	Median	Max	SD	Coef Var	SE
1	0	165.840	24.000	1,744.400	344.390	207.660	6.315
2	0	142.360	24.294	1,813.800	301.530	211.810	12.393
3	0	59.408	0.0003	1,829.900	175.280	295.050	0.627
4	0	62.560	0.0003	1,505.200	175.220	280.080	4.650
5	0	16.715	0	1,542.600	98.499	589.290	1.960
6	0	80.498	0.002	1,812.800	210.080	260.980	1.176
7	0	76.794	24.004	1,723.100	182.630	237.820	4.876
8	0	95.504	5.005	1,730.100	249.650	261.400	6.168
9	0	152.790	24	1,840.800	276.420	180.910	2.307
10	0	70.064	0.0003	1,671.600	205.000	292.580	9.318

at a 0.05 significance level. In other words, the different clusters are associated with different security performance. This underlines the value of the preceding work of mapping and then condensing the complexity and heterogeneity of the hosting market. Explaining the differences in uptimes from the properties of the providers in the clusters is beyond the scope of this chapter and is further discussed in Chapter 5. We can, however, explore what these results show, without drawing any hard conclusions.

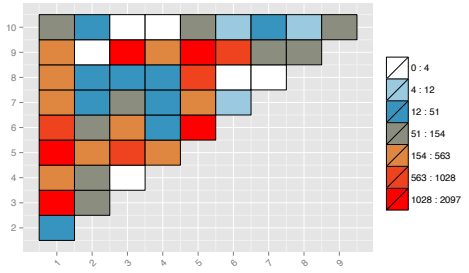


Figure 3.5: Log-rank test for cluster pairs

Figure 3.4 and table 3.4 shows that phishing websites in clusters 1 and 9 have the highest survival rate – in other words, these clusters perform the worst in terms of take-down speed. Clusters 1 and 9 contain the largest providers in the market (together with cluster 5, see table 3.2). Providers in these two clusters have a relatively low percentage of webhosting, but a significant portion of that webhosting is shared hosting. Around half of the phishing sites in these clusters indeed map to shared hosting servers. The third-worst performer is cluster 2. This contains the providers with the smallest allocated IP ranges, which are

used completely for shared hosting services.

An intriguing contrast emerges when looking at the best performer: cluster 5. It is very similar to 1 and 9, except for the fact that it contains virtually no shared hosting. It is too early to draw any conclusions from these findings, but it seems clear that size of the allocated address space itself does not explain performance. Perhaps it is more related to the position and size of shared hosting services in the overall portfolio. This is consistent with earlier security research that focused on shared hosting as a problem area. The underlying economic mechanism would be that this part of the market is driven by fierce price competition and low profit margins.

Whether the uptimes of phishing sites are really related to the incentives and practices around shared hosting is a question that we further explored in Chapter 5. In a more general sense, our findings demonstrate that a better mapping of the market and its providers will allow us to focus security efforts in the most urgent areas, as well as allowing us to compare apples to apples when evaluating the security of different providers.

3.6 Related Work

To the best of our knowledge, there is no study that has systematically and transparently mapped the hosting market. Recent work by Noroozian et al. underlines the need for such mapping, by demonstrating how provider heterogeneity influences security performance metrics [114].

A number of studies map security incidents to hosting providers by equating them with ASes and normalizing the incidents by the AS size [115]. Mahjoub studies the concentration of maliciousness in ASes by analyzing AS topology, hosted content and IP space reservation [133]. Other studies identify malicious ASes using AS topology, BGP-related features and by exploring ASes providing transit for malicious ASes [33, 32]. Although useful, these studies neglect the organizations within ASes and their properties, which influence all metrics of maliciousness.

Industry is more active in producing rankings for ASes as hosting providers e.g., [36]. Netcraft's uses reverse DNS to map providers, but the complete methodology and data are not available to researchers [116]. Canali et al. examine the security performance of a small group of shared hosting providers and conclude that the majority of the providers are unable to detect even basic attacks on their networks [10]. Although they study providers with specific characteristics, the sample of providers is non-random and too small to draw

any conclusions about providers in general.

A separate branch of research focuses more on how hosting providers deal with the take-down of malicious websites [34, 134]. Nappa et al. explore lifetime of drive-by download URLs and rank their associated ASes [135]. Moore and Clayton study lifetime of domains and variables like hosting providers of the website that might influence take-down speed and conclude that website removal is not yet fast enough to completely mitigate the problem of phishing [129]. Gañán et al. examine characteristics of botnet C&Cs that might influence their lifetime [136]. Again, treating providers as ASes, the paper concludes that hosting provider, hosting types (e.g., bulletproof or free) and popularity of the sites are significant factors associated with the uptime of the C&Cs.

We believe that this work is the first to map the hosting market and discuss its heterogeneity by analyzing the differences among the providers in terms of their services and their abuse handling practices.

3.7 Conclusions and Discussions

A variety of initiatives seek to improve security in hosting services, but none of them have taken even basic information about the market into account, which makes it hard to identify best practices and evaluate performance. Security research has mostly relied on routing data and AS-level aggregations of security incidents, equating ASes to providers. To overcome these limitations, we have developed a systematic approach to uncover and grasp the complexity of the hosting market. We combined passive DNS data to determine the address space of hosting infrastructure with WHOIS data to determine the associated providers and their IP address space. Next, we applied several filters to conservatively remove false positives (non-hosting providers).

This process resulted in a set of 45,434 hosting providers, somewhat log-normally spread around a median size of 1,517 IP addresses. Using five provider characteristics we extracted from the data, we familiarized the reader with the hosting landscape. There is surprisingly little consolidation in the market, given that the services are commoditized and thus amenable to economies of scale, as can be seen in the market for cloud services. In hosting, it takes 1,210 providers to account for 80% of the address space used for webhosting. A large number of small players dominate the landscape as much as a small number of larger providers. There are providers with millions of IP addresses and around a thousand with a handful or even just a single address. We found providers who are offering only webhosting versus those who are using only a small share

of their allocated address space for webhosting.

We explored what combinations of the characteristics occur in reality via cluster analysis. This uncovered a diverse set of business profiles and an indication of what fraction of the market fits each profile. Since these profiles are proxies for different types of organizations, we assessed whether the clusters were associated with different security performance using data on the uptime of phishing websites. The clusters were indeed very different in how fast they take down phishing domains. The results suggest that our mapping of the hosting market is helpful in deepening our understanding of the driving forces of security threats, as well as in developing best practices. Both benefit from being able to compare apples to apples, rather than using the current crude analytical approaches based on routing data and AS-level abuse metrics, which cannot account for the heterogeneity in the market.

Several limitations need to be acknowledged. These results are just a first step towards a thorough understanding of the market. We assumed that all providers offer at least some webhosting in their portfolio, so as to be able to use passive DNS data to identify potential providers. There might be some providers who do not offer webhosting. They would be invisible to this approach. Another limitation is the fact that WHOIS records are notorious for containing stale, inconsistent and inaccurate data. Related to this are the inconsistencies in organization names in the WHOIS data. When different names point to the same entity, they might actually be operated under one entity or they may point to entities belonging to the same parent company but operating independently of each other. How to distinguish these two cases is still unsolved. The accuracy of the filters to separate hosting providers from other entities that host websites is rather limited and this impacts the mapping.

In future chapters, we will study different factors that can explain the significant differences that were found among the clusters of providers. The characteristics of providers can also be enriched by adding other variables that might shape their incentives and performance, such as their jurisdiction, privacy and security regulations and development indicators.

The map of the hosting provider landscape that has been developed in the course of this study will be made available to other researchers, so as to contribute to better analysis and mitigation of the security threats that plague this market.

CHAPTER 4

Measuring the Impact of Providers' Structural Properties on Abuse

In the previous chapter, we established a reliable methodology for identifying hosting providers. We also found that providers' business models correlate with their security performance, as measured by the speed with which they take down phishing webpages on their network. For example, shared hosting providers with smaller IP address ranges tend to take more time to take down a compromised domain than providers with other business models. In this chapter, we take a more fundamental approach. We analytically and empirically disentangle the different factors that cause variation in measured abuse rates across providers. In the first part of this chapter, we develop an analytical model that outlines different factors at work in explaining security of hosting providers as observed by abuse incidents. The model includes two types of provider properties that drive abuse: exposure, as measured by structural properties like size and business model, and their security efforts. In the second part of this chapter, we focus only on providers' exposure (structural properties) and statistically quantify their impact on the incident rates for one type of abuse, namely phishing.

4.1 Introduction

Abuse data is an important foundation for security and policy research. It associates technical identifiers – typically IP addresses, domain names or URLs – with malicious activities, such as spam, infected machines, command-and-control servers, and phishing sites.

Scientific studies and industry reports draw on abuse data to make inferences about the security practices of the parties in charge of the networks or services where the abuse is located. Concentrations of abuse are seen as evidence of

poor security practices or even criminal behavior, explicitly or implicitly characterizing certain providers as "rotten apples", or at least as actors who can and should do more remediation [36, 137, 50, 56, 34, 115, 114].

Industry representatives often counter these kind of incomplete/wrong inferences based on right abuse data, when they make media headlines. For example, a 2013 McAfee report ranked the Netherlands as number three worldwide in terms of hosting botnet command-and-control (C&C) servers [138]. A leading news organization concluded: "Netherlands Paradise for Cybercriminals", prompting a debate that reached the national parliament [139]. The Dutch Hosting Provider Association responded that it "disagrees vehemently" with this conclusion [140]. It argued that the high ranking for hosting C&C servers was an artifact of the large hosting infrastructure in the country, not of any negligence or malice on the part of providers.

The hosting provider association raised a valid point. We know that concentrations of abuse are, to a large extent, a function of the size of the network and the service portfolio of the provider, rather than being indicators of the provider's security effort [141]. Previous research looked into the effect of size – as one of the providers' structural properties – on abuse levels [34, 115]. A common problem in these studies is that size is controlled for by dividing the number of abuse events by the number of IP addresses associated with an Autonomous System (AS). This is not only a naive normalization approach, considering all other size indicators that could influence abuse counts, but also contains errors in aggregation and attribution of abuse. It is also problematic to use Autonomous Systems (ASes), who are entities responsible for routing IP addresses, as proxies for hosting providers, the organizations who operate the IP space. Moreover, previous work does not take into account other inherent or structural properties of providers, such as pricing strategies or the type of the hosting service offered. All of these can potentially influence abuse [35].

Therefore, to advance our ability to make reliable inferences from abuse data and to address the limitations of prior work, in this chapter, we develop an analytical approach and propose a statistical model of the abuse data generation process. The model helps to understand to what extent abuse levels are determined by structural properties of providers versus being mainly determined by other factors including, but not limited to, the security efforts of individual providers. Structural properties in our study are different size and business model variables, pricing strategy, time-in-business, popularity index, WordPress use and ICT development index.

We use phishing abuse data as a case-study to demonstrate our approach. In short, this chapter makes the following contributions:

- We present a scalable approach to reduce attribution error in studying abuse across the population of hosting providers
- We propose an analytical and statistical approach to explain abuse concentrations. Our model improves on previously utilized naive normalization methods, through decomposing the different sources of variance present in abuse data such as providers' characteristics, attacker behavior and measurement errors.
- In a case study on phishing data, we show that more than 84 % of the variance in abuse data can be explained by four size and business model properties of providers, collected for the entire population of hosting providers;
- We develop an approach to measure the impact of factors that are difficult to observe at scale. Using "statistical twins", we present the first empirical evidence of the impact of pricing, time-in-business on phishing abuse. Together with other factors related to providers' business models, we are able explain a further 77 % of the remaining variation in abuse, while controlling for country-level differences;
- We demonstrate how our approach generates comparative abuse metrics by controlling for the structural differences among providers. Such metrics are more suitable to evaluate policy impact on concentrations of abuse than absolute counts or naively normalized metrics.

While our study provides an unprecedented view into the interpretation and attribution of abuse in the case of hosting providers, a limiting factor is that we measure only structural properties. We obtain security effort, the variable of key interest, as an unobserved residual, which is conflated with measurement noise. As a result, this indirect approach leaves us with upper bound estimates of the effect of security effort on abuse.

In Section 4.2, we outline the analytical approach that sets up the rest of the chapter. Section 4.3 describes data sources and collection methods. Section 4.4 details our general modeling approach and present results for the entire sample of hosting providers. In Section 4.5, we explain the "statistical twins" approach and present results for the subset of enriched data points. In Section 4.6, we evaluate the robustness by assessing the impact of measurements errors in the abuse data and size estimates. Section 4.7 revisits related work, structured by the level of analysis. Finally, we discusses our main conclusions and implications in Section 4.8.

4.2 Analytical Approach

To explain the driving forces behind abuse observations at the level of hosting providers, we need to disentangle several sources of variance. Figure 4.1 summarizes these sources of variance and provides an overview of our approach.

4.2.1 Decomposition of sources of variance

We assume that abuse observations are broadly driven by two phenomena: explanatory factors (left branch) and measurement errors (right branch).

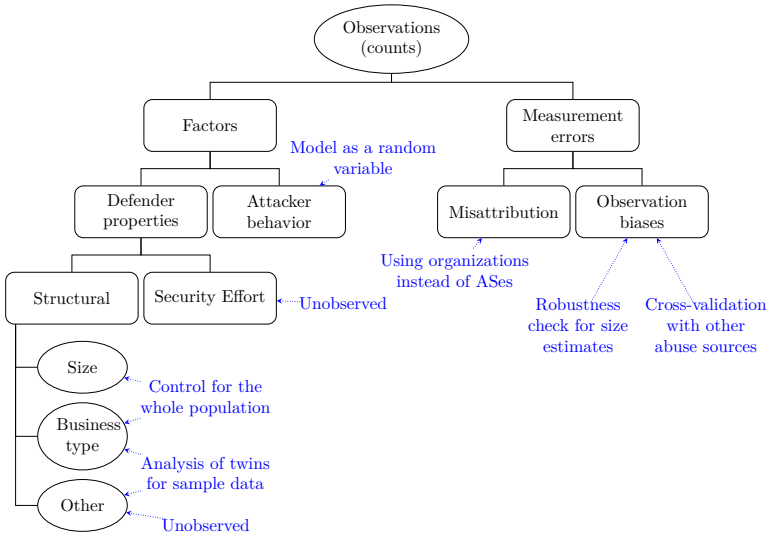


Figure 4.1: Overview of our analytical approach

We further divide the explanatory factors into defender properties and attacker behavior. Defender properties are then branched into two main groups: structural properties and (indicators of) security effort. The most relevant structural properties are size metrics. Another structural property is the type of business. Providers offer different types of hosting services, e.g., unmanaged, dedicated, shared, virtual private server, etc. These services differ in the amount of responsibility the provider assumes and in its role of providing security [9]. Business type also includes the pricing strategy. Other characteristics include

the legal framework under which providers operate as well as the overall maturity of ICT development in a country. Last but not least, providers' security efforts influence the abuse counts.

Next to explanatory factors, we distinguish measurement errors that cause variance in the abuse counts: biases in observations (for factors and abuse data) and problems in attributing abuse incidents to the responsible economic entity.

To asses the influence of each of the above mentioned factors in explaining the variance in the abuse counts of hosting providers, we develop a statistical modeling approach and implement it for one type of abuse data as a case-study: phishing domains. The blue text in Figure 4.1 indicates how we deal with the various sources of variance as decomposed in the analytical approach.

4.2.2 Model of the abuse data generating process

Abuse counts consists of non-negative integers that arise from a direct observation of a point process in a given time period, suggesting to model the data generation process as Poisson distribution. The number of phishing domains per provider enters a Poisson Generalized Linear Models (GLMs) as dependent variable. Structural defender properties related to size and business type (detailed in Section 4.4) are used as independent variables. We improve on the size estimates used in previous work in two ways: (i) we improve attribution by calculating size estimates for hosting providers instead of technical entities, like ASes, (ii) in addition to the size of the allocated IP space, we also include the sizes of the domain name space and the IP space used for webhosting in the regression. This allows for more precise control of providers' customer base and attack surface. All these variables can be economically collected for all hosting providers.

Other structural properties (detailed in Section 4.5) are more cumbersome to collect, since they require manual work or are costly to others when measured at scale. A statistician's response would be to estimate from a random sample. The size of a random sample depends on the target level of confidence and on the effect size (akin the signal-to-noise ratio). In the case of hosting providers, the heterogeneity in the population may hide subtle effects of security effort, which would require uneconomically large samples to control for.

A more efficient approach is to modify the sampling strategy and select subsets of cases which appear homogeneous according to the observable structural properties. Specifically, we select a set of "statistical twins" – subsets of size two – covering the domain of the known population. We collect additional variables for each twin. The subsequent analysis looks for factors explaining differences

within twins, disregarding differences *between* twins. Technically, this can be achieved by adding one fixed effect per subset to the GLM specification. This method allows us to control for large parts of the heterogeneity and at the same time account for linear bias introduced by the systematic sample selection. Another way of looking at this approach is that we select the a priori most informative cases from the population for further analysis. It rests on the implied assumption that cases which have a twin in the population do not systematically differ from those which do not. We conjecture that this is not unreasonable for the population of global hosting providers.

Of course, there always remain unobserved factors, including additional structural properties, variables related to security effort, attacker behavior, and all kinds of measurement error. Although all unobserved factors are conflated in the unexplained variance of the model, below we explain how we deal with each of these factors.

4.2.3 Security effort as residual

We model security effort as an unobserved variable because direct measurement of security effort for hosting providers is very difficult. First of all, there is no way to directly observe the actions of providers' security staff, such as applying security patches, educating customers, and putting application firewalls in place. At best, we can observe some technical indicators that might be influenced by those efforts, but that will always constitute a very partial measurement. Second, even if certain technical indicators can be collected as proxies for effort, they are not necessarily causally related to abuse. Indicators used in prior work measure network hygiene (e.g., BGP misconfiguration, untrusted certificates, open mail relays). This type of hygiene is not preventative of web compromise. At best, such hygiene indicators can be interpreted as measuring some more generic security effort, which might or might not correlate with providers' effort in fighting abuse. Third, useful technical proxies for effort are still hard to attribute to providers. Who actually caused the presence or absence of the vulnerability? The web master, the hosting provider, the software vendor, or someone else? Fourth, and last, even if useful indicators can be observed and correctly attributed, it is hard to draw the correct inference from them because of the heterogeneity in the market. Hosting providers are very different and so are domains within providers. An indicator that signals lack of security effort for one domain and provider might not signal the same for another domain or provider, because the affected users may face very different threats. For instance, not all websites are more secure by adopting HTTP Strict Transport

Security (HSTS) or Content Security Policy (CSP). And many providers cannot cause BGP misconfigurations, as they do not operate their own AS and, hence, BGP announcements.

In short, measuring effort is very difficult under the best of circumstances. This makes it valuable to develop an alternative approach to approximate effort that does not depend on direct observation. In this work we explore the option of treating security effort as residual, i.e., as part of the unexplained variance, after accounting for the observable factors. As a consequence, our results must be interpreted as upper bounds for the effect of security effort on abuse. This introduces some asymmetry in our research logic: only if the residual variance is small, we can rule out the hypothesis that security effort is very influential.

We model attacker behavior as random variable, assuming that most attackers behave rather opportunistic than strategic. The realization of this random variable is also part of the unexplained variance. The realization of discrete incidents can be interpreted as part of the measurement error of an underlying latent variable of "attack strength", or as the result of seemingly random attacker behavior. To manage expectations, we note that our statistical approach is generally not suitable for studying targeted attacks and other rare events.

Avoiding observational biases is outside the scope of this chapter, but we try to limit their effect on our core results by cross-validation against a different set of phishing data. In addition, we test the robustness of the size estimates in the model against errors due to possible model mismatch by simulating the impact of distorted size estimates against a simulated phishing count variable drawn from an ideal Poisson distribution model.

4.3 Data Collection Methodology

Our study uses a variety of data sources, which are summarized in Table 4.1.

4.3.1 Mapping to hosting providers

Our goal is to accurately identify the IP ranges that belong to hosting providers. Most of the existing work uses BGP data to map IP addresses of abuse incidents to ASes, equating the latter with hosting providers. However, the entity that is routing an IP address is not always the same as the organization that is hosting an IP address. While some organizations operate under several ASes, other organizations share a single AS. Our prior work in Chapter 3 finds that on average an AS consists of 7 organizations [142]. WHOIS registration and

IP allocation information, which is collected and stored by Regional Internet Registries (RIR), provide more direct visibility into the responsible organization behind an IP address. It should be noted that WHOIS data comes with its well-known limitations, such as different data formats, they are less detrimental on analysis of the hosting market than starting with BGP [119].

In this chapter we build on the methodology introduced in the previous chapter and slightly improve our True Positive Rate (TPR) by discarding more organizations that are not hosting providers. We take 47,446,082 IP addresses and 214,138,467 domain names observed in passive DNS data in 2015 and map them to 161,891 corresponding organizations to whom they are allocated, using the MaxMind API for WHOIS IP allocation, as discussed in previous research [120, 143, 144]. The resulting list contains all organizations to whom IP ranges are allocated. Many of them are not hosting providers, e.g., `Massachusetts Institute of Technology` and `DoD Network Information Center`. We compile a final set of organizations that offer hosting services by filtering out non-hosting providers through a series of keywords related to educational and government-related organizations, ISPs, broadband providers, mobile service providers, domain parking services and DDoS protection services [142, 144]. The final set consists of 45,358 hosting providers.

4.3.2 Abuse data

In order to demonstrate the application of our proposed analytical method, we model the count of abuse in the networks of hosting providers, using phishing data as a case-study. The main reason behind this choice is, since phishing sites are known to be mostly compromised accounts [49], bypassing security is very much required in the bulk of cases. To that end, we analyze phishing domains collected from two sources: the Anti-Phishing Working Group (APWG) [75] and Cyscon GmbH [128].

APWG data contains URLs used in phishing attacks together with their blacklisting times. We collect the IP addresses associated with second-level domains[1] in the APWG feed by retrieving the corresponding passive DNS entry at the time when the domain is blacklisted. The final set consists of 131,018 unique second-level domains and 95,294 unique IP addresses hosted by 5,391 hosting providers for the entire 2015.

Cyscon phishing data contains the same attributes (URLs, IP addresses, blacklisting times). We collect 40,292 unique second-level domains and 23,021 unique IP addresses hosted by 2,782 hosting providers in June–December 2015.

[1] Domains such as `example.co.uk` are considered to be second-level domains as well.

We use the phishing second-level domains in APWG data as the default response variable in our regression models in Section 4.4 and 4.5. In Section 4.6, we use the phishing second-level domains in Cyscon data to cross-validate the results.

4.3.3 Provider properties

To explain the differences in phishing incident counts between hosting providers, we collect a number of variables on provider (defender) properties, some for the entire population and some for the sample of "statistical twins".

Variables collected for all providers

In addition to identifying providers, we can collect variables related to size and business model (see the leftmost factors in Fig. 4.1) from passive DNS and WHOIS data.

Number of assigned IP addresses. Size of IP address block(s) assigned to a provider based on WHOIS.

Number of IP addresses hosting domains. Count of IP addresses seen to host domains in passive DNS. The combination of these two variables provides information about the kind of business the hosting provider is running. For instance, providers who use a large part of their assigned IP space for hosting domains such as webhosting providers can have a different business model from providers who use their IP space for hosting other services such as data centers.

Number of hosted domains. Count of the second-level domains in the passive DNS data. In addition, note that since the first three variables have a skewed distribution, we log-transform them with base 10.

Percentage of domains hosted on shared IPs. We consider an IP address shared, if it hosts more than 10 domain names [142].

This variable measures the ratio of domains that are hosted on shared IP addresses over the total size of the hosted domains, in percent. This variable not only conveys information about the size of the shared hosting infrastructure of the provider, but also about the provider's business model: the degree to which it relies on low-cost shared hosting services.

"Statistical twins" sampling method

It is not possible or desirable to collect data at scale for all factors in Figure 4.1. For example, pricing information must be collected manually. It involves search, interpretation, and human judgment. Applying a standardized procedure is too

Table 4.1: Descriptive statistics of variables used in the full model for all providers and providers in the sample of twins

	min	**mean**	**median**	**max**	**sd**
for all data points (n = 45,358)					
# assigned IPs [log10]	0	3.1	3.2	8.4	1.2
# IPs hosting domains [log10]	0	1.8	1.7	6.2	0.8
# hosted domains [log10]	0	2.0	1.8	7.6	0.9
% domains hosted on shared IPs	0	51.0	59.0	100	37.1
# phishing domains in APWG	0	2.8	0	11,455	91.3
# phishing domains in Cyscon	0	0.9	0	5,515	37.4
for statistical twins (n = 210)					
# assigned IPs [log10]	0.3	4.0	4.0	7.5	1.4
# IPs hosting domains [log10]	0.3	3.0	3.0	5.6	1.2
# hosted domains [log10]	1.5	3.9	3.7	7.6	1.2
% domains hosted on shared IPs	0	78.6	87.9	99.3	22.3
# phishing domains in APWG	0	159.2	3	9,805	967.6
# phishing domains in Cyscon	0	54.8	1	3,819	375.0

costly for the entire population of 45,358 hosting providers for some of these variables. Even collecting some technical indicators, such as the number of Wordpress installation on *all* websites of *every* hosting provider, are inefficient to collect in bulk.

For this reason, we employ a data-driven sampling strategy and select a small set of homogeneous "statistical twins" for which we can collect as much information as possible. The steps are:

(i) We start with a set of randomly and uniformly selected *seed data points* (105 hosting providers) for which we have collected pricing information. Let $\mathcal{S}$ be the set of seed data points and $\mathcal{T}$ be the total set (or population) of providers. The random seed should ensure a good coverage of the population.

(ii) We calculate the distance between all the data points in $\mathcal{S}$ and data points in $\mathcal{T}$ using the Euclidean distance between all explanatory variables collected for the entire population. This results in a distance matrix of 105 rows and around 45 k columns.

(iii) For each of the 105 providers in $\mathcal{S}$, we select the closest match; that is the provider in set $\mathcal{T}$ that has the minimum distance to the provider in set $\mathcal{S}$, in terms of variables in Table 4.1.

(iv) This results in a set of *rich data points* $\mathcal{R}$ consisting of 105 homogeneous statistical twins and 206 unique hosting providers in total, where a few providers became part of two twins. We further use set $\mathcal{R}$ as our (notoriously biased) sample to study the effect of additional factors on abuse. We account for the bias in the analysis (Section 4.6).

(v) We collect additional variables for all elements in $\mathcal{R}$.

While the method is economical and increases the information gained per effort, it comes with some limitations. First, it is unlikely to spot outliers simply because there is nothing to pair up against a unique provider like Amazon. Second, it assumes that data points in dense regions of the population do not systematically differ from those in sparser regions, where the probability of finding twins is lower. Third, it requires that the bias introduced by the selection strategy is approximately linear. Non-linear bias correction is possible in principle, but requires prior information on the functional form.

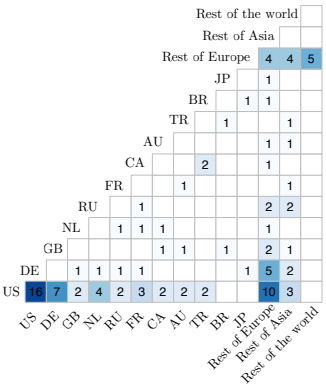


Figure 4.2: Number of twins per country combination

Variables collected for a sample of providers

The below variables are collected only for the providers in $\mathcal{R}$, as explained in Section 4.3.3.

Country. For the providers in $\mathcal{R}$, we use MaxMind API to identify where the majority of IP addresses are located. In Figure 4.2, we show the general coverage of twins in our data. For instance, there are 16 twins with both providers located

in US while there is one twin with one provider located in Brazil and the other in Japan. The figure indicates a good variation in the geo-location of twins in our sample data.

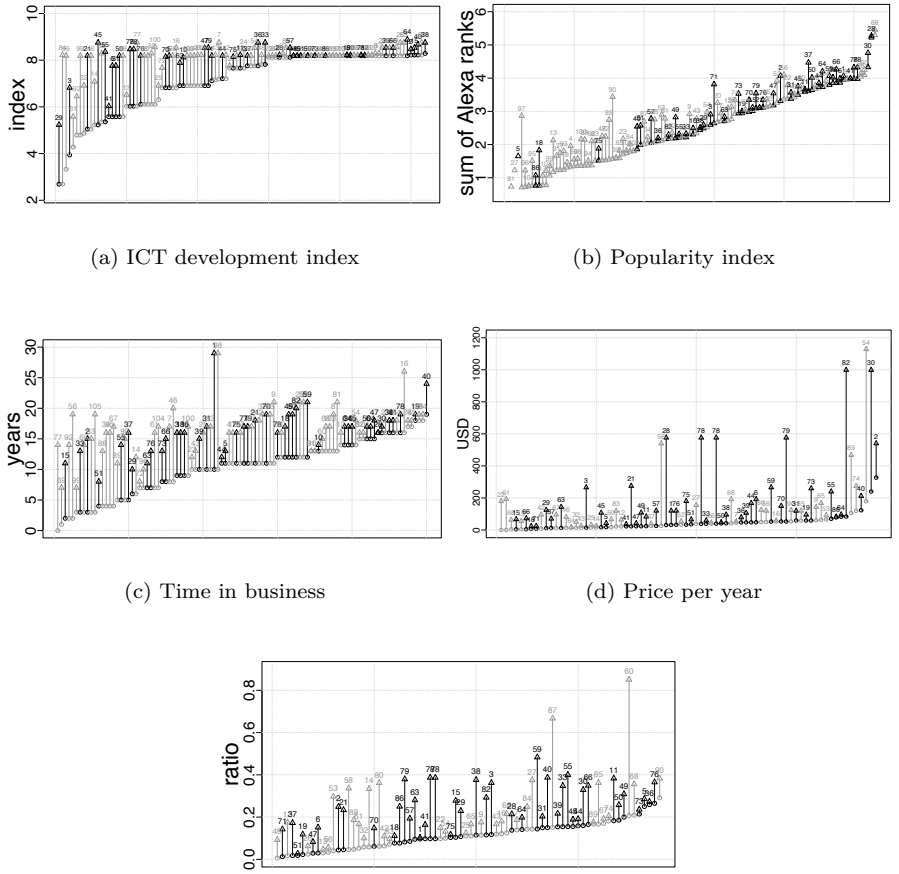
ICT development index. The ITU publishes country-level variables that proxy the development in information and communication technology (ICT) using several indicators [145]. (This and the previous variable came into view for our enriched data points, but they could also be economically collected for the whole population.)

Popularity index. We use Alexa's one million top-ranked domains as a proxy for online popularity. A provider is assumed to host more popular content when more domains are on the list. If attacker behavior is not completely random, one may expect that providers with more popular sites are targeted more in order to compromise domains and set up phishing sites. The popularity index per provider in calculated by aggregating the Alexa ranks of second-level domains in our sample $\mathcal{R}$. We first reverse the Alexa ranks per domain, that is the most popular Alexa domain gets the rank 1,000,000. We then calculate a score per provider by summing up the base-10 logarithm of the reverse rank of all ranked websites. This score combines website (i.e., customer) popularity and the density of popular sites at a hosting provider and the method allows us to account for extreme popularity of large providers.

Time in business. Time in business is a proxy for how experienced a provider is. We collect this information by querying the WHOIS database for the registration date of the provider's own domain name. Missing values were entered if we could not find the website or there was no public registration date in WHOIS. We cross-checked the results with the Internet Archive for all data points [146]. Almost all domains in our sample were captured by Web-archive a couple of months after the day they were registered.

Pricing. Finding comparable pricing information for hosting providers is complicated. We visit the provider's website and collect prices for the least expensive hosting plan on offer, as an indicator to tell 'bottom-end' from 'top-end' apart. We hypothesize that providers with cheaper hosting plans have fewer resources and more vulnerable customers, so higher phishing counts. We converted all prices to US dollars by taking the 2015 average exchange rate of the local currency to USD. Price information for providers is missing if (i) we could not find the provider's website; (ii) the prices are not available online; and (iii) we do not receive a reply to our inquiries through other channels.

WordPress use. Previous studies suggest that domains with popular content management systems (CMS) in general have higher odds of being compromised for phishing attacks [44]. Therefore, we use WordPress as a proxy of popular



(a) ICT development index

(b) Popularity index

(c) Time in business

(d) Price per year

(e) Wordpress use

Figure 4.3: Visualization of the variation of additional variables within and between sampled twins

and targeted software and hypothesize that providers with such business models have higher chance of incurring a phishing incident. More specifically, the more WordPress websites a provider hosts, the higher the chance of a compromise

that is used for a phishing site. Note that we collect this indicator to capture information about the business model. One could also collect indicators of software installations and patch levels to measure to measure the impact of provider's patching practices on abuse, but this is outside the scope of this chapter.

To collect data on WordPress installations per provider, we randomly sample 2 % of the domains of providers in $\mathcal{R}$. This results in 1,398,928 domains. We use WPScan, a black box WordPress vulnerability scanner developed by Sucuri, to determine if a domain uses WordPress [147]. The variable "WordPress use" is calculated by taking the ratio of scanned domains with a WordPress installation over all scanned domains, excluding those that we were unable to scan.

To raise confidence in the selection of twins and facilitate the interpretation of estimated coefficients, in Figure 4.3 we show a collection of plots visualizing the variance within and between twins for the additional variables. Circles display the value of the provider with lower value in a twin, and are the basis for sorting all twins horizontally. Triangles display the respective value of the provider with higher value.

In general, the plots suggest a broad enough coverage of data points in the twins. More specifically, as Figure 4.3(a) demonstrates, variance of the ICT development index decreases with increasing the base level (The provider shown with circle, those with lower value in a twin). This is probably due the concentration of hosting providers (and hence twins) in a few large and highly developed countries, as witnessed in the country pairings displayed in Figure 4.2. The popularity index exhibits moderate differences at all levels (Fig. 4.3(b)). This indicates that the sampling strategy accounts well for the heterogeneity in the size of hosting providers, which is also included in the index calculation. At the same time, the remaining variation allows for the statistical identification of potentially influencing factors within the twins. Also time in business is very "healthy" in this regard, with a smaller difference for values between providers in twins which are in business since the `.com` ages (Figure 4.3(c)). The differences in price are rather small and exhibit occasional spikes (Figure 4.3(d)). This may reflect the generally low cost of the cheapest package of one of the twins, differences in business models at the spikes, and potential issues related to comparing US dollar amounts among countries with very different labor and infrastructure costs. Finally, the Wordress use also shows a good mix of variation within and between twins, increasing the chance of extracting a meaningful signal if the indicator has explanatory power (Figure 4.3(e)).

The general coverage of the rich data points $\mathcal{R}$ of the population $\mathcal{T}$ is best assessed by comparing the descriptive statistics of the four explanatory variables

and the response variables, which are available for the entire population. This information is included in the lower half of Table 4.1. As it is clear from the table, the distribution of most variables follow the same pattern within the sample and the population, except from the phishing distribution which is a lot more concentrated in the sample.

4.4 Modeling Phishing Counts

We now propose a statistical model to analyze the extent to which structural properties of hosting providers can explain the concentration of abuse in their networks, for the case of phishing abuse.

4.4.1 Regression model

Abuse is measured by counting abuse events per provider. These counts consists of non-negative integers that arise from a direct observation of a point process. In a minimal model, the underlying process is assumed to be stationary and homogeneous, with i.i.d. arrival times for abuse events and thus can be modeled with a Poisson distribution. In Section 4.4.2, we explain in more details the reason why we opted for Poisson regression over other Poisson-family models, such as Quasi-Poisson and Negative Binomial.

Let us define our response variable Y_i as the number of *abused* second-level domains hosted by provider i, for $i = 1, \ldots, n$, with n being the total number of hosting providers. Let Y_i follow a Poisson distribution with parameter $\lambda \geq 0$, The Poisson distribution has equal mean and variance $E[Y_i] = var[Y_i] = \lambda_i$. In the log-linear version of the general linear model (GLM), λ_i is modeled as:

$$\ln(\lambda_i) = \beta_0 + \boldsymbol{x}_i'\boldsymbol{\beta} = \beta_0 + \sum_{j=1}^{k} x_{ij}\beta_j, \tag{4.1}$$

where β_0 is the intercept, x_{ij}, $j = 1, \ldots, k$, are explanatory variables representing the structural properties that drive the response variable Y_i, and β_j are parameters to be estimated with maximum likelihood (ML). Statistical hypothesis tests can tell if a parameter β_j significantly differs from zero. If the null hypothesis is rejected, the corresponding explanatory variable is considered influential and the parameter's sign and magnitude can be interpreted.

4.4.2 Model goodness of fit

The fitted values produced by inserting the ML estimates $\hat{\boldsymbol{\beta}}$ into Eq. (4.1) will not match the values of the phishing data perfectly, chiefly because the data points are realizations and the fitted values are parameters of Poisson distributions. The discrepancy between the model and the data is a measure of the inadequacy of the model. Several measures exist to assess the goodness of fit of GLMs such as Log-likelihood, Akaike Information Criterion (AIC), the dispersion parameter of the Poisson model and R-squared. Here we discuss a few of them that are more specific to our Poisson model.

Over-dispersion Recall that the Poisson model assumes equal mean and variance for the response variable, that is $var\left[Y_i\right] = \phi E\left[Y_i\right] = \phi\lambda_i$, with $\phi \stackrel{!}{=} 1$, where ϕ is a dispersion parameter. However, this assumption is often "violated" in practice; that is, a likelihood function which leaves ϕ as a parameter to be estimated ($\hat{\phi}$) fits the data much better. In case of heterogeneous count variables, $\hat{\phi} > 1$ indicates signs of over-dispersion, which can be interpreted as unobserved heterogeneity in terms of a missing structural factor that leads to concentrations of observable events.

One might approach over-dispersion by starting from a Poisson model and adding a multiplicative random effect to represent unobserved heterogeneity. This leads to a Negative Binomial GLM. However, even if both parameters of the assumed Negative Binomial distribution are correctly specified, if the distribution of the response variable is not in fact the negative binomial, the maximum-likelihood estimator becomes inconsistent [148]. To make sure this holds for our data as well, we have constructed other models that control for over-dispersion, such as Quasi-Poisson and Negative Binomial models, and observed that all significant relationships stayed the same, with minor or no variation in the coefficients and minor variations in standard error values.

The literature suggests that in the absence of a precise mechanism that produces the over-dispersion, it is safe to assume $var(Y_i) = \phi\lambda_i$, for positive values of ϕ. This approach is generally considered robust since even significant deviations have only a minor effect on the fitted values, their standard errors, confidence intervals, and p-values of hypothesis tests [149]. Moreover, over-dispersion, is a sign of unobserved heterogeneity and is an indicator for structural variance in our model. Any attempt to compensate it with the choice of more complex distribution functions, such as negative binomial or zero-inflated Poisson, may hide the very signal we are looking for.

The dispersion parameter ϕ of a Poisson regression model is calculated using the chi-square statistic χ^2 divided by degrees of freedom, as it is more robust

against outliers [150].

$$\hat{\phi} = \frac{\chi^2}{(n-k-1)} = \sum_i \frac{\left(y_i - \hat{\lambda}_i\right)^2}{\hat{\lambda}_i \cdot (n-k-1)}, \tag{4.2}$$

with n being the number of observations and k the number of coefficients. $\boldsymbol{y} = (y_1, ..., y_n)'$ are the observed values of the response variable; $\hat{\lambda} = (\hat{\lambda}_1, ..., \hat{\lambda}_n)'$ are the corresponding predicted values under the fitted model, respectively.

Pseudo R-Squared A popular measure to assess the fraction of variance explained by a linear model is the R-squared statistic. Similar statistics that take values between 0 (not better than intercept-only model) and 1 (perfect fit) have been proposed for GLMs and are called *pseudo* R-Squared. According to [149], a pseudo R-Squared measure for Poisson models that takes the effect of over-dispersion into account is given by

$$R^2 = 1 - \frac{D(\boldsymbol{y}, \hat{\boldsymbol{\lambda}}) + k \cdot \hat{\phi}}{D(\boldsymbol{y}, \bar{Y})}, \tag{4.3}$$

where $D(\boldsymbol{y}, \hat{\boldsymbol{\lambda}})$ is the deviance of the fitted model, $D(\boldsymbol{y}, \bar{Y})$ is the deviance of the intercept-only model, $\hat{\phi}$ is the estimated dispersion (Eq. (4.2)), k is the number of covariates fitted, (excluding intercept) and $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

4.4.3 Model specifications

After selecting the proper regression model and discussing goodness of fit measures, we choose to fit different specifications of the model with a step-wise inclusion of the variables that capture providers' structural properties. For each of the variables, we hypothesize the direction of its relation with phishing counts.

We expect that the number of phishing counts increases as the 'Number of hosted domains' and 'Number of IP addresses hosting domains' increase. Both variables are correlated and measure some aspects of the size of a provider. We may expect that the coefficient sizes decline if both enter the model together, but it is unlikely that one of them becomes redundant. In case of 'Number of assigned IP addresses', the assumption is slightly different since the more assigned IP addresses does not necessarily mean that the provider uses them for web hosting. In contrast, we found that the business model of providers with larger assigned IP space is closer to a broadband provider who uses only a very small portion of its assigned space for web-hosting. Accordingly, since phishing attack – as an instance of abuse – directly depend on web-hosting, we expect

providers with large 'Number of assigned IP addresses', to have less phishing events in a specification where size is already controlled for with the two other variables.

In addition, note that our log-transformation of the top three variables shown in Table 4.2, perfectly matches with the log-link function of the Poisson model.

We expect that the variable 'Percentage of domains hosted on shared IP addresses' correlates positively with the phishing counts of providers due to commonly known vulnerabilities of shared hosting services [39, 122], assuming that attackers would go for targets that are easier to compromise.

Table 4.2: GLM for count of phishing domains in APWG for all the hosting providers

	Response variable: count of phishing domains				
	Poisson with log link function				
	(1)	(2)	(3)	(4)	(5)
Number of assigned IP addresses		1.186***	−1.665***	−0.728***	−0.768***
		(0.002)	(0.006)	(0.006)	(0.006)
Number of IP addresses hosting domains			3.623***	1.104***	1.570***
			(0.006)	(0.008)	(0.010)
Number of hosted domains				1.686***	1.238***
				(0.004)	(0.006)
Percentage of domains hosted on shared IP addresses					0.027***
					(0.0003)
Constant	−0.122***	−3.594***	−2.732***	−5.072***	−6.755***
	(0.005)	(0.010)	(0.011)	(0.014)	(0.024)
Observations	45,358	45,358	45,358	45,358	45,358
Log likelihood	−223,113.400	−514,546.600	−236,442.400	−117,601.700	−111,570.800
Akaike Inf. Crit.	446,228.800	1,029,097.000	472,890.800	235,211.400	223,151.700
Dispersion	2934.775	619.708	554.695	12.149	13.166
Pseudo R^2		0.221	0.648	0.831	0.841

Note: Standard errors in brackets

*p<0.05; **p<0.01; ***p<0.001
Standard errors in brackets

4.4.4 Estimation results

We construct several models by performing a step-wise inclusion of the explanatory variables explained in Section 4.4.3. A summary of these regression models is presented in Table 4.2. The table contains 5 models with estimated coefficients of explanatory variables in each model. Each coefficient demonstrates the amount of variance in the phishing counts that a variable can explain given other variables in the model.

As we move from model (2) to (5), we aim to improve our models in terms of goodness of fit metrics explained in Section 4.4.2. Likewise, the pseudo R-

squared values increase when explanatory variables are added from model (2) to (5). More specifically, with the four size and business-related variables, we are able to explain 84 % of the variance in abuse counts across 45,358 hosting providers with simple structural properties of providers. One should note that the dispersion parameter $\hat{\phi}$ across the models has been reduced from 2934.77 for the intercept-only model to 13.17 in model (5). In addition to other factors, the significance and signs of the estimated coefficient for explanatory variables do not change between model (3) and model (5), which further indicates we can safely take model (5) as our final model.

The results in model (5) clearly suggests that the number of phishing sites increases as the 'Number of hosted domains' and 'Number of IP addresses hosting domains' increases. One should note that these two variables together capture the *attack surface* of the hosting providers for the case of phishing attacks, the best. Hence, the results demonstrates that for larger attack surfaces, there are more phishing websites.

As hypothesized, 'Percentage of domains hosted on shared IP addresses' show a statistically significant relation with the abuse counts, indicating that having more domains on shared servers make a provider more exposed to phishing attacks. The 'Number of assigned IP addresses' shows a statistically significant negative relation with the abuse count, as expected when controlling for size. As pointed out earlier, a manual inspection of the providers with large 'Number of assigned IP addresses' suggests that they are mostly broadband providers who offer web hosting only as a very small part of their business. Therefore, the negative sign of 'Percentage of domains hosted on shared IP addresses' works in line with our hypothesis of having more web hosting domains and IPs as attack surface, increases the number of phishing victims.

In addition, the coefficients and pseudo R-squared values in the models (2) to (5) further confirm the point we made earlier in the introduction of the chapter that a simple bi-variate correlation or a *naive* normalization of abuse with one size metric, while neglecting other size properties, cannot comprehensively explain the variance in abuse counts.

Taking model (2) as an example, the value of estimated coefficient for 'Number of assigned IP addresses' suggest that increasing this variable by one unit (i.e., one decimal order of magnitude), multiplies the number of expected abuse counts by $e^{1.186} = 3.273$. However, a *naive* normalization of abuse (abuse counts divided by network size) would have assumed a coefficient equal to 1 for the 'Number of assigned IP addresses'. Here, our study distances itself significantly from the prior work, where just one size metric is taken into account. In the multivariate models, several size indicators offset each other, making the es-

timation more precise. In addition, in model (2) where 'Number of assigned IP addresses' is the only size variable, we are only able to explain 22% of the variance in phishing counts whereas by adding three other size metrics, the explained variance is improved up to 84%.

4.4.5 Quantitative interpretation

To illustrate the economic significance of the parameters in the fitted model (5), we familiarize the reader with three scenarios. The scenarios are based on hosting provider groups discussed in Chapter 3, which is medium, small and large hosting providers.

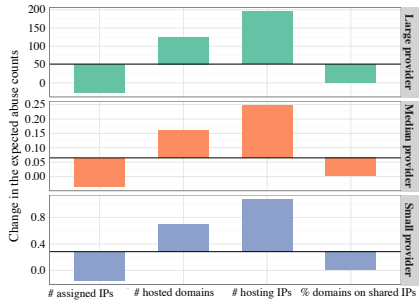


Figure 4.4: Partial effect of one unit increase in the predictors on the expected phishing counts (mind the different scales)

For each scenario, we show the partial effect of changes in the abuse counts given each of the explanatory variables (see Figure 4.4). In the first scenario (scenario 1), all explanatory variables are set at their median as a baseline situation (see Figure 4.4). In scenario 2, the baseline shows a typical small shared hosting provider with a small number of assigned IP addresses (0.47: $\approx 10^{0.47}$ IPs assigned), a small number of IP addresses used for hosting domains (0.47: $\approx 10^{0.47}$ IPs), a small number of hosted domains (1.95: $\approx 10^{1.95}$ domains) and a high percentage of domains hosted on shared IP addresses (100%). In scenario 3, the baseline situation indicates a large web hosting provider with huge number assigned IP address space (6.85: $\approx 10^{6.85}$, large IP address space used for hosting domains (5.67: $\approx 10^{5.67}$ IPs assigned), large amount of hosted domain names (5.68: $\approx 10^{5.68}$ domains) and very small percentage of domains hosted on shared IP addresses (0.48%). Apparently, this is a common case for web hosting providers that are mostly offering dedicated services [142].

The bars for each of the scenarios in Figure 4.4 illustrate the change in the expected count of abuse events for a unit change in each of the explanatory variables, while holding the others constant. Given the coefficients for the explanatory variables in Model (5) in Table 4.2, one can easily observe that changes in 'Number of IP addresses hosting domains' while other variables are constant, changes the phishing counts the most, while the effect of one unit change in 'Percentage of domains hosted on shared IP addresses' on the phishing counts is small although statistically significant.

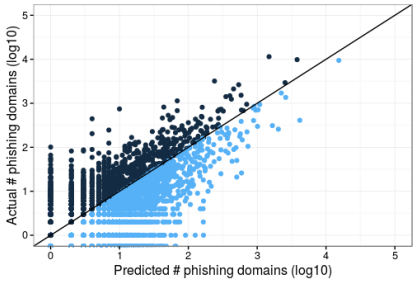


Figure 4.5: Observed and predicted number of phishing domains for Model (5) in Table 4.2

Obviously, evaluating how effective providers are in keeping phishing sites off their networks is important for developing policies and best practices. Absolute concentrations of abuse are still useful, of course. Even if they reflect structural factors, those providers could be asked to go beyond the security practices of the industry, because of their prominent position. Such a call is less plausible, however, if attackers can easily switch among the thousands of providers, as has been found in other work [48].

Rather than looking at absolute counts of abuse alone, measuring the amount of abuse relative to the providers' structural properties adds valuable information. Figure 4.5 visualizes the difference between actual phishing counts and the counts predicted by our model. Providers below the line are performing better than average and the distance tells us by how much.

4.5 Additional Provider Structural Properties

In model (5) of Table 4.2, we see that around 84% of the variance in phishing counts is explained by a number of the structural properties of providers, namely,

four variables related to size and business model of the hosting providers.

In this section we continue with interpreting regression models, this time on a sample of the rich data points (set $\mathcal{R}$) as described in Section 4.3.3. Recall that we need to adjust the estimation method by introducing two sets of fixed-effects, (i) for level differences between statistical twins and (ii) for level differences between countries. Fixed effects take away known linear dependence in the residuals. This is essential to obtain accurate test statistics (which assume independent residuals). Sources of dependence are within twins due to the selection strategy and within countries due to the inclusion of country-level variables.

We define the model with (one set of) fixed-effects as:

$$\ln(\lambda_{ij}) = \beta x_{ij} + \cdots + \delta_i, \tag{4.4}$$

where β is the estimated coefficient for x_{ij}, x_{ij} are explanatory variables collected for hosting providers in set $\mathcal{R}$ and δ_i is the "fixed-effect" parameter [151]. Subscript i refers to different twins and $j \in \{1, 2\}$ refers to different measurements within each twin, i.e., the same variable measured at different hosting providers belonging to the same twin.

The model uses the variables explained in Section 4.3.3 as predictors. We add two fixed-effects to the model – twin and country – by fitting a separate dummy variable as a predictor for each class of each variable. The twin fixed effect controls for the bias introduced by selecting similar samples. The country fixed effect prevents undue dependence in the residuals if country-level predictors are included. In addition, in case of missing values per explanatory variable, we perform a list-wise exclusion on a twin, if one of providers is missing. This further reduces the number of pairs per model depending on missing values of the included variables.

With fixed effects, the definition of pseudo R-squared requires some reflection. It is possible to use an intercept-only baseline, which results in artificially high pseudo R-squared values because the level differences of the fixed effects are counted as "explained". A more conservative measure is to use the fixed-effects-only model as baseline. Table 4.3 shows the summary of our results.

Table 4.3 contains two baseline models (model (1) and (2)). While the former only models twins as a fixed-effect, the later models both twins and countries as fixed effects. Model (3) broadens model (2) by fitting more explanatory variables with fewer missing values. In model (4) we add all the explanatory variables collected for the set of rich datapoints ($\mathcal{R}$), except for the ICT development index, having only twins as a fixed effect. In addition to the variables

Table 4.3: GLM for count phishing domains in APWG for the "statistical twins"

	Response Variable: count of phishing domains				
	Poisson with log link function				
	(1)	(2)	(3)	(4)	(5)
Price per year				0.0003	-0.007^{***}
				(0.0002)	(0.001)
Popularity index (in thousands)			0.001^{***}	0.02^{***}	0.1^{***}
			(0.000)	(0.002)	(0.01)
Time in business			-0.017^{*}	-0.059^{***}	0.015
			(0.007)	(0.005)	(0.012)
ICT dev. index			0.951^{***}		-165.065
			(0.214)		($> 10^3$)
Wordpress use				5.858^{***}	2.203^{***}
				(0.271)	(0.450)
Twin fixed-effects	Yes	Yes	Yes	Yes	Yes
Country fixed-effects	No	Yes	Yes	No	Yes
Observations	210	210	180	84	82
Log likelihood	$-2{,}783.157$	$-1{,}111.825$	-966.249	-795.838	-249.780
Akaike Inf. Crit.	5,776.315	2,521.650	2,192.499	1,683.677	641.560
Dispersion	40.133	25.770	27.352	31.554	11.243
Pseudo R^2			-0.055	0.625	0.772
Total pseudo R^2	0.974	0.991	0.992	0.966	0.995
Note: Standard errors in brackets					*p<0.05; **p<0.01; ***p<0.001
					Standard errors in brackets

in model (4), in model (5) we add ICT development index, setting both twin and country as fixed effects.

The regression results in Table 4.3 indicate that by including both twins and country as fixed effects, keeping in mind that the datapoints in set $\mathcal{R}$ are already homogeneous in terms of the other variables, we are able to explain around 77% of the variance in phishing counts for the twins (pseudo R-squared value). Note that this variance is the 77% of the 16% unexplained variance that is remained in model (5) of Table 4.2, for the full population of providers. The results further highlight the importance of accounting for other –non-size related– structural properties of providers, while explaining the variance in concentration of abuse.

Even though we have around hundred fixed-effects, we still get very clear and significant results for the main effects. In addition, by moving from model (1) to

model (5) in Table 4.3, we are reducing the amount of unexplained heterogeneity (model's dispersion) from 40.133 to 11.243.

As hypothesized, we see a significant negative relation between the price of hosting and phishing counts in model (5). That is, if we increase price by one unit holding the other variables constant, the phishing count will be multiplied by $e^{(-0.007)} = 0.99$ which means that the cheaper a service is, the more the hosting provider is prone to phishing attacks. Interestingly, variable 'Price per year' shows a different relation in model (4) where we do not control for cross-country differences. This change is expected, however, as properties of hosting markets in different countries can differ a lot, which eventually influences the cost structure of the country in regards to hosting services. In addition, the cost of a hosting plan is relative to the economy of the provider's country. Conversion of prices in a specific country to USD, if not controlled for the country differences, can be very crude. The variables 'Wordpress use' and 'Popularity index' also show a significant positive relation with phishing count indicating that more Wordpress sites per provider, as a proxy for providers that host oft-attacked software, translates to more phishing attacks, which is quite logical considering the fact that the majority of phishing sites are on compromised machines. One unit increase in 'Wordpress use' while holding the other variables constant, multiplies the phishing counts by $e^{(2.203)} = 9.052$. Similarly, the more popular websites a provider hosts (popularity index), the more that provider is a victim of phishing attacks.

For 'ICT development index', we observe both a sign and significance change from model (3) to model (5). This can be understood by looking back at the distribution of twins in Fig. 4.3(b), where the gray color marks the twins that were excluded from Model (5) due to missing values. From the figure it is visible that the 100 observations that were excluded because of missing price information are clustered among lower developed countries, thereby removing the variance needed to find the effect of ICT development. The effect is also easily observable in Model (3). Without the price variable, 'ICT development index' shows a significant and positive relation.

Now the question is, to what extent does our sample reflect the population? Looking back to our sampling strategy, in model (5) of Table 4.2, we have a model that explained 84% of heterogeneity; so looking for neighbors in the projection of model (5) increases the chance of getting twins that are very comparable for all the factors that we cannot observe and are already somewhat correlated to size measures. This means that the enriched data points contain more valuable information than others from the total population. However, since instead of a totally random sample, we are creating twins of providers,

our targeted sampling strategy might introduce possible biases. In order to deal with that bias, we make an assumption that the bias is linear in the modeling domain, i.e., can be captured by linear fixed effects parameters. In Section 4.6, we further perform additional cross-validations, to check for possible biases our methodology might have introduced.

4.6 Robustness Checks

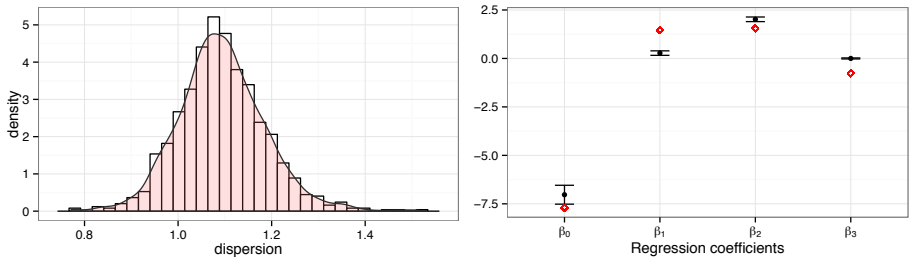
During the course of our analysis, we pointed out a few assumptions that we have made, most notably about the impact of the deviations from the Poisson model due to over-dispersion; and about the impact of measurement errors in the abuse data. In this section, we address these two concerns. Regarding the first assumption, we use a simulation study to perform a robustness check on our size estimates. To check the robustness of our method against possible errors in the our phishing blacklists data, we cross validate our results with another, largely independent data source.

4.6.1 Size estimates

Assume attack events **are** Poisson and the only structural factor that affects them is the size, i.e., the attack surface. In the absence of perfect information of the true attack surface, it can only be approximated in practice through the size variables we used in Table 4.2. Now we would like to study: to which extent are deviations from Poisson observable only by using the imperfect size estimations? The precise steps towards estimating new models for phishing abuse counts using imperfect size estimations are as follows:

We generate a *true size* variable that is following normal distribution using mean and standard deviation of variable 'Number of hosted domains'. We then simulate attack events –*hits*– where the phishing counts follow a Poisson distribution ($Pois(\lambda_{sim})$). where λ_{sim} is the mean number of phishing domains. We then build the simulated size variables used in model (3) of Table 4.2, namely, 'Number of hosted domains', 'Number of IP addresses hosting domains', 'Number of the assigned IP addresses' by adding random noise ($N(\mu_{f_i}, \sigma_{f_i}), \forall i \in \{1..3\}$) to our true size variable. μ_{f_i} and σ_{f_i} are estimated using the mean and standard deviation of the corresponding explanatory size variables.

We generate 1,000 times the synthetic data representing both the size and the dependent variables, and model them using the GLM regression specified in



(a) (b)

Figure 4.6: (a) Density of dispersion estimates obtained from synthetic Poisson phishing arrival with imperfect size control and (b) Error bars of the simulated regression coefficients

model (4) (see Table 4.2). Then we calculate the dispersion parameter for each one of the simulated models using Eq.(4.2). Figure 4.6(a) shows the density distribution of the dispersion parameter for each of the Poisson models using simulated size measures. The model indicates that the dispersion parameter is on average greater than 1. The dispersion parameter value is far from the dispersion parameter that we obtained using our real dataset in model (4) of Table 4.2. This however is expected since the real size estimates are far from being perfect and contain several outliers. Moreover, the over-dispersion in simulated size variables indicates that some part of the over-dispersion in model (4) of Table 4.2 – probably not everything – can be attributed to the approximate measurement of the size estimates. Finally, Figure 4.6(b) displays the coefficients of 1,000 model fits as error bars. The red diamonds are the coefficients obtained with the full Poisson model. The coefficients follow the same trend as in the model given the over-dispersion, which validates its robustness.

4.6.2 Phishing data

Limitations of abuse blacklists, such as comprehensiveness and independence, have been studied at length in prior work [e.g., Metcalf and Spring 2013; 2015]. In order to study the effect of such limitations on our results, we applied our approach to an alternative dataset: the Cyscon phishing data. These in APWG and Cyscon data have a 13% overlap of unique second-level phishing domains.

This provides enough independent observations to corroborate our approach and asses the sensitivity of our results to measurement errors in phishing blacklists.

Table 4.4: GLM for count of phishing domains in Cyscon for all providers

	Response Variable: Count of phishing domains	
	Poisson-Log Link Function	
	(1)	(2)
Number of assigned IPs	-0.719^{***}	-0.776^{***}
	(0.011)	(0.012)
Number of IPs hosting domains	1.170^{***}	1.751^{***}
	(0.014)	(0.018)
Number of hosted domains	1.663^{***}	1.115^{***}
	(0.007)	(0.011)
Percentage of domains hosted on shared IPs		0.033^{***}
		(0.001)
Constant	-6.432^{***}	-8.488^{***}
	(0.026)	(0.045)
Observations	45,358	45,358
Log Likelihood	−49,763.500	−47,208.470
Akaike Inf. Crit.	99,535.010	94,426.950
Dispersion	20.153	17.444
Pseudo R^2	0.791	0.803
Note:		*p<0.1; **p<0.05; ***p<0.01
		Standard errors in brackets

Similar to before, we model the arrival rate of phishing counts using a Poisson GLM with log-link function with size and business model predictors as used in model (4) and model (5) of Table 4.2. The result of the two final models is displayed in Table 4.4. We then model the statistical twins in the set of rich datapoints $\mathcal{R}$, having two sets of fixed effects for differences between twins and countries. The results are shown in Table 4.5. Model (1) and (2) contain the same explanatory variables as Model (4) and (5) of Table 4.3. Reassuringly, the resulting estimated coefficients and significance levels for both of the analyses are quite similar to those of the model with APWG data.

Table 4.5: GLM for count of phishing domains in Cyscon data for the "statistical twins"

	Response Variable: Count of phishing domains	
	Poisson with Log Link Function	
	(1)	(2)
Price per year	0.0003	-0.012^{***}
	(0.0003)	(0.002)
Popularity index (in thousands)	0.02^{***}	0.1^{***}
	(0.004)	(0.01)
Time in business	-0.048^{***}	0.004
	(0.010)	(0.037)
ICT dev. index		13.610
		($> 10^3$)
Wordpress use	7.848^{***}	3.079^{**}
	(0.583)	(1.125)
Pair Fixed-Effect	Yes	Yes
Country Fixed-Effect	No	Yes
Observations	84	82
Log Likelihood	−476.818	−145.676
Akaike Inf. Crit.	1,045.635	433.352
Dispersion	20.712	2.993
Fixed-effects Pseudo R^2	0.538	0.889
Total Pseudo R^2	0.970	0.987
Note:		*p<0.05; **p<0.01; ***p<0.001
		Standard errors in brackets

4.7 Related Work

There is a large body of work on methods for detecting abused resources on the Internet. Observational data on abuse incidents is the starting point for our study, but we do not engage with the detection methods themselves and therefore will not survey them here.

Blacklists as a source of observational data on abused resources have already been extensively studied [80, 81, 82]. Closer to our research is the line of work that identifies and explains patterns in abuse data. The patterns are studied at different levels of analysis: (i) individual resource (host, IP address, domain);

(ii) network, or other aggregates of individual resources; (iii) actor, meaning the economic entity providing the resources or otherwise responsible for them; and (iv) country. We briefly survey relevant work at each level.

Individual resources. A variety of studies have been successful at explaining or predicting the occurrence of abuse, such as compromised websites, from properties of individual resources, such as content-specific features, webserver software versions, or website misconfigurations [43, 35, 44]. The factors identified in these studies impact the distribution of abuse on domain names, e.g., domains with outdated software are abused more than others. Others used DNS characteristics to predict whether domain names are malicious [47, 30]. More tailored towards phishing, some authors propose to detect phishing websites using URL and content-specific features [45, 46].

Networks. Another strand of work looks at how abuse is distributed across aggregate units of resources, such as address blocks, Autonomous Systems [48, 114] or top-level domains (TLDs) [49]. These studies often identify concentrations of abuse in certain networks [152, 153, 48] and then identify network features that correlate with abuse rates, such as poor network hygiene [90, 34] or rapidly changing network connectivity [115, 32, 33]. These studies aim to detect malicious or poorly managed networks, rather than disentangling the factors explaining the causal relationship of why abuse is concentrated or how it is distributed across all networks. Furthermore, to be useful for policy and interventions, the aggregated resources need to be attributed to the relevant economic actors rather than a technical entity. For drawing inferences on providers, an explicit attribution method is needed. This takes us to the third level of analysis.

Actors. Actors are the economic entities that operate resources or are otherwise responsible for them. Work at this level has to bridge the gap between the technical identifiers in abuse data, like ASes, and the organizations responsible for specific resources. This is not straightforward as many Internet Service Providers (ISPs), for example, operate multiple ASes [121], and many hosting providers share an AS with other providers [142].

Our work is situated at this level. We are not aware of any work that explains abuse patterns across hosting providers. Liu et al. have studied the extent to which organizations' properties, such as symptoms of mismanagement, size of allocated IP space, and corresponding abuse counts can predict data breach incidents. This work is amongst the first studies that predicted incident rates from one structural property of the organizations (i.e., the number of IP addresses allocated) and several effort-related indicators (i.e., indicators of network mismanagement and misconfiguration). However, it does not distinguish

between organizations that offer hosting services and other organizations. It also does not model the structural properties of organizations comprehensively [35]. Somewhat related is a study on security practices of a small sample of hosting providers [10]. More mature work exists for domain names. Studies have identified which registrar or registries are associated with malicious domains, and quantified the effect of different interventions on abuse rates [50, 51].

Our study extends the prior work in a variety of ways. Unlike the existing work, we are not trying to find technical features to correlate with abuse, but we are trying to understand to what extent abuse levels are a function of structural properties of the industry versus being determined by the security efforts of individual providers. For that, first we adopt a better attribution method for identifying the relevant actors, by moving from ASes towards hosting providers, to whom the IP space is allocated. We fit a multivariate statistical model and include a set of important explanatory factors, such as size of IP space, which have been explored before [35]. Other properties of hosting providers, such as size of domain name space, size of IP space used for web hosting, portion of shared hosting business, hosting price, time in business, are studied for the first time here.

Countries. The highest level of analysis is countries. Work in this area studies the relationship between country-level factors, such as GDP, rule of law and ICT development, and the distribution of abuse, most notably infected hosts [89, 28, 154]. In contrast to this research, we take providers as the unit of analysis, because that is where agency is located in terms of fighting abuse. That being said, country-level factors describe institutional differences in the environments of providers that are also relevant to take into account. In our study, we estimate the impact of the ICT development of a country on abuse, while controlling for other unobserved country-level differences using fixed effects models.

4.8 Conclusions and Discussions

The core question of this chapter is: to what extent are abuse levels determined by the structural properties of providers versus being mainly determined by other factors including, but not limited to, the security efforts of individual providers? Below we summarize our findings and discuss the implications of the results.

We reduced errors in the attribution of abuse by empirically studying the population of hosting providers, which are defined based on organizations to

whom IP address space is assigned, rather than by routing data and AS ownership. Next, we advanced the existing work that uses simple regression analysis and naive normalizations by studying a variety of factors and errors at work that can potentially explain abuse counts. By building several GLM models for phishing abuse counts as the response variable, we demonstrated that a handful of providers' structural properties–such as the number of domain names, number of IP addresses used for web-hosting, and the size of their shared hosting business–can already account for 84% of the variation in phishing counts. These variables are easily measurable at scale and capture the 'attack surface' of providers along with aspects of their 'business model'.

Additionally, we measured the impact of previously unstudied factors, such as price, time-in-business of a provider, and the amount of Wordpress sites per provider. These were collected via a tailored sampling approach and explained a further 77% of the remaining variation in phishing abuse.

Finally, we performed a set of robustness checks on the assumption we made during our analysis. The results of the simulation study performed to check the robustness of our size estimates indicated that coefficients of size variables follow the same trend as in our model (Table II) given the over-dispersion. To check robustness of our method against observational biases in the phishing data, we cross-validated our results from APWG blacklist data against Cycson phishing blacklist data and observed very similar results.

Our findings suggest that abuse rates for phishing reflect an overall bad harvest, rather than being driven by some rotten apples, i.e., providers that don't care about security. In other words, referring back to the explained and unexplained variance by our models, we observe that structural properties of providers explain the majority of variance in phishing abuse counts, leaving a thin margin for other unmeasured factors including, but not limited to, the security efforts of providers. When structural factors are so dominant in driving abuse, it undermines the common narrative to call for better security practices of apparently under-performing actors or for even more intrusive interventions, such as sanctions. However, our findings do not limit the action space for policy. Quite the contrary: data-driven policy could try to improve the factors identified as influential, e. g., require higher security standards at providers who host more popular websites.

Our approach enhances more informative comparisons of provider's security performance. In other words, it generates comparative abuse metrics by controlling for the structural differences among providers. Such relative metrics are more suited to evaluate countermeasures than absolute counts or relative counts that generated by naive normalization of one size estimate. Additionally,

relative metrics can, in themselves, incentivize better security [33, 155]. In sum: throwing out a few rotten apples might appear more tractable, but producing a better harvest is definitely possible.

Here, we should also acknowledge several limitations of our work. First, our method is geared towards identifying the main explanatory factors in the population of hosting providers. Our conclusions should not be misinterpreted as evidence that there are no misbehaving or negligent hosting providers, only that their impact on the population of phishing incidents is surprisingly limited.

Second, the presence of certain unobserved factors, including security effort and attacker behavior, is a limitation of this work. We have reduced the likelihood of these being major factors in the abuse patterns, as witnessed by the variance explained by the structural properties. We are able to explain 84% of the variance by the structural factors alone – and even more when we take the findings of the statistical twins sample into account. The remaining unexplained variance, which is the combination of provider's security effort, attacker behavior and measurement errors, suggests that the impact of provider's security effort should be limited. That being said, the only way to determine the precise impact of security effort of providers on abuse levels is by directly measuring it. Following this limitation, in chapter 6 of this dissertation, we study possible patterns in attacker behavior for the case of malware used to attack financial institutions. In chapter 7, we measure security effort as a latent variable. More details will be discussed in the relevant chapter.

Third, certain structural factors might indirectly capture some information about security efforts. One could argue, for example, that the pricing model chosen by a provider might also contain a signal about the amount of resources available for security. On the other hand, the fact that 84% of the variance is explained by purely technical structural properties, unrelated to price, suggests that also this impact is limited. Only a more direct observation of security effort can establish how it is related to price.

A final limitation is that our empirical evidence is specific to phishing. Our modeling approach is agnostic to the type abuse, however. The independent variables and model design are not specific to phishing. Future work can use our approach to identify the impact of these structural and business model factors on other types of abuse in the hosting market. For some sources, like drive-by-download sites, we expect similar patterns. For other, more idiosyncratic types of abuse, like long-living botnet C&C servers, we might expect different patterns. There, we might indeed find that rotten apples drive the abuse rates, rather than a bad overall harvest. In order to test our hypothesis we carried out an analysis of abuse concentrations for botnet C&C servers in the next chapter.

CHAPTER 5

Measuring the Impact of Providers' Reactive Security Efforts on Abuse

The analytical model in the previous chapter identified structural provider properties and security effort as the two main causal factors affecting the performance of providers. The subsequent empirical analysis also showed that over 85% of the variance in the number of phishing sites can be explained from the providers' exposure to such attacks, as measured by structural properties. This leaves remarkably little room for the impact of the actual security efforts of providers. In this chapter, we apply the same methodology used in the previous chapter, and apply it to a different type of abuse, namely botnet command&control servers. In the first parts of this chapter, we study the effect of the structural properties of providers on the number of C&C domains hosted. In the second part of this study, we examine the impact of providers' take-down reactive security efforts on the occurrence of C&C abuse.

5.1 Introduction

Research into the disruption of botnets has mainly focused on two strategies: comprehensive take-down efforts of the command and control (C&C) infrastructure and the cleanup process of the infected end user machines (bots) [53, 156, 157]. The first strategy has the promise of being the most effective, taking away control of the botnet from the botmasters. In reality, however, this is often not possible. The second strategy is not about striking a fatal blow, but about the war of attrition to remove malware, one machine at a time. It has not been without success, however. Infection levels have been stable in many countries [158].

In practice, a third strategy is also being pursued. Similar to access providers

cleaning up end user machines, there is a persistent effort by hosting providers to take down C&C servers, one at a time. This line of mitigation has been studied much less, perhaps because most botnets have been resilient to these efforts.

Could this strategy be made more effective? This depends on how attackers distribute their C&C domains. Do they randomly distribute them over many hosting providers? Or do they locate them predominantly in carefully selected providers, perhaps those who are negligent in terms of abuse handling or who offer bulletproof services to actively support criminal activities [33, 137, 36]?Depending on the answer, there are different directions for improving mitigation.

This chapter sets out to discover the strategies of attackers for the placement of their C&C servers across the hosting market. We focus on botnet families that have, in varying degrees, been used to attack financial services. Well-known examples are Zeus, Citadel and Dyre. These are widely understood to be among the most harmful botnets. The industry association M3AAWG has listed them as a top priority for abuse handling by providers [9]. This means that if providers do anything in terms of mitigation, it would be most visible for these botnet families. Put differently, if attackers care about the security practices of providers, we should see it first and foremost in the location of the C&C for these botnets.

Do attackers prefer providers with little or no abuse handling? Or are the C&C domains more or less randomly distributed across the overall attack surface of the hosting market? We study seven years of data on the location of C&Cs for 26 botnet families engaged in attacks against financial services.

We model the distribution of C&C domains across the overall landscape of the hosting market. Using several datasets for approximating the size and attack surface of providers, we can quantify the extent to which the number of C&C domains per provider can be explained as the outcome of a random selection process by attackers. We then analyze whether there is a relation between the concentration of C&C in providers and the speed with which providers take down such domains.

Our contributions are as follows:

- We track the trends in the hosting locations of C&C for 26 different malware families that, to varying degrees, have been used in attacks on financial services. We find that, over time, C&Cs domains are spread out over more providers, diluting the concentrations of C&C;
- We model the distribution of C&Cs across providers and show that the

mere size of the provider can explain around 71% of the variance in the number of C&Cs per provider, whereas the rule of law in the country only explains around 1%, suggesting a predominantly random selection process by the attackers for locating their C&C;

- Using a sample of hosting providers, we show that business model characteristics – such as pricing, popularity, time in business and the ratio of WordPress websites – all have a significant impact on the concentration of C&C domains;
- We demonstrate that there are statistically significant differences among providers in C&C take-down speed. Despite such differences, the take-down speed only has a weak relation with the concentration of C&Cs across providers, suggesting that attackers have little or no preference for hosting their domains in hosting providers that allow longer C&C uptime;

The remainder of this chapter is organized as follows: Section 5.2 describes our data collection methodology. Section 5.3 provides a descriptive summary of our datasets and studies the concentrations of C&Cs in terms of malware and hosting types and across different geo-locations. Section 5.4 outlines a set of variables that capture different aspects of provider s' characteristics and next use them to model the C&C concentrations across providers. In this section we discuss our modeling approach and results at length. We then extend our model in Section 5.5 with taking the effect of provider take-down speed of C&C domains into account. Our finding are compared to the related work in Section 5.6. Finally, we discuss the main conclusions and limitations of our work in Section 5.7.

5.2 Data Collection Methodology

To understand the attacker's strategy for the placement of their C&C servers across the hosting market, we employ two types of datasets: (i) data on C&C domains; and (ii) data on hosting providers. We first provide an overview of these datasets.

5.2.1 Command-and-Control Data

As stated earlier, we focus on C&Cs of botnets engaged, to varying degrees, in attacks on financial services. We make use of two datasets which in conjunction provide information on C&C domains located in 109 countries:

ZeusTracker: Provided by Roman Huessy from Zeus-Tracker [77], is a C&C panel tracker that contains meta data on C&C servers online at any point of time between 2009 and 2016 for the Zeus malware family.

Private honeypots: Captured by a security company specialized in threat intelligence for banks and financial institutions using honeypots located all over the world, this dataset contains a list of botnet C&C domains from various botnets. Some of those botnets are predominantly used for attacks on financial services, like Citadel. Others are more generic malware families, but the security company has observed them as participating in attacks on financial services. The data is collected over a period of one year (2015Q1-2016Q1) using two methods: by running live malware samples and using honeypots.

The combined dataset contains 11,544 unique domain names associated with 8,528 IP addresses. A more detailed summary of our C&C data is shown in Table 5.1.

Table 5.1: C&C data summary

Year	# Domains	IP addresses	Families
2009	934	771	1
2010	1016	806	1
2011	1071	638	1
2012	1189	922	4
2013	1761	1365	3
2014	2188	1768	4
2015	3897	1819	28
2016	3718	969	34

5.2.2 Hosting Provider Data

The next steps towards studying the location of C&Cs is to attribute them to their responsible service providers. To that end, we need to reliably identify hosting providers. We use methodology and data introduced in more details in Chapter 3 and further improved in Chapter 4. The final set consists of 45,358 hosting providers, representing the population of hosting services from all over the world.

5.3 Characterizing C&C Concentrations

Given our C&C and hosting provider datasets, we can examine the distribution of C&C domains across different hosting providers to gain insight into attacker C&C placement strategies. Do they prefer certain hosting providers? Do they prefer certain locations? In this section, we provide a descriptive summary of our data and examine such different aspects of C&C concentration.

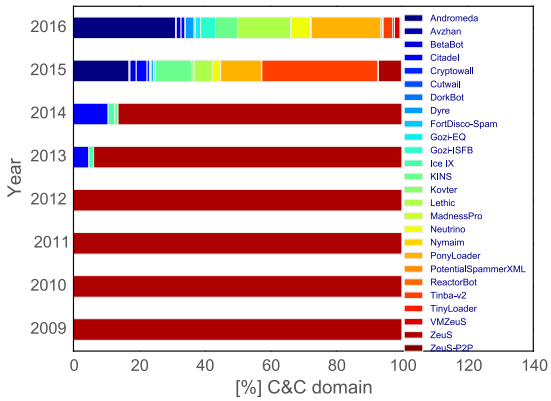


Figure 5.1: Distribution of malware types over years

5.3.1 Descriptive Summary of C&C Domains

Figure 5.1 displays the distribution and evolution of the financial malware families over years, given the first time a malware is seen in our data. The trend indicates the presence of Zeus as the main financial malware between 2009 and 2012. Starting from 2012, we observe the emergence of ZeuS-related families such as Citadel and Ice-IX and gradually other malware families such as Dyre, Cryptowall and Avzhan.

The portion of our C&C data that comes from ZeusTracker also includes information on the type of hosting for some of the C&C domains. The information about the hosting type is gathered by ZeusTracker based on manual analysis of a sample of C&Cs.

Figure 5.2 shows the distribution of these types over the measurement period. Since the hosting types are known only for a minority of the domains, it is not easy to make any substantive conclusions from the exact numbers. However,

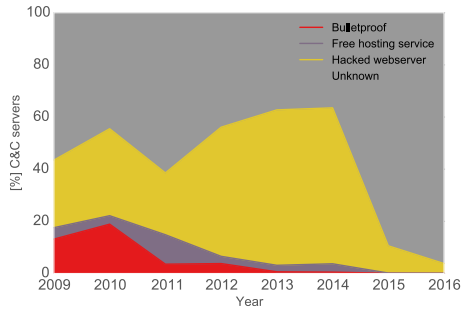


Figure 5.2: Distribution of malware hosting types over years

the plot suggests that the majority of C&Cs with known types are located on compromised servers, followed by a minority located at free or bulletproof hosting providers. This further highlights the importance of measures taken by providers to protect the machines they are hosting from getting compromised.

5.3.2 Concentration of C&Cs across Providers

Next, we examine the trends in concentration of C&C domains across providers, to examine if C&C domains are mostly concentrated in specific hosting providers. This could help us to gain a better understanding of attacker preferences.

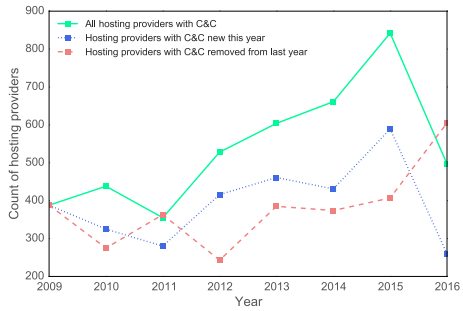


Figure 5.3: Time-series plot of providers hosting C&Cs

Figure 5.3 depicts the number of providers hosting C&C domains over time. The green line indicates the total number of providers hosting C&C domains in a given year. The blue line indicates the amount of newly observed providers

hosting C&C domains for a specific year while the red line depicts providers that were no longer hosting C&Cs in comparison to the previous year. It should be noted that the removal of a hosting provider is not necessarily due to clean-up efforts, but could be the consequence of attackers' choices. The plot gives a better sense of the total number of hosting providers that are linked with hosting C&C domains.

Over time, we observe a general increase in the total number of providers. At the same time, the number of newly added and removed providers follow a similar upward trend which points to a relatively high entrance and exit rate of providers. The pattern also indicates that an attacker's choice of provider is highly dynamic and shifts from provider to provider over time.

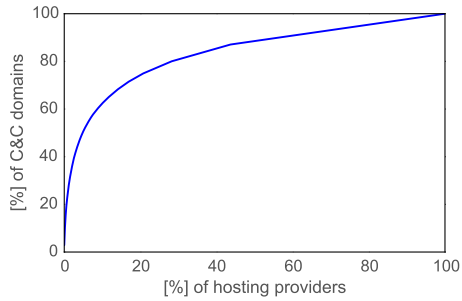


Figure 5.4: Cumulative percentage of C&C domains for the percentage of hosting providers

Figure 5.4 displays the cumulative percentage of C&C domains against the percentage of hosting providers. The blue line in the plot follows a power-law distribution: a large number of C&C domains are concentrated in a small number of hosting providers, 80% of C&Cs are located in less than 30% of the hosting providers. This shows a clear concentration of C&C infrastructure. While the majority of C&Cs are hosted by a minority of providers, it is still unclear whether this concentration in caused by an attacker's preference to choose lax hosting providers in terms of security, or whether it is just an artifact of a provider's size and business model and therefore is randomly distributed. We further examine this question via modeling various provider characteristics in section 5.4.

5.3.3 Geography of Providers Hosting C&C Domains

We also examine the geographical distribution of the C&Cs and the providers who host them. Hosting providers operate from various jurisdictions and therefore specific geographical parts of their business could be prone to more abuse due to factors such as weak rule of law or enforcement institutions.

We map the C&C servers to their geo-location using the MaxMind GeoIP API [120]. While the C&Cs in our data are located in 109 various countries around the globe, figure 5.5 suggests that the majority of C&C domains in the top-20 most abused hosting providers are located in US and western Europe. There are a few exceptions such as `Confluence Networks` that seem to operate in part from the Virgin Islands and `SoftLayer Technologies` that hosts domains in Panama.

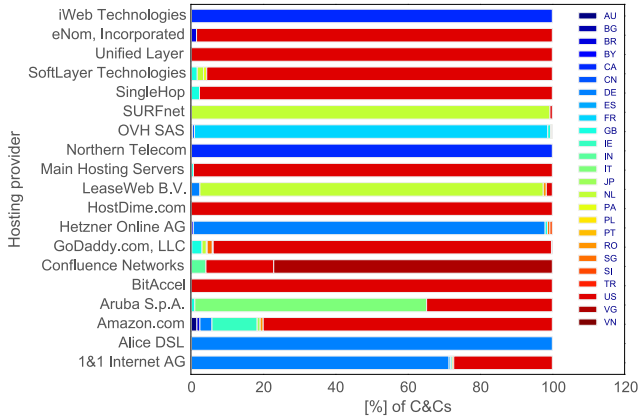


Figure 5.5: Geo-location of C&C domains for the top-20 providers hosting C&Cs

5.4 Statistical Model of C&C Concentrations

As we explained earlier, we aim to have a better understanding of why C&C domains are concentrated in certain providers through building a statistical model that explains C&C counts from provider characteristics. In Chapter 4, we proposed an approach to study phishing abuse counts across hosting providers using regression models that carefully decomposes different sources of variance in abuse counts for different characteristics [159]. Our current goal is to see whether

we see similar patterns in attacker preferences for hosting C&C infrastructure. Contrary to phishing sites, one might expect C&C to be more selectively located.

We define a set of explanatory variables that capture structural characteristics of providers and their security effort, as defined in the previous chapter [159]. In this section we study the relation between C&C abuse and structural characteristics of providers. In the next section, we examine the 'average C&C uptime' as a proxy for the security effort of providers. We categorize the variables characterizing structural properties of providers into those that capture *size*, *regulatory* aspects of the country in which providers operate and those that capture providers' *business model* characteristics. A summary of these variables is provided in Table 5.2.

5.4.1 Structural Characteristics of Providers

Size

Allocated IP space is the size of the IP address netblock(s) assigned to a hosting provider according to WHOIS data provided by Regional Internet Registries (RIR). We use this information as an indicator of the attack surface of a provider, assuming that the address range is a proxy for the amount of server infrastructure the provider is operating and that any machine in that infrastructure has a certain probability to be abused by miscreants – i.e., more servers means a higher count of C&C. This variable ranges from one IP address to many thousands of allocated addresses, suggesting a large heterogeneity in the market for hosting services, in terms of attack surface but also business models of providers.

Table 5.2: Descriptive summary of variables in our model

variables	n	min	mean	median	max	sd
Allocated IP space size (log_{10})	45,363	0	3.08	3.19	8.35	1.16
Webhosting IP space size (log_{10})	45,363	0	1.78	1.66	6.24	0.76
Domain name space size (log_{10})	45,363	0	1.98	1.83	7.64	0.88
Portion of shared hosting (%)	45,363	0	50.99	58.99	100	37.13
Rule of law	46,269	−1.89	1.05	1.62	2.12	0.95
Best price (USD)	235	0	20.89	6.95	419	47.34
Popularity index	90	0	8,328.34	1,279.29	187,454.30	26,828.03
Time in business (years)	150	2	14.01	14.19	30	4.43
Vulnerable software ratio	86	0.01	0.19	0.16	0.48	0.11

Webhosting IP space is the number of IP addresses hosting a domain name. To collect information on this variable we make use of passive DNS data. We

calculate this variable by summing up all the IP addresses associated with domains per provider that have been observed in our DNSDB passive DNS data. The combination of the allocated IP space and web hosting IP space not only indicate the size of a provider's infrastructure, but also reflect the kind of business a hosting provider is running. For instance, providers who use a large part of their allocated IP space for hosting domain names have a business model more focused on web hosting and are different from providers who use their allocated IP space for other services such as providing virtual private servers (VPS), collocation, or access services.

Domain name space is the number of domains hosted by a particular provider. Again we use passive DNS data to collect information on this variable. It is calculated by summing up the number of second-level domains hosted on the IP addresses of provider in the passive DNS. Note that due to the large variance and skewed distribution of the first three variables, we use a log-transformation of these variables (Log_{10}).

Proportion of shared hosting measures the ratio of domains that are hosted on shared IP addresses divided by the total size of domain name space. As defined in Chapter 3, an IP address is considered shared if it hosts more than 10 domain names [142, 44]. This variable not only conveys information about the size of the shared hosting infrastructure of a provider, but also about the provider's business model, i.e., the degree to which a provider's business relies on low-cost shared hosting services.

Regulation

Rule of law is an index that we use as a proxy for law enforcement against illegal activity within a country. It is a well-established indicator relying on a large number of periodic surveys to measure how the rule of law is experienced in practical, everyday situations by the general public. The index is provided by the World Justice Project, a non-profit organization working to advance the rule of law around the globe and is based on indicators such as constraints on government powers, absence of corruption, order and security, civil and criminal justice, open government, fundamental rights, regulatory enforcement and justice experienced by ordinary people from 99 countries around the globe [160]. Lower index values represent a stronger rule of law.

Business Model

Most of the business model variables in this section cannot easily be collected at scale for the total population of hosting providers. While collecting price in-

formation requires manual inspection of the provider's webpage, collecting some other variables at scale such as vulnerable software can be very time consuming. Therefore, we collect information for variables in the business model category for only a sample of the providers.

Popularity index proxies the online popularity of a hosting provider. We use Alexa's one million top-ranked domains to calculate the popularity index. We assume a provider is more popular when more top-1M domains are on the list of domains that it hosts and speculate that more popular providers are exploited more often for setting up C&C domains. In order to reduce the bias towards the very large hosting providers, the index is calculated by summing up the base-10 logarithm of the reverse Alexa rank of all domains. The score communicates information about both website popularity (i.e., customers) and the density of popular domains in a hosting provider.

Time in business is a proxy for capturing the extent to which a provider can be exploited, given the amount of years it is operating in the hosting business. The expectation is for more experienced providers to be exploited less due to learning effects. The data for this variable is collected by querying the WHOIS databse for the registration date of the provider's website. We have cross-checked the results with the Internet Archive database [146] for all data points. Almost all domains in our sample were captured by Web-archive a couple of months after they were registered.

Best Price is basically the least expensive hosting plan on offer by the hosting provider. Our hypothesis is that providers with less expensive hosting plans are more popular to host C&C domains, not only for the case of malicious registrations but also in the case of compromised domains. The intuition being that providers with cheaper plans most probably dedicate less resources to the security of their services. All prices are converted to US dollars by taking the 2015 average exchange rate.

Vulnerable software ratio is the proportion of domains operating on vulnerable software installations hosted by the providers in our study, as explained in Chapter 4. The 'vulnerable software ratio' is calculated by dividing the number of scanned domains with Wordpress installations by all scanned domains excluding those that we were unable to scan.

5.4.2 Effect of Providers' Structural Characteristics

To disentangle the effects of the various structural characteristics of hosting providers which we have outlined previously on the concentration of C&Cs, we use a generalized linear model (GLM) with log-linear link-function of the form:

$$\begin{aligned}\ln(\lambda_i) = {} & \beta_{1i} AllocatedIPSize + \beta_{2i} WebhostingIPSize \\ & + \beta_{3i} DomainSize + \beta_{4i} SharedHosting \\ & + \beta_{5i} RuleofLaw,\end{aligned}$$

where the dependent variable – count of C&C domains – follows a Poisson distribution with parameter $\lambda \geq 0$ and βs are the estimated coefficients for the explanatory variables collected for all the hosting providers. Subscript i refers to measurements in different hosting providers.

Table 5.3: Generalized Linear Regression Model (GLM) for the Population of Hosting Providers

	Response Variable: Count of C&C domains				
	Poisson with Log Link Function				
	(1)	(2)	(3)	(4)	(5)
Allocated IP space size		−0.991***	−0.356***	−0.358***	−0.398***
		(0.019)	(0.020)	(0.020)	(0.020)
Webhosting IP space size		2.725***	0.711***	0.868***	0.931***
		(0.020)	(0.027)	(0.031)	(0.032)
Domain name space size			1.465***	1.301***	1.300***
			(0.014)	(0.020)	(0.021)
Portion of Shared hosting business				0.009***	0.009***
				(0.001)	(0.001)
Rule of Law					−0.213***
					(0.013)
Constant	−1.380***	−5.058***	−6.834***	−7.319***	−7.130***
	(0.009)	(0.039)	(0.049)	(0.066)	(0.068)
Observations	46,455	45,358	45,358	45,358	45,166
Log Likelihood	−50,777.110	−21,665.390	−15,485.920	−15,418.070	−15,253.920
Akaike Inf. Crit.	101,556.200	43,336.780	30,979.840	30,846.140	30,519.850
Dispersion	46.62	11.049	9.296	9.641	10.328
Pseudo R^2		0.587	0.717	0.719	0.722
Note:					*p<0.05; **p<0.01; ***p<0.001
					Standard errors in brackets

We construct several models using the variables in the equation above, for the whole population of hosting providers. Our goal is both to maximize the amount of overall explained variance of the model and to find out which of these variables influence the concentration of C&C abuse in providers the most. The result of our regression models are displayed in Table 5.3.

It is important to note that the reason for building more than one model is to be able to compare goodness of fit values while adding new variables to each model. Hence, to asses how the models are performing in absolute terms and relative to the other models, we use the Log-likelihood, AIC statistic, the Poisson dispersion parameter and the pseudo R-squared as measures of goodness-of-fit. We aim to minimize log-likelihood and AIC (the closer to 0 the better). The Poisson model assumes that $var[Y_i] = \phi E[Y_i] = \phi\lambda_i$, with $\phi \stackrel{!}{=} 1$, where ϕ is a dispersion parameter. The dispersion parameter hence captures the extent to which variance is different from the mean and more specifically the heterogeneity of the model. A model is over-dispersed when ϕ>1. The pseudo R-squared [149] is likewise calculated for our Poisson model given the dispersion parameter ϕ, using the following formula :

$$R^2 = 1 - \frac{D(\boldsymbol{y}, \hat{\boldsymbol{\lambda}}) + k \cdot \hat{\phi}}{D(\boldsymbol{y}, \bar{Y})}, \tag{5.1}$$

where $D(\boldsymbol{y}, \hat{\boldsymbol{\lambda}})$ is the deviance of the fitted model, $D(\boldsymbol{y}, \bar{Y})$ is the deviance of the intercept-only model, $\hat{\phi}$ is the estimated dispersion parameter and k is the number of covariates fitted, (excluding intercept). By building several models, we aim to maximize the value of the pseudo R-squared hence maximizing the amount of variance explain in C&C abuse counts by the dependent variables.

By inspecting Table 5.3, model 1 is the intercept-only model with count of C&C domains as dependent variables and no independent variable. In Model 2, we take into account the size variables – 'Allocated IP space size' and 'Webhosting IP space size'. The model indicates a significant negative relation between the variable 'Allocated IP space size' and C&C abuse counts, while 'Webhosting IP space size' correlates positively with C&C abuse counts. This is very much expected as pointed out earlier in the chapter, these two variables together determine to what extent the provider is using its allocated IP space for web hosting services. In addition, our manual inspection of the hosting data shows providers with very large allocated IP space are normally not pure hosting providers but rather broadband providers who use a small portion of their IP space for hosting. Moreover, the value of our goodness-of-fit criteria shows that only by adding these two size variables, we have substantially reduced the log-likelihood, AIC and dispersion values and are able to explain approximately 58% of the variance in abuse counts.

We build on model 2 by including additional variables, namely 'Size of the domain names space' along with the extent to which a provider is hosting its domains on shared hosting services. Model 4 displays the estimated coefficients.

The results indicate more domains in general and specifically shared hosting domains relate significantly with more C&C abuse. To put the value of the coefficients in perspective, by holding all other values constant in the model, a unit increase in the value of 'Size of the domain names space', multiplies the number of C&Cs by $e^(1.300) = 3.7$.

In addition to size variables analyzed in Models 1 to 4, we hypothesized that the rule of law index of a hosting provider's country might play a significant role in explaining the concentration of abuse in that country. Previous work has shown that the location of banks targeted by Zeus malware is not random [161]. Similarly, our dataset shows that some C&Cs are hosted in several islands all around the globe which are mostly the so-called tax-heavens. We examine this effect by including the Rule of law index variable in addition to the other previous 4 variables in model 5. We see a clear negative relation between the rule of law index and the concentration of C&Cs abuse. Although the Rule of law index is a combination of several country-level regulation indicators, it provides valuable insight about the proportion of abuse in certain geographical locations.

With the fitted values in our final model – model 5 –, we are able to explain approximately 72% of the observed variance (i.e., Pseudo R-squared = 0.72) in C&C counts only through considering the size variables and the rule of law. This highlights a very important point: regardless of the security measures a provider has in place, certain characteristics, driven by the nature of a provider's business, are driving the majority of the abuse.

Note we hypothesize that there are additional factors that influence the concentration of C&C abuse in hosting providers such as variables that capture the business model of a provider, for example price of a hosting service. However, such variables are much harder to collect at scale for all hosting providers. In the next section, we assess the impact of such factors on the concentration of C&C abuse within a smaller sample of hosting providers.

5.4.3 Concentrations of C&Cs in a Sample of Providers

As explained earlier, we collected additional business model variables for a sample of providers 5.4.1. We initially started from a set of 235 randomly selected providers for which we collected price information, however due to missing values in other variables we ended up with 85 providers for whom we have data on all the four variables in this category.

Note that the downside of our sampling strategy is that we might end up with geographical biases. In order to control for such effects, we fit a "fixed-effects" GLM model with the count of C&C domains as dependent variable

following a Poisson distribution. We add a country fixed effect, δ_i, by fitting a separate dummy variable as a predictor for each country. The country fixed effect prevents undue dependence of the residuals.

Table 5.4: Generalized Linear Regression Model (GLM) for a sample of hosting providers

	Response Variable: Count of C&C domains					
	Poisson with Log Link Function					
	(1)	(2)	(3)	(4)	(5)	(6)
Best price		-0.004^{***}	-0.019^{***}	-0.043^{***}	-0.018^{***}	-0.084^{***}
		(0.001)	(0.002)	(0.006)	(0.005)	(0.015)
Time in business			0.063^{***}	0.075^{***}	0.060^{***}	0.070^{***}
			(0.006)	(0.008)	(0.009)	(0.014)
Vulnerable software ratio				1.463^{***}	1.462^{***}	2.035^{***}
				(0.327)	(0.366)	(0.508)
Popularity index					0.00001^{***}	0.00002^{***}
					(0.000000)	(0.000002)
Constant	-4.146^{***}	-2.363^{***}	-3.121^{***}	-3.533^{***}	-3.881^{***}	-20.624
	(0.011)	(0.024)	(0.095)	(0.145)	(0.160)	(2,103.363)
Country fixed-effects	No	No	No	No	No	Yes
Observations	45,363	230	144	85	85	85
Log Likelihood	$-21,854.350$	$-1,625.306$	$-1,133.003$	-715.212	-564.210	-343.260
Akaike Inf. Crit.	43,710.690	3,254.612	2,272.005	1,438.424	1,138.420	754.521

Note: *p<0.05; **p<0.01; ***p<0.001
Standard errors in brackets

Similar to before, we add the variables one by one to the baseline model (model 1) to observe the extent to which a model is improved in comparison to others. The resulting models are shown in Table 5.4. Model 5 contains all of the 4 variables discussed before. In addition to those, in our final model, (model 6), we add the country fixed-effect variable as well.

Inspecting the estimated coefficients of model 6, we observe a significant negative relation between price of hosting and C&C counts. That is to say that if we were to increase price by one unit while holding all other variables constant, the C&C counts would be multiplied by $e^{(-0.084)} = 0.91$ as a result. The cheaper a provider's price is, the more likely it is for the hosting provider to host C&C domains. As expected, the variable 'Best price' shows a weaker relation in model 5 where cross-country differences are not controlled by the fixed country effects of model 6. This is because the properties of hosting markets in different countries can differ substantially, which then eventually influence the

cost of infrastructure in a country with respect to hosting services. Moreover, the cost of a hosting plan is in proportion to the economy of the provider's country. Hence, our conversion of prices in different specific countries to USD, if not controlled for the country differences, can be very crude.

The Variables 'Vulnerable software ratio' and 'Popularity index' also show a significant positive relation with C&C counts. One unit increase in 'Vulnerable software ratio' while holding other variables constant, multiplies the C&C counts by $e^{(2.035)} = 7.652$. The 'Time in business' variable shows a significant positive relation with abuse as well, indicating that well-known providers or those who are in business for a longer time are attacked more. Note that this can partially be caused by the fact that our data is longitudinal. In the following section, we will study the effect of C&C take-down speed on its concentration across providers.

5.5 Effect of C&C Take-down Speed

Up to this point, we have demonstrated that the concentration of C&C domains can be explained by structural characteristics of providers, mostly related to their size and business model. Together, these factors form a proxy for the attack surface of the industry. The attack surface of providers accounts for at least 72% of the variance in the number of C&C in their networks. Providers with more infrastructure get more C&C. This does not indicate selective location choices by the attackers. Quite the opposite, in fact. The bulk of C&C can be explained from attackers randomly distributing their C&C domains across the overall global hosting infrastructure.

In this section, we investigate whether attackers prefer providers who are lax in taking down C&C servers. Longer uptime of C&Cs seems valuable for the attackers, so we would expect higher C&C counts in those networks. We examine if and how C&C uptimes influences the number of servers at that provider. C&C uptime has been used in previous security research as a standard metric for studying the lifetime of different attack types [136].

We define the "uptime" of a C&C domain as the number of days between the first and last time the C&C domain is observed online as reported by our datafeeds. Some of the C&C domains remain online beyond the measurement period, which unavoidably leaves their uptime unknown. The average uptime of C&C domains is depicted in Figure 5.6. There is no clear trend one way or the other. This also suggests that there are no learning effects among hosting providers that enables faster take-down over time. We first examine if the

average C&C uptime is driven by a few providers, or whether it reflects the overall performance of hosting providers.

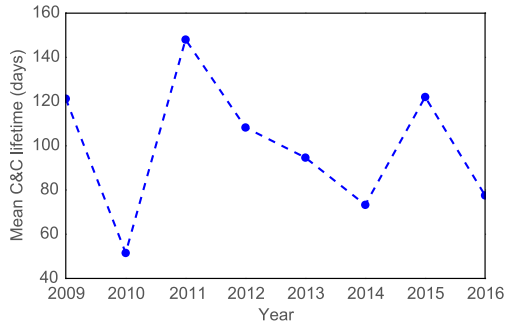


Figure 5.6: Mean uptime of C&C domains per year

5.5.1 Distribution of C&C Lifetime

Are long-lived C&Cs concentrated in certain providers? Figure 5.7 depicts the distribution of the average uptime of C&C domains per provider, for 2009-2016.

Note that each individual plot is also indicative of the amount of C&C domains that are taken down in the corresponding year. What is clear from all the plots is that, there is always a majority of providers with a shorter uptime followed by a long tail of providers hosting C&Cs with very long uptimes. In some case there are examples of providers that hosts C&C domains for more than a year. Assuming no measurement errors are at play here, such examples could indicate ignorant or perhaps even bulletproof hosting. This leads us to the next question: are these providers preferred by attackers?

5.5.2 Differences between C&C Take-down Speed of Providers

To examine the differences between providers more carefully, we model the survival rate $S(t)$ of C&C domains using a Kaplan-Meier Survival Estimate which also allows to correctly account for the C&C domains that are not taken down by the end of our measurement period, i.e., right-censored data points [131]. The survival rate $S(t)$ basically expresses the probability that a C&C domain is online at a specific time during the observation period.

Figure 5.8 displays the survival curves of C&C domains in the top-10 providers with the highest number of C&C domains in their network. Figure 5.9 depicts

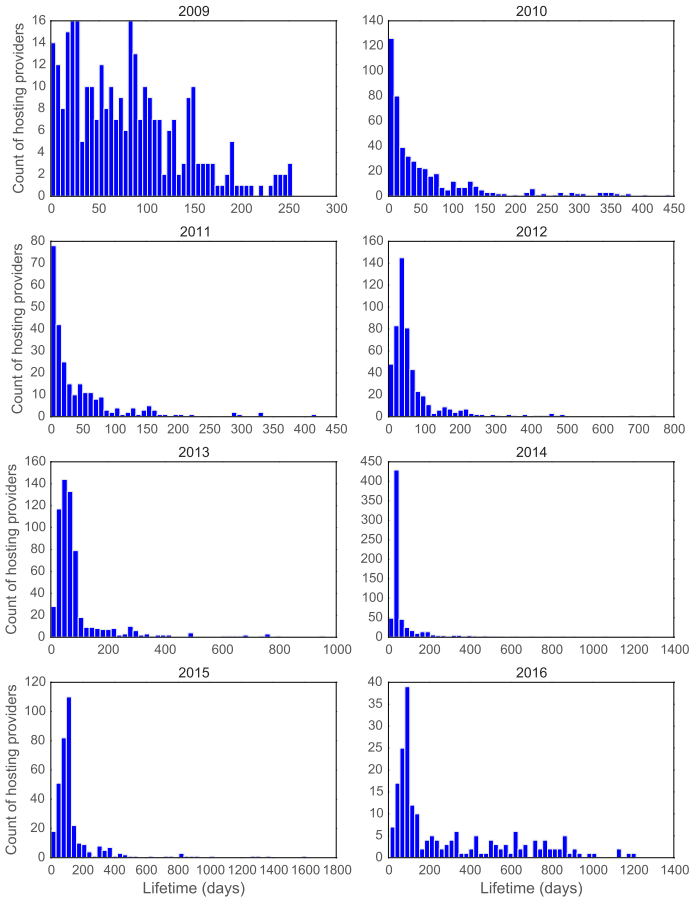


Figure 5.7: Distribution of C&C average uptime hosted by providers over years

the χ^2 value of the Log-Rank test in which the providers are compared two by two in terms of their survival rate. Only the light blue tiles indicate non-significant differences at a 0.05 significance level.

As both plots suggest, hosting providers perform differently either in terms of survival probability or in terms of the total number of days that their C&C domains remain online. For example, more than 95% of the C&C domains in `Main Hosting Server` are taken down after approximately 60 days which is

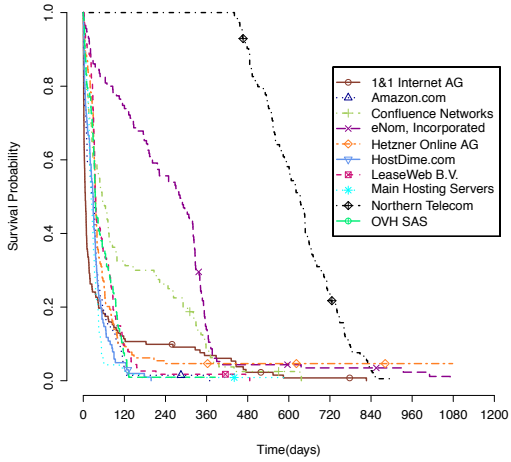


Figure 5.8: Kaplan-Meier estimated survival function of C&C uptime for top-10 most attacked hosting providers

very similar to `HostDime.com`. However C&C domains hosted by `HostDime.com` are in total taken down after maximum of 4 months whereas this takes about more than a year for `Main Hosting Server`. On the extreme side are providers such as `Northen Telecom` and `eNom, incorporated` that host C&C domains that are online for more than 2 years.

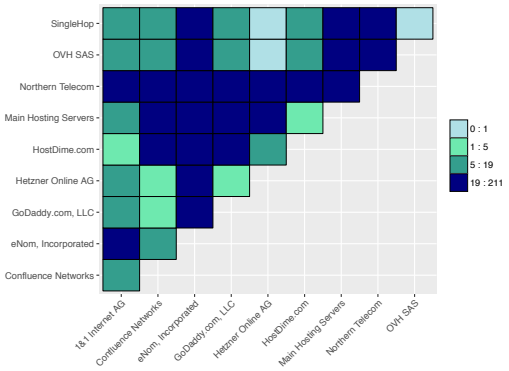


Figure 5.9: Log-rank test for pairs of providers (legend shows χ^2 value of Log-rank test)

5.5.3 Does C&C Uptime Explain Abuse Concentration?

Given that there are significant differences in the C&C uptime among hosting providers, we now analyze the impact of C&C uptime on the concentration of C&C abuse within providers. Do attackers prefer providers with long C&C uptimes?

Table 5.5: Generalized Linear Regression Model (GLM) for all hosting providers

	Response Variable: Count of C&C domains		
	Poisson with Log Link Function		
	(1)	(2)	(3)
Allocated IP space size		-0.398^{***}	-0.447^{***}
		(0.020)	(0.020)
Webhosting IP space size		0.931^{***}	1.054^{***}
		(0.032)	(0.032)
Domain name space size		1.300^{***}	1.162^{***}
		(0.021)	(0.020)
Portion of Shared hosting business		0.009^{***}	0.011^{***}
		(0.001)	(0.001)
Rule of Law		-0.213^{***}	-0.170^{***}
		(0.013)	(0.013)
Average C&C uptime			0.003^{***}
			(0.0001)
Constant	-1.380^{***}	-7.130^{***}	-7.077^{***}
	(0.009)	(0.068)	(0.069)
Observations	46,455	45,166	45,166
Log Likelihood	−50,777.110	−15,253.920	−14,699.260
Akaike Inf. Crit.	101,556.200	30,519.850	29,412.510
Dispersion	46.623	10.328	9.032
Pseudo R^2		0.722	0.733
Note:		*p<0.05; **p<0.01; ***p<0.001	
		Standard errors in brackets	

We fit a similar GLM model to model 5.3 from Section 5.4.2 for the population of providers, having the count of C&C domains following a Poisson distribution. The result of our model with the addition of a C&C uptime variable is shown in Table 5.5.

In order to be able to make a relative comparison, model 2 is the final model from our inquiry into the structural properties of providers (see Table 5.3). We add the 'Average C&C uptime' variable to that model. The resulting estimated coefficients are observable in model 3. The model suggests that the variable 'Average C&C uptime' shows a statistically significant positive relationship with the number of C&Cs. Although the relationship is significant, we are only able to explain a total of 73% variance of C&C abuse counts, which is a 1% increase compared to the model with only structural provider variables. This rather indicates that there is very little or no preference by attackers for hosting their C&C domains at providers who allow long-living C&C domains.

5.6 Related Work

With the increasing number of attacks on financial services, efforts from industry and academia have focused on botnet evolution and mitigation strategies.

A first area of research aims to understand the functionality of the different malware families to develop countermeasures that could disrupt these botnets. Different studies have investigated the communication protocols of these botnets and their spreading techniques [162, 163, 164].

These studies have collected data on C&C and other botnet infrastructure. These are typically presented in a descriptive analysis, such as their distribution over countries. Rossow et al. [165] analyzed the lifetime and domain name characteristics of malware downloaders. They observed steady migrations of malware downloaders from domains and TLD registrars to others, and notice that attackers redundantly deploy their critical infrastructures across providers. Han et al. in [166] investigated the way cyber-criminals abuse public cloud services to host part of their malicious infrastructure, including exploit servers to distribute malware and C&C servers to manage infected terminals. Our work complements the insights obtained by these works by analyzing the factors that drive attackers to choose certain type of hosting provider.

A second strand of work has developed approaches to better detect botnet infrastructure. Cyberprobe [167] describes an active probing approach for detecting malicious servers and compromised hosts. ASwatch [33] aiming at detecting and identifying malicious ASes that exhibit "agile" control plane behavior (e.g., short-lived routes, aggressive re-wiring). In this context, fast flux also appears as a technique that uses compromised computers to provide scalability, geographic diversity, anonymity and redundancy to organized cybercrime operators. The fast flux infrastructure relies on computing resources stolen from

the unwitting users of infected endpoints. Cybercriminals rent these fast flux proxy networks to create a profitable black market hosting environment. The authors of [168, 169] have analyzed the structural relationships (domain, nameserver, IP connectivity) of fast-flux botnets and identified recurrent structural clusters across different botnet types. In [168], the authors have used a social network connectivity metric to show that {Command and Control and phishing} and {malware and spam botnets} have similar structural scores using the proposed metric. In this chapter, we have defined metrics to capture not only the attacker behavior but also the hosting provider effort toward mitigating the malicious infrastructure located in their networks.

A third strand of work is the development of reputation systems for providers, especially focused on those that facilitate cybercrime [34, 114, 118]. For example, FIRE [34] introduced a ranking system using uptime of botnet hosting services to identify and expose providers that demonstrate persistent, malicious behavior. In [114] the authors propose various reputation metrics based on the concentration of abuse, while taking some structural hosting provider characteristics into account. During the explanatory analysis conducted in this chapter, we use the structural properties of hosting providers to assess the impact of these on their security performance.

All these approaches help to identify and enumerate botnet C&C infrastructure and to describe their distribution across networks and countries. We extend this related work via explanatory analysis to determine the driving factors for the locations of the C&C infrastructure in the hosting market. We statistically model and explain the distribution of C&C from the structural properties of hosting providers, business models and factors like rule of law. We expand the work by Gañán et al. [136] by studying the properties of providers hosting C&C domains.

Hosting providers play a key role in the size and spread of these botnets. Different abuse reporting strategies have been proposed and evaluated to analyze the performance of hosting providers [135, 134, 170]. However, as shown by Canali et al. [10], hosting providers are often not taking appropriate measures, probably due to a lack of incentives. Millions of websites are often poorly managed by inexperienced users, shared web hosting providers have not developed reliable mechanism to keep their users safe. Moreover, with the emergence of cloud providers, attackers have a new platform to host their infrastructure. Current studies have shown that these type of providers are being used to launch long-tail spam campaigns because of their low cost [171, 172]. Only a few specific providers have attempted to create added value by providing "add-on" security

services. For instance, a Dutch web hosting provider [173] has added a free automated website vulnerability scanning, fixing and recovery service.

On the other end of the spectrum there are hosting providers acting as cybercrime facilitators [33, 174, 137]. Researchers and law enforcement agencies are searching better ways at squashing these providers. While these efforts are critical for the overall fight against cybercrime, our analysis suggests that the C&C of the botnets engaged in attacks on financial services do not depend on malicious hosting providers, nor do attackers seem to prefer these providers when locating their C&C.

5.7 Conclusions and Discussions

Over the years, hosting providers have spent a great deal of effort taking down C&C infrastructure for botnets engaged in attacks on financial services.

This chapter aimed to enlighten the strategies of the attackers using these botnets for the placement of their C&C servers across the hosting market. More specifically, we examined if attackers have shown a preference for providers with lax security efforts. Or, conversely, whether the placement choice of C&C domains is rather randomly distributed across the hosting space, as measured via the provider's structural properties.

We studied seven years of C&C data for 26 botnet families engaged in attacks on financial services and demonstrated a general increase in the total number of providers hosting C&C domains over time. We also found a dynamic pattern of providers who enter and exit the population of providers that host financial malware C&C. Our results show that C&C abuse is highly concentrated in a small number of providers. That being said, this concentration can be explained from relatively large portion that these providers have of the overall attack surface of the hosting market.

To study the effect of hosting provider characteristics on C&C concentrations, we modeled the distribution of C&Cs using Generalized Linear Models (GLM), with C&C counts following a Poisson distribution. We showed that a provider's attack surface characteristics such as IP and domain space size and the proportion of shared hosting can explain around 71% of the variance in the number of C&Cs per provider. The rule of law in a country only explains an additional 1% of the variance, suggesting that the attackers do not prefer providers in jurisdictions with weak law enforcement. All in all, the selection process for C&C seems to be random: the probability of hosting C&C is highly

proportional to the attack surface of the providers, as measured the by observed effect of indicators of size of the provider.

In addition, business model characteristics of providers show a significant relation with C&C concentrations for a sample of hosting providers. While the pricing of a hosting plan negatively affects C&C concentrations, provider's popularity, time in business and the ratio of vulnerable software, have a significant positive relation with C&C concentrations. Despite statistically significant differences in C&C take-down speeds among providers, when modeled in conjunction with attack surface variables, take-down speed shows only a very weak relation with the concentration of C&Cs across providers, suggesting that attackers are rather impervious to the take-down efforts of hosting providers.

On a more general level, our results suggest that the amount of C&C abuse in the network of a provider is a function of a provider's structural properties such as its size and its pricing strategy, rather than being driven by the effort they put in abuse handling.

Additionally, our approach helps in developing evidence-based policies in the hosting market. That is, we demonstrate an approach that enables better comparative abuse metrics by controlling for the structural differences among providers rather than relying on absolute counts.

Our work comes with a set of limitations as well. The dataset contains only malware families that have been used to attack financial institutions. Some are predominantly used for this purpose, like Citadel, but others are much more generic malware families. Although our methodology in generalizable, it is an open question whether the patterns we found are different for different kinds of abuse data. Future work could explore this. In addition, our uptime analysis can contain biases from unknown measurement errors in the first-seen and last-seen observations of C&C domains. Such observations are known to be quite noisy. We do however think that the effects would be negligible since the biases (if any) would be systematic. Finally, because we have used pooled data for the whole measurement period, our models do not account for changes of C&C counts over time. Future work can look into whether these patterns we discussed in this chapter change over time.

CHAPTER 6

Understanding Attacker Behavior

In the previous chapters, we focused on quantifying the impact of hosting provider properties and effort on abuse, given random attacker behavior – that is, attacks are randomly distributed across the attack surface, as measured by the exposure indicators. In this study we shift our focus to the attacker's side and aim to study attacker behavior in target selection, as observed in Zeus financial malware data. Using Zeus configuration files, the first part of this chapter explores different characteristics of a target that increase the likelihood that it is attacked. The second part investigates the attacker behavior in sharing, buying, and changing the attack code of Zeus malware.

6.1 Introduction

Online banking fraud has increased in past decades as web-based banking platforms have become popular among consumers and businesses. A variety of controls and countermeasures have been put in place by the banking sector and security firms, from better authentication of users to real-time supervision of transactions. Yet, online banking fraud remains a serious problem [175]. The annual global losses caused by financial fraud are in the magnitudes of billions of Euros [109]. The European Central Bank recently published fraud statistics for the Single European Payment Area, reporting "card-not-present" (CNP) fraud at around 800 million Euros (approximately $1.1 billion) [176].

Notwithstanding the fact that impacts are substantial across industrialized countries, we also see remarkable differences in fraud levels. For 2012, UK published a total loss of around 299 million Euros for CNP-fraud [177]. Over the same year, France reported 160 million Euros and the Netherlands reported 35 million Euros of online payment fraud [178, 179]. Relative to the number of inhabitants, France and the Netherlands suffer roughly half the level of online

fraud of the UK. Sullivan has estimated relative fraud levels in different countries in 2006. He found that Spain and Australia experienced the lowest rates of fraud, around $.022 and $.024 per $100 of transactions respectively. This is while the UK and US suffered worse fraud levels, losing $.086 and $.092 respectively [180].

One of the foundations for designing effective mitigation strategies would be to better understand what factors drive the differences in fraud levels. An obvious driver is the extent to which payment services are targeted by attackers. Very little empirical work has been done regarding the underlying reasons of why certain targets are selected more often than others. Financial service providers differ in many respects such as total revenue, market share, number of users, authentication mechanisms, money transferring policies, regulatory framework, and the properties of their home markets. Yet, we do not know how these differences affect provider's relative attractiveness as a target.

An important hurdle for work on attacker's preference is the fact that not much data is available from which target selection patterns could be extracted. In this chapter, we present a hitherto untapped source of data on target selection by cybercriminals: the instructions sent to the machines infected with financial malware or *banking Trojans*. Given that many attacks utilize financial malware, these instructions provide an insight into the population of targets that have been selected by the attackers. We have studied a dataset of instructions, so-called configuration files, which have been distributed within the ecosystem of Zeus botnets. Analyzing a set of 11,000 malware configuration files, intercepted over 4 years and containing 1.2 million targeted URLs, we specifically set out to increase our understanding of the underground economy around malware-based financial fraud and answer the following questions:

- What services have been targeted via Zeus malware (Section 6.4)?
- What metrics could be developed to rank the relative attractiveness of targets (Section 6.5)?
- What factors could explain the target selection behavior of attackers (Section 6.6)?
- How are new targets identified (Section 6.7)?
- What is the effect of Zeus source code leakage on the target selection of attackers (Section 6.7)?
- How does the inject (attack) code develop over time (Section 6.8)?

By answering the aforementioned questions, we aim to lay the foundation for future research into the interactions between the security trade-offs of financial service providers and those of the attackers.

6.2 Background

In the course of the 1990s, banks started offering access to the bank's computer systems via the Internet using a browser or specific application [181]. The online channel reduced the need for costly retail branches and paper transactions. This was not only a way to offer new services, but also a strategy that created cost-savings for financial institutions [181, 182]. More recently, mobile devices have become another channel for electronic banking activities. Predictably, these innovations also meant that financial services became the target of a variety of online attacks.

6.2.1 Online banking fraud

Online banking fraud is typically account takeover: removing money from someone else's bank account. It can take place via different attack vectors. Two of the most prominent types of attacks are credential stealing and content manipulation, which can be used separately or in combination. Credential stealing attacks attempt to access users' credentials via phishing or through the use of financial malware [183]. Content manipulation, also called man-in-the-browser (MitB) attacks, installs malware to manipulate the ingoing and outgoing communication between the unaware user and the bank, at the system level [184]. This type of attack allows attacker to be selective in choosing the target domains and the type of data she intends to steal or manipulate [183].

6.2.2 Zeus malware

Zeus, also known as Zbot, is a readily available malware kit that contains the tools required to build and control a botnet. The kit is very simple to use, since it does not require any in-depth technical knowledge [185]. Zeus was first exposed on July 2007 and worked on computers using Microsoft Windows operating system. Since 2012, there are also Zeus variants for Blackberry and Android phones. Zeus malware has been primarily known for its use in financial fraud, but its features also allow other types of data theft (e.g., password sniffing). The Zeus source code was leaked in 2011 and since then it has been sold and

traded widely in underground forums. Numerous variants of the original Zeus malware have appeared ever since [186].

Zeus malware operates based on instructions that are specified in a so-called "configuration file". The configuration file has two parts: the static and dynamic section. Information located in the static section is hard-coded into the bot executable and contains information that the bot needs when it is first executed, such as the URL where to get the dynamic section [187, 188].

The dynamic configuration file (config.bin) is downloaded by the bot immediately after it is installed on victim's computer. The static configuration contains an RC4 key, which is used to encrypt the communication within the botnet, including the dynamic configuration file. In this implementation, a key stream is generated from the botnet password, and is XORed with the data [188]. Note that the keys effectively segment the total population of Zeus clients into different botnets by connecting each one to a specific command-and-control (C&C) channel. It can therefore serve as a proxy for distinct attack campaigns and, thereby, of an attacker – though the latter is a lot less reliable, as an attacker can be behind multiple campaigns simultaneously as well, over time.

The file is updated by the C&C server and is downloaded by the bot at certain time intervals, providing it with new instructions. Most of the entries in the file control how and what information is collected from the infected computers, how to attack the banking clients and and where drop the information collected from the victims [187]. Note that for the sake of simplicity, these dynamic configuration files are called configuration files in the remainder of this chapter.

As soon as the victim's computer gets infected, the Zeus malware attaches itself to the user's web browser. This enables it to monitor everything the victim does on the web, including her online banking and credit card transactions [188]. Zeus records everything the victim types in the browser, including usernames, passwords and banking credentials. It then sends them back to the command and control server where information is stored in a *dropzone*. The criminal can then use this information directly to steal money from the victim's accounts or he can sell the information to other criminal organizations that have the infrastructure for large-scale online banking and credit-card fraud operations.

Moreover, Zeus can act as a Man-in-the-Browser (MitB) and modifies what the victim sees on her bank's web page. Fraudulent transactions are executed by modifying certain web pages and injecting data into certain fields, invisible to the user. These so-called 'inject codes' are located in a section called 'web Injects' within the configuration file. To illustrate: sometimes criminals inject extra fields to the bank's login webpage that ask for additional login information,

such as credit card details or PIN numbers. These type of information are normally not required for the login process. In other types of attacks, the webpage is modified to show a fake account balance to the user, thereby hiding a fraudulent transaction that has been executed in the background.

Given the fact that Zeus is the most widespread banking Trojan, a better understanding of its operations can yield useful insights into target selection and other criminal behavior around online financial services. To this end, this work is dedicated to exploring Zeus instructions included in the configuration files.

6.2.3 State-of-the-art: target selection

Earlier work has often focused on the technical vulnerabilities of banking services and on developing more secure online banking technologies [189]. However, several authors have pointed out how the incentives of financial service providers shape the security decisions associated with these vulnerabilities [190, 191, 192].

Moore et al. observed that attackers tend to favor certain financial services over others. Moore and Clayton studied a sample of phishing sites and found that some banks are targeted much more frequently than others [129]. PayPal was impersonated by 399 of the 1695 sites, while 52 banks were only spoofed once. They do not explain this discrepancy, except indirectly: banks can influence how long phishing sites stay online. Perhaps that serves as a deterrent. It is unlikely, however, that PayPal is much less vigilant than 52 rarely attacked banks.

A study on click fraud by [193] concluded that online-only banks were targeted more than banks with physical branches. In the analysis done by Levchenko et al. on the spam value chain, the authors found a significant concentration on certain merchant banks that assist sellers of online pharmaceuticals [48].

As far as we know, there has not yet been any in-depth empirical investigation into the extent to which some banks are targeted more often than others. While there are many factors at play, here, it seems clear that criminal's decisions and preferences play a major role. Recent security reports claimed that online banking attacks were getting more target-specific [194, 195]. This although suggests a conscious selection process on the side of criminals, factors that drive such decision process have not yet been uncovered. Is the selection process for online banking fraud based on specific characteristics of the bank, its policies, its location, or another set of considerations altogether? Perhaps the decisions of attackers are less guided by informed cost-benefit trade-offs and

more by herding behavior: in the absence of good information about the likelihood and magnitude of success, they mimic whatever other attackers (their peers) are doing, driven by underground forums or chat rooms where experiences are exchanged [196]. Yet another strategy might be to do the opposite: select targets that nobody else is attacking?

From the literature on economics of crime, Routine Activity Theory (RAT) can help to describe why criminals go after a certain target. RAT argues that for a crime to be committed, three ingredients are needed: a motivated offender, a suitable target, and the absence of a capable guardian at a specific time and place. RAT has been developed in the context of conventional "offline" crime. However, Yar has argued that the differences between the virtual and non-virtual worlds made the applicability of RAT to cyber-crime limited [100]. These differences include cyberspace's different socio-interactional characteristics, such as the collapse of spatial–temporal barriers, many-to-many connectivity, and the anonymity and plasticity of online identity.

Yar has adapted RAT to cybercrime. He identifies the four key properties that derive the so-called "suitable target" to be selected as: *value, portability, visibility, and accessibility* [100]. Value of a target in online banking fraud can be defined as the value that can be gained by the offender if the attack is successful. This might mean that banks located in richer countries or with higher account balances would be selected more often, all other things being equal. Portability is about the ease with which the criminal gains can be moved, such as money being transferred in near real-time via irreversible transactions. Visibility is about how visible and exposed the target is to the cybercriminals. Finally, accessibility is about how easy the target can be reached.

In summary, we are not aware of any prior empirical work that attempts to identify the factors that drive criminals to target certain online financial service providers more often than others, in online banking fraud.

6.3 Data Collection Methodology

6.3.1 Dataset : Zeus configuration files

In this chapter, we analyzed a collection of Zeus financial malware data files provided to us by Fox-IT, a leading Dutch security firm with many clients in the financial sector. The dataset consists of around a hundred and fifty thousand (144,625) captured files that were suspected to be configuration files. Of these, we investigated 10,673 configuration files that revealed targeted domains.

The rest of the captured configuration files could not be used due to either of the following reasons: (i) the file could not be decrypted with one of the keys extracted from the executables or due to an unknown format; (ii) the file was decrypted, but was not actually a configuration file; (iii) the file was corrupted or incomplete (some files were only captured in full after multiple attempts); (iv) the file was decrypted but did not contain a web inject section and therefore no information regarding the targeted domains[1].

The configuration files were collected over a period of just over four years (2009-2013Q1). They were captured using honeypots located all over the world (with more concentration in western countries and less in Asia). The configuration files were collected using two different methods: they are gathered by running live Zeus samples or by emulating the malware to download configuration files.

The configuration files are encrypted plain text files. Each captured file, along with the time stamp of when it was captured and the key with which it was decrypted, if applicable, was stored in a MySQL database. As already explained in section 6.2.2, each Zeus configuration file contains a 'web inject' section which includes targeted URLs, attack instructions, HTML scripts to be injected into the pages served from the attacked URL, and mechanisms for bypassing the authentication procedures of the institution. Below is an example of the web inject section of Zeus configuration files. These configuration files typically contain multiple injects (113 of them, on average), each of which varies from just a few lines of code, as in the example below, to over two thousand lines:

Table 6.1: Example of Zeus inject code targeting a specific URL in configuration file

```
set_url  https://removed.com/OLB/secure/AccountList.aspx <FLAG_GET><FLAG_LOG>
         data_before
         id="dgDepositAcctsheader0"*>
         data_after
         </table>
         data_inject
         datas_end
```

[1]Sometimes the malware only monitors the machines http and https post requests and gathers system information without modifying anything. That is why some of the configuration files do not contain the web inject section and could not reveal targeted URLs.

6.3.2 Extracting targeted domains

The first step towards answering our research questions is to parse the configuration files. First, we extract targeted URLs from the configuration files and associate them with the time stamp and the RC4 key belonging to the corresponding file.

The configuration files revealed that a total of 14,870 unique URLs were targeted. Many of them contain different paths of the same domain. All in all, we identified 2,412 unique domain names. Extracting the domains from targeted URLs was not a straightforward process, as some configuration files contain URLs with wild cards. Table 6.2 displays some examples of such URLs.

Table 6.2: Example of target URLs in Zeus configuration files

Targeted URL	Targeted domain
`*/bancopostaonline.poste.it/*/formslogin.aspx`	`poste.it`
`*banking.*sparkasse*.de/cgi/anfang.cgi*`	`sparkasse.de`
`*mpresas*gruposantander.es*opaccesoempresasabe*`	`gruposantander.es`
`http://www.google.*&q=*`	`google.?`

Figure 6.1 contains information regarding the algorithm we use to extract the targeted domains from the URLs. When URLs do not contain wildcards, we extract targeted domains by trimming the path. When URLs include wild cards, we reconstruct the domains using regular expressions for comparing the last part of the URL (path, query or fragment) against the set of targeted URLs without wildcards. If a URL matches the same URL without wildcard in more than 90% of the cases, we assume that the targeted domain is the same as the domain of the URL without a wildcard. For around 6% of all URLs we could not reliably determine the targeted domain. Either the URL did not match any of the URLs in our set, or it matched with multiple URLs and none of them reached the threshold of 90%.

6.3.3 Extracting botnet keys

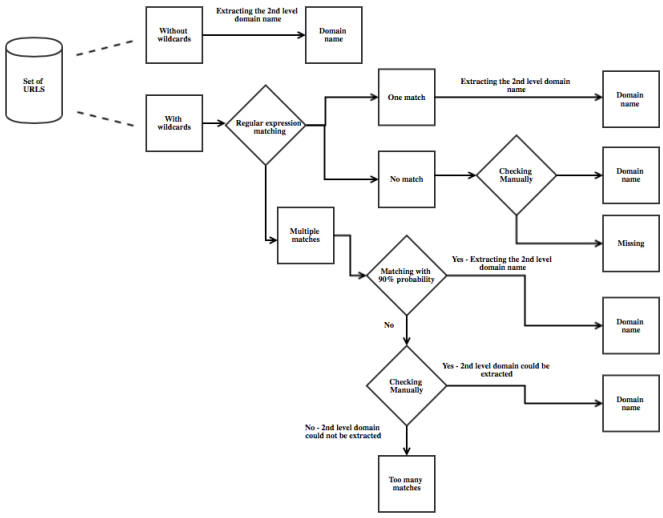
Ideally, we would like to count the number of attackers going after each domain. However, there currently is no reliable way to measure this. We use the RC4 keys as proxies for the number of botnets in use, i.e., the number of campaigns that have been undertaken – in effect treating the keys as unique identifiers for different Zeus botnets. These keys can only be changed by updating the malware with a new binary and migrating the command-and-control server to a new URL. Typically, this happens when a botnet is taken down. The operator then

continues with new bots and a new location for downloading the configuration, which technically makes it a new botnet, as most of the old bots are no longer able to connect.

6.3.4 Extracting targeted countries

We have enrich the dataset by adding some information about the targeted domains from other sources. The geographical location of targeted domains is determined semi-manually. For this purpose, we have used four sources of data: (i) Where the server/infrastructure is located [197]; (ii) Where the domain's traffic is coming from, according to Alexa [198]; (iii) Where the site owner's headquarter is located, according to the domain's homepage; (iv) The top-level suffix of the domain (TLD). However, most of the times, these sources consistently pointed to the same county. We manually checked cases where they did not match. One interesting pattern that emerged and could be explored in future work is that quite a number of the banks were located in micro-states and known tax havens. These *offshore* banks are most probably used by a small fraction of the population, and it is interesting to see that Zeus has been used to target this small group.

Figure 6.1: Algorithm used to extract targeted domains from targeted URLs



6.3.5 Size of targeted domains

We estimate the size of a target in two ways: using the traffic volume that is based on Alexa ranking, and for U.S. financial institutions, also by the total of deposits held by the institution, as reported by the U.S Federal Deposit Insurance Corporation (FDIC). The FDIC is a government corporation operating as an independent agency that provides deposit insurance guaranteeing the safety of a depositor's accounts in member banks. As of February 2014, they insure 6,790 institutions [199]. They provide certain statistics for these institutions, such as its total assets, deposits, the locations of its headquarters, its web address, etc. Through the web address field, we can connect 170 of these institutions to the data in our Zeus dataset.

Data on the traffic volume of the targeted domains is gathered from Alexa Internet, a subsidiary of Amazon.com that provides commercial web traffic data [198]. Using data gathered via the Alexa tool-bar and that provided by sites owners, Alexa ranks sites based on their traffic data. They also use data from the DMOZ open directory project to categorize websites [200]. We have pulled in this data from the Alexa website for the majority of the domains in our dataset. In the later sections of the chapter, we use the FDIC deposit and Alexa ranks as proxies for domain size.

6.4 Descriptives of the Zeus Attacks

6.4.1 Targets

As described in section 6.3.2, from a total of 14,870 unique targeted URLs, we have identified 2,412 unique domains. Not all of these are financial service providers. Among the targeted domains we could find anti-virus companies, news websites, webmail providers, and social networks, along with domains we could not categorize, because the sites were no longer online. Using Alexa categories, we are able to map 43% of all domains to a specific sector. Of these mapped domains, 760 (74%) are financial service providers and 272 (26%) domains belonged to other sectors. The remaining 1,380 domains are uncategorized.

Among uncategorized domains, we selected a random sample of 100 and manually checked their associated sector(s). Of these domains, we were able to map 72% of them. Of these mapped domains, 53 (73%) were financial service providers and 19 (26%) were from other categories such anti-virus companies, security service providers, and online consultancy firms.

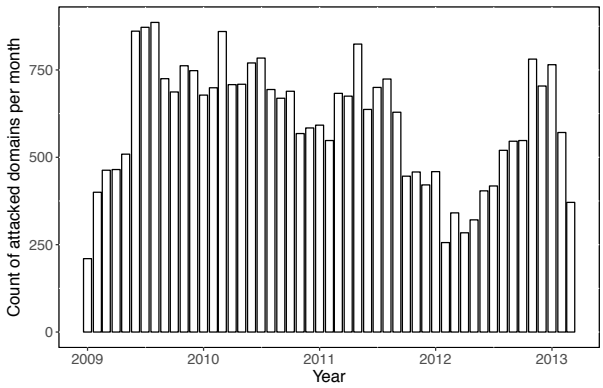


Figure 6.2: Number of attacked domains per month (2009-2013Q1)

Over the whole period, on average 600 unique domains were attacked each month across all observed botnets (Figure 6.2). In section 6.6, we will explore how the number of attacked domains vary over time in more detail. In terms of the geographical coverage of the data, the targeted domains are located in 92 different countries all over the world (Figure 6.5). Unsurprisingly, some countries suffer substantially more attacks than others. In section 6.7 we will discuss these geographical distributions in more detail.

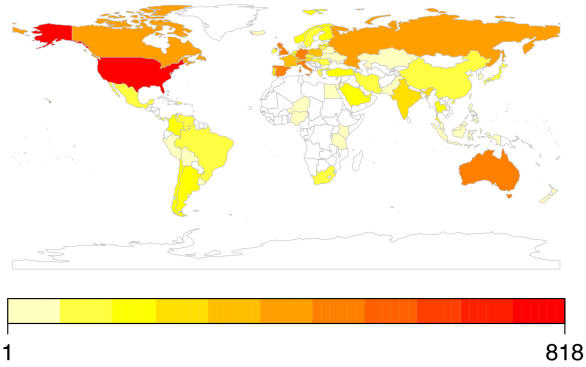


Figure 6.3: Geographical distribution of the attacked domains (number of domains per country in the dataset (2009-2013Q1)

6.4.2 Attackers

As detailed in Section 6.3.3, there are 2,131 unique RC4 keys in our dataset, which we interpret as an approximation of the number of botnets in use. Figure 6.4 displays the trend of botnet activity from January 2009 up to March 2013. The blue (upper) line displays the number of configuration files sent each week by all botnets together. The black (lower) line indicates the number of botnets that were active in that week, as counted by the total number of RC4 keys in use. As the trends indicate, the number of active botnets decreases over time. The same happens with the number of configuration files that were distributed. This might be attributed to the Zeus take-down efforts that were coordinated by Microsoft with different governments and security firms around the world [201], although the downward trend had started well before those efforts.

Comparing the number of configuration files against the number of active botnets per month (the two lines in Figure 6.4), we see that they roughly follow the same trend. This is to be expected, as the number of active botnets is determined by whether or not they have distributed a configuration file that week. However, it also can be seen that, on average, botnets sent out multiple files. This is unevenly distributed. Some are much active than others. This discrepancy highlights that raw counts of the number of times a domain shows up in configuration files is not really a good metric for the relative degree in which a domain is attacked. Accordingly, we explore more informative metrics in the next section.

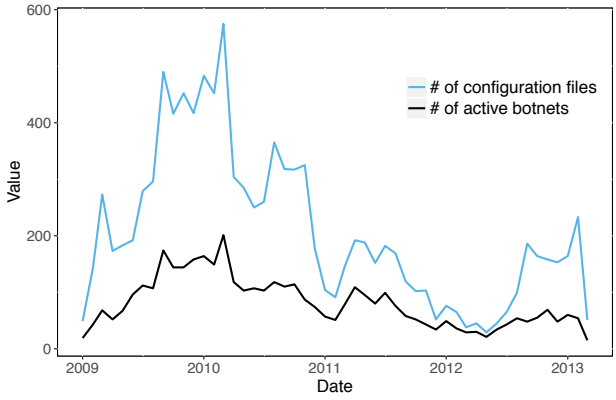


Figure 6.4: Number of configuration files and active botnets per month

6.5 Attack Metrics

To understand how popular different domains are as targets of online banking fraud, we need to rank them against a reliable metric. Until now, there is little empirical work on the popularity of targeted domains. Even where it exists, the ranking is based on poorly conceived metrics. For example, a security paper published by F-Secure reported a list of top-20 most attacked domains by the SpyEye malware in 2012 [202]. The rankings are calculated by simply counting the number of times a domain appeared in the malware configuration files. We believe that such raw counts are not reliable, mainly because the number of times a configuration file is sent does not necessary hold a one-to-one relation with the number of attacks. Taking the example of SpyEye malware, the configuration file is built into the binary, so attack instructions are released as often as the binary is changed. These changes are likely to be driven by signature updates in the anti-virus software that SpyEye tries to evade, rather than by the target selection process of cybercriminals.

There are numerous ways in which bot herders may choose to update their configuration files; one may update a configuration file once per day, while another one might adopt a lower update frequency, perhaps because she herds multiple botnets or the botnet has more stable attack code. Therefore, the number of configuration files per day sent by a botnet may have little correlation with the actual attacks and, thus, with target popularity. To illustrate, Figure 6.5 shows the configuration files of three different Zeus botnets sent in the same week. Using the raw counts, one would say Botnet 1 attacked ebay.com three times this week and Botnet 3 two times, so in total ebay.com is attacked five times in this week. However, in practice, all of the configuration files sent by botnets in one week (as an estimate of a threshold) are only the updated versions of the initial ones. It makes no sense to count them as separate attacks.

In short, to determine the popularity of targets and to deal better with these differences, we need to develop more reliable metrics. Below we suggest three alternatives as indicators for domains attractiveness:

- **Number of botnets attacking a domain:** Using this metric, domains attractiveness is defined by the number of Zeus botnets, counted by different RC4 keys that simultaneously targeted a domain.
- **Number of weeks a domain was under attack:** The Zeus data can also provide information about the persistence of attacks over time.
- **Average number of botnets attacking a domain per week:** This metric is

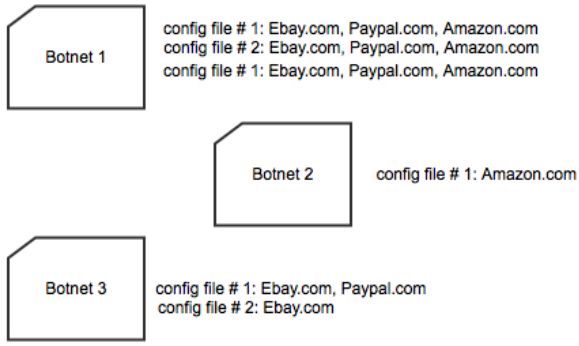


Figure 6.5: Illustrative example of why raw counts of attacked domains are not reliable as metric

basically a combination of the previous two. This metric eliminates some of the limitations of the raw counts by normalizing the data: it merges all configurations for a single botnet sent during a week and then counts the number of botnets attacking a unique domain in that week. To compare over longer periods, one could add up the counts for each week ('botnet weeks') or average them. The formula below displays how the metric is calculated when being averaged across n weeks.

$$Average\ number\ of\ botnets\ attacking\ a\ domain\ per\ week = \left(\sum_{k=1}^{n} botnets\ attacking\ domains\ in\ week(k)\right)/n$$

6.6 Relative Attractiveness of Targets

Having more reliable metrics at hand, we are going to look into the relative popularity of different targets along with the attackers' incentives behind this pattern. We do this by discussing three questions: (i) how are the attacks distributed across targets? (ii) what is the relation between target size and its popularity? Or to put it differently: do bigger targets attract more attacks? and (iii) how is the attack persistence distributed across different targets?

6.6.1 Distribution of attacks

In Figure 6.6, we rank the popularity of domains as attack targets using the number of botnets attacking a domain per week as a metric. The rank shows a highly concentrated pattern. The pattern is a power-law distribution, where 15% of the domains account for 90% of the attacks.

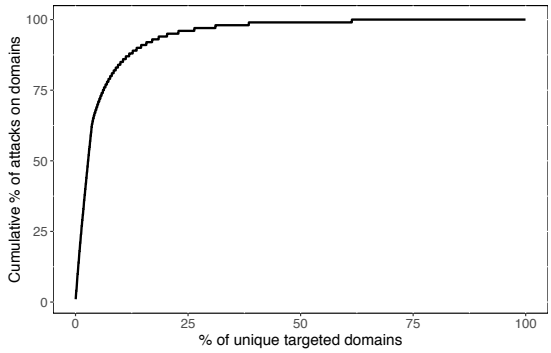


Figure 6.6: Cumulative percentage of attacks on unique domains (in botnet-weeks)(2009-2013Q1)

Target popularity can also be expressed by the number of weeks a domain is under attack or *attack persistence*. Figure 6.6 shows the domains distributed over the number of weeks they were under attack between January 2009 and March 2013. Here, too, we see a highly skewed distribution: some domains are attacked only briefly, while others remain under attack for the whole period of our dataset. A small number of domains (88) were always under attack for the whole 216 weeks. A much larger group of domains (1,108) are under attack for four weeks or less. Finally, 1,216 domains fall between the two extremes: occasionally and often-attacked domains.

One interpretation of Figure 6.7 could be that the range of potential attack targets for criminals is wide: the fact that some of the attacks are short-lived might indicate trial-and-error on the part of attackers, i.e., the attacks are not successful or don't prove attractive. We will revisit this idea in section 6.7. An alternative explanation might be that some of these domains are attacked for a specific purpose, i.e., as a part of targeted attacks.

We categorized domains in Figure 6.7 into the different groups (see Ta-

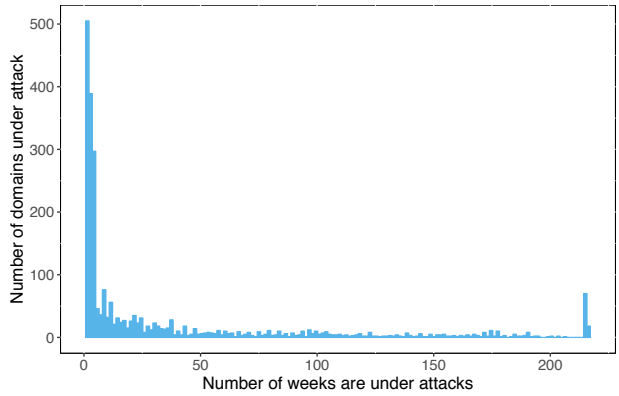


Figure 6.7: Attack persistence on domains attacked by Zeus malware (2009- 2013Q1)

ble 6.3)[2]. What stands out is the fact that the portion of financial service providers as targets increases with attack persistence, while the number of countries decreases. Here, the attackers reveal their core business: domains in always-attacked category are located in a compact set of countries, most notably Spain, U.S., U.K., Italy, Russia, and Germany.

Table 6.3: Domains grouped by attack persistence

Group	Defitition	Domains	% Banks	Countries
Briefly-attacked	Active for 1 month or less	1,108	54%	81
Occasionally-attacked	Active between 1 and 1.5 months	571	80%	58
Often-attacked	Active between 1.5 and 48 months	645	94%	46
Always-attacked	Active for 48 months	88	91%	13

6.6.2 Size and attractiveness for U.S. banks

The concentrated patterns of the attacks raise an obvious question of incentives: why do so many attackers go after the same cluster of targets? Routine Activity Theory (see section 6.2.3) identifies four factors that drive target selection: value, portability, visibility, and accessibility. At this moment we cannot

[2]The percentage of financial institutions is calculated using the Alexa categories for the sites that are categorized by Alexa - the numbers do not include the uncategorized domains in each group

systematically assess portability or accessibility, though the configuration files do contain inject code aimed to bypass two-factor authentication mechanism of domains, which might tell us more in the near future. Value and visibility can however correlate with the size of the attacked financial service providers, as measured by its customer base and the wealth of those customers. In other words, are the largest providers in the richest markets attacked more? The logistics of malware-based attacks seems to favor banks with a large customer base, as it increases both the odds of finding infected customers as well as spreading out the costs of developing the inject code over more attacks (similar to the reason why most malware writers target Windows-based machines).

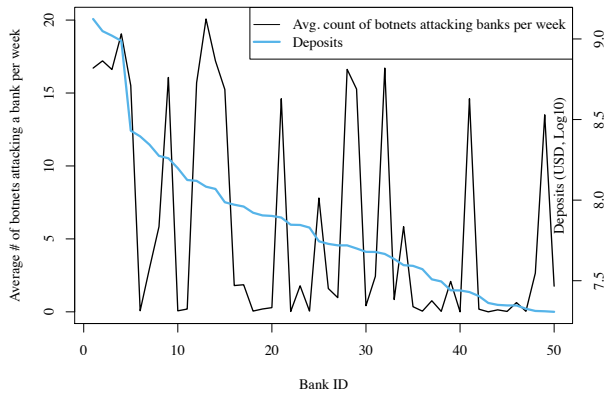


Figure 6.8: Deposits (top) and Zeus attack intensity (bottom) for the top fifty U.S. financial institutions

We have crudely estimated the size of a target via its web traffic volume (Alexa ranking) and, for the U.S. financial institutions, via FDIC data on the volume of deposits (see section 6.3.5 for more details). The FDIC lists U.S. banks and financial institutions with some of their financial and administrative properties, including assets, deposits, and net-income. In Figure 6.8 we have plotted the top fifty U.S. banks – in terms of their deposits[3] against the average number of botnets attacking these banks per week over our observation period

[3]We prefer deposits over assets as a measure of size, to distinguish between financial institutions that might provide mainly mortgages and have little banking and saving services;

(2009-2013Q1). The blue line in the graph describes deposit per institution on a log-scale; the black line indicates the average number of botnets attacking all domains[4] of that institution per week. Clearly, the two variables do not maintain a strong relationship.

However, mapping the attacked domains to the FDIC list yields to two interesting points. The first relates to the fact that out of around 6,500 active US-based financial institutions, only 175 have been targets of Zeus attacks in our data. Assuming that our sample is representative (see section 6.9), this is a surprising low number. Almost all of the largest banks (48 of the top 50) are attacked and present in the dataset[5]. The situation for the smaller banks is completely different. This might be caused by the fact that many smaller banks in the U.S. either did not provide online banking services or have outsourced these to a smaller number of third parties who may or may not be among the attacked domains. Another explanation is that they simply are not attractive targets for Zeus-based attacks, given their small customer size or limitations on how and where stolen funds can be transferred.

Therefore one can claim that whether a bank gets attacked is related to its size; above a certain threshold, a bank becomes a target. Beyond the threshold, however, size no longer seems to be a factor. The intensity of attacks is hardly related to size: the average number of botnets attacking each week fluctuates from less than 1 to 20. The result of the regression analysis is an adjusted $R-squared$ of 0.25 (N=50, F=0.00) – a weak correlation. This clearly suggests that there are other factors driving target attractiveness than merely the size of bank or its customer base.

6.6.3 Size and attractiveness worldwide

Does the pattern we observed for the U.S. banks hold across the whole population of targets? We use the Alexa rank as a proxy for the traffic size, with lower rank numbers indicating more incoming traffic, i.e., more users. Figure 6.9 on the left, shows the relation between Alexa rank of a domain and attack persistence. On the right, we have included box plots of the Alexa ranks for the different persistence groups discussed in section 6.6.1.

these will have high assets but low deposits, and so far have been of less interest to botnet-based forms of attacks.

[4]Some of the institutions have multiple web addresses and domains. We aggregate the attacks on all of these related domains in this figure.

[5]The two missing banks offer online banking via a third party. These sites handle online banking services for multiple banks. They are both present in our dataset, but attacks on them cannot be attributed to the individual banks.

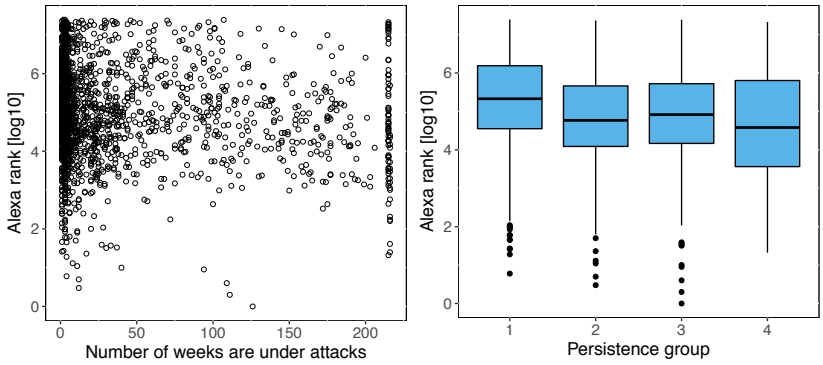


Figure 6.9: (Left) Scatter plot of Alexa rankings vs. attack persistence; (Right) box-plot of Alexa rankings per different attack persistence groups (only financial institutions, log-transformed. Lower rank number indicates a site has more visitors).

The result of Spearman's rank correlation test indicates a weak but significant negative correlation between Alexa lower ranks (domains with more visitors) and the attack persistence, i.e., the number of weeks they are under attack (rho=-0.13, σ=0.00, N=1,995). The results are similar if we rerun the correlation only for financial institutions (rho=-0.19, σ=0.00, N=731). The correlation is driven mainly by the difference between the ranks of the briefly-attacked domains and the always-attacked domains. The result of Kruskal-Wallis test for comparing sample means presented in Table 6.4 supports this. The same test does not find a significant difference in the Alexa traffic ranks of domains that are briefly-attacked and occasionally-attacked, nor between those that are occasionally-attacked and often-attacked.

Table 6.4: Kruskal-Wallis test results comparing means of Alexa ranks among different target groups

	Persistence group	N	Mean Rank
Alexa rank	1	866	1112.76
	2	479	889.73
	3	571	936.38
	4	79	841.88
	Total	1995	

Test Statistics : a,b	
	Alexa rank
$chi - Square$	63.626
df	3
α	0.000

a. Kruskal Wallis Test
b. Grouping Var: Pers. group

In short: the size of a financial service provider seems to influence target selection mostly in terms of a threshold: providers above a certain size are much more likely to be targeted. Beyond that threshold, however, size does not really seem to impact attack intensity. Within the top 50 of U.S. financial institutions, we see large differences in attack volume. The same holds for the wider group of larger targets. So far, it is unclear what other factors are at work here.

6.7 Seeking New Targets

A substantial number of domains (1,108, about half of the total) were targeted for four weeks or less in the four-year period (see Table 6.3). Taking a deeper look into the domains in that group, we realize that this group stands out from the others by its diversity: from the largest to the smallest domains, spread out over 81 countries and multiple industries, only half of which are financial services. This diversity makes sense if we interpret it as the result of a process of trial-and-error by the attackers. A new target is chosen for attack. The attacker identifies the relevant URLs, develops the inject code, and pushes the new configuration to bots under her command, and waits for victims. If the attack is successful, the attacker will persist. If, however, within a few weeks and a handful of attempts, the attack is not successful, the attacker has to decide how long to keep incurring costs or lack of benefit before moving on to a different target. The lack of success might be caused by effective defense measures by the targeted institution or its users. It might also be the case that the attack was technically successful, but the value of the loot – e.g., the price that the underground market was willing to pay for the captured data – did not merit further attacks.

The rate of trial-and-error is reflected in the number of new domains that show up over time. New domains are being tried all the time however, with peaks now and then: on average 119 domains are new each month – either never attacked before, or briefly in the months before the last. The overall number of domains getting attacked per month seems to be remarkably constant with a clear ceiling. An average of 601 domains each month become targets of Zeus attacks (σ=172, CV= 0.29). This is across all botnets in our dataset.

The stable ceiling on the number of targets pursued simultaneously is surprising, given the ongoing development of malware-as-a-service, which supposedly reduces entry barriers and would attract new attackers. It suggests there are bottlenecks elsewhere in the criminal value chain. With money mules, for exam-

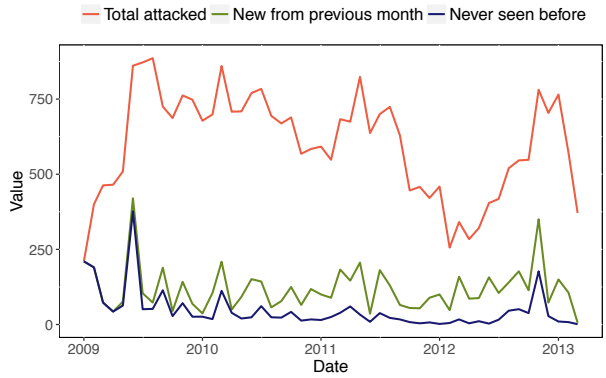


Figure 6.10: Trial of new Zeus target domains per month (2009-2013Q1)

ple, there appear to be no economies of scale [203]. In fact, the security architect of a multinational bank recently told us that they work on the assumption that recruiting mules is the most expensive and vulnerable resource for the attackers. The bank changed its defense strategy based on this insight: when a fraudulent transaction was detected, it was not blocked, but simply flagged and tracked until it is was completed and the mule had been revealed. Only then was the attack terminated. This meant that the attacker had to burn through his scare resource, mules, without knowing the odds of success.

The ceiling remained in place even after the Zeus source code was leaked and became widely available around May 2011. Several security firms predicted that this would increase the volume of attacks, as the leak would depress prices of Zeus-related services in the underground economy and further reduce the entry barriers for new attackers [204, 205].

Table 6.5: Number of attacked domains and active botnets before and after Zeus code leakage

	# attacked domains	# active botnets
Only before May 2011	786	1,334
Before and after May 2011	949	87
After May 2011	519	712

Our results however do not support this prediction, even though there is

hardly a shortage of potentially profitable targets[6]. Table 6.5 summarizes the point: the last row in the table indicates the number of new domains that were targeted only after the code leak, and the number of new botnet-keys that were activated. The numbers are lower than those for the period prior to the leakage. If we normalize these counts per month, compensating for the fact that the earlier period lasted a bit longer (28 vs. 22 months), the rate before the code leak was 48 botnets per month vs. 33 afterwards (if we leave out the botnets active in both periods).

The lack of growth in the population of Zeus targets resembles the phenomenon discussed by [206]: the majority of users go unharmed each year, despite the claims of security experts that many attacks are getting cheaper and easier. One of their explanations is victim diversity: if the fraction of all users who succumb to a certain attack is too small then the entire attack is rendered unprofitable. This is especially true when the gains per victim are unclear[7].

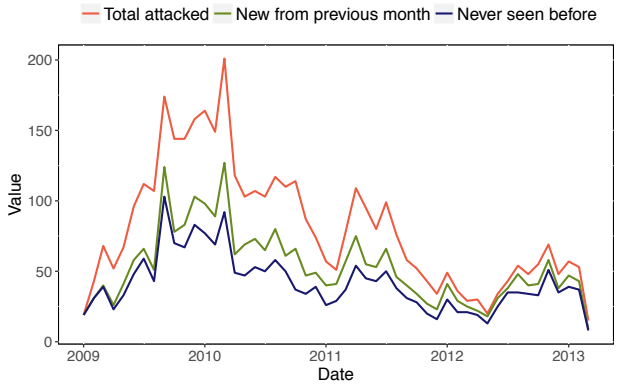


Figure 6.11: Entrance of new Zeus attackers (botnet) per month (2009-2013Q1)

The relatively stable pattern also belies another claim: that the Microsoft-coordinated take-down effort of Zeus command-and-control infrastructure – co-denamed B71 – had a noticeable impact on the operations of the attackers.

[6] Others also reported the absence of a rise in attacks [203].

[7] An exception to this rule is when the attacker has information at hand showing that the victim is actually valuable. This might explain the existence of banks from the micro-states in our datasets, given the guess is that these are used for instance tax evasion or similar purposes, and belong to wealthy people.

Microsoft never claimed that have fully disrupted Zeus, but rather a "strategic disruption of operations to mitigate the threat" [201]. Within the population of botnets that we have tracked over the course of four years, no such disruption is visible. Although there was a temporary dip in activity around the time of the take-down, March 2012, the decline towards that low has started well before operation B71. In fact, briefly after the operation, botnet activity started to rise again to previous levels.

Table 6.6: Botnets categorized by their lifetime

Lifetime category	Definition	# Botnets (RC4 keys)
Botnet lifetime 1	Active 1 day	1,315
Botnet lifetime 2	Active between 1 and 30 days	272
Botnet lifetime 3	Active between 30 and 105 days	272
Botnet lifetime 4	Active equal or more than 105 days	274

We also took a look into the relationship between botnet lifetime and the different groups of targets in terms of persistence of being attacked. To do that, we first categorized botnets in terms of their lifetime into four different groups (see Table 6.6). The first group contains botnets that were only active for one day and we treated them as a separate group. For the rest, we divided the total number of botnets or RC4 keys into groups of almost equal size.

Table 6.7: The number of botnets with different lifetime in different attack persistence categories

		Attack persistence			
		Briefly	Occasionally	Often	Always
Lifetime 1	Count	62	209	657	1235
	Expected count	101.121	31.342	736.510	107.025
Lifetime 2	Count	23	89	217	241
	Expected count	27.537	86.146	200.566	292.750
Lifetime 3	Count	32	121	236	250
	Expected count	29.397	91.965	214.113	312.524
Lifetime 4	Count	70	166	252	262
	Expected count	28.943	90.545	210.809	307.701

Next, we looked at the relationship between botnet lifetime[8] and the attack persistence category that they attacked. Table 6.7 shows a cross table attack

[8]Lifetime of each botnet or RC4 key is calculated by subtracting the first and last time

persistence and botnet lifetime of all botnets. As it is clear from Table 6.7, more botnets attacked domains that are located in the always-attacked category rather than the domains that are located in the briefly- attacked category. Moreover, most of the attacks on domains that are briefly attacked are performed by botnets with the longest lifespan. We hypothesize that those might belong to the most professional attackers, who are able to keep the botnets up and running the longest. The attackers in this category also do the most trial and error, which fits with the hypothesis that they are also the most capable. The number of botnets in this cell is higher than the expected value (observed count: 70, expected count: 28.94)[9].

Finally, upon investigating the country of the attacked domains, the pattern seen in other graphs is confirmed. Figure 6.12 shows the overlap between the targeted countries over the course of four years (2009-2012). Out of the total 92 attacked countries, seven were only attacked in 2009, and seventeen only in 2012. This shift in the variety of the attacked countries, despite the overall stability in the size of the attacks, points to a trial-and-error process with finite resources and players; i.e., the attacks are not spreading like mushrooms.

Figure 6.12: Venn diagram showing the overlap among the countries of the attacked domains in different years

that the key is seen. This has strong ($rho = 0.97$) significant correlation with the number of weeks that a botnet is active.

[9]It should be mentioned that cells in this table are not independent and therefore the applicability of chi-square expected value is limited.

6.8 Attack Code Development

6.8.1 Descriptive analysis

Our dataset contains 1,146,860 target URLs with associated inject codes. These inject codes are by no means unique. In fact, on average each inject code is repeated 27 times. Figure 6.13 shows the number of times a specific piece of code is used in different configuration files. Note that virtually all inject codes are reused two or more times among the different configuration files.

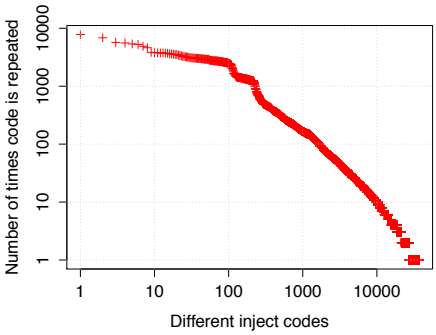


Figure 6.13: Number of times an inject code is used among different configuration files

Just 5% of the inject codes was re-used less than 10 times. More striking is the fact that 43% of all inject codes was repeated over 1,000 times. It is a safe bet that any new configuration file found in the wild will contain some inject codes that are identical or trivially different of a previous file. Just 9,679 inject codes (1.19% of the total amount) that were not repeated in any other configuration file. Even in this group, the bulk consisted of slight revisions of previous code aimed at the same URL.

This high amount of code repetition is intriguing. Attackers have clear incentives to reuse old code: it is more cost effective to make incremental modifications and reuse the same proven configuration files than to develop new ones from scratch. The modifications are necessary to evade new security measures. What is puzzling is that the attackers can get away with this little effort. Despite many countermeasures that have been proposed [207], malicious users

continue to use the same inject code. This pattern holds both before and after the code leakage.

To acquire a better understanding of these repetitions we analyze the amount of code lines per attack. Results show that the average length of an inject code is around 36 lines (with a standard deviation of 76.8 and coefficient of variation of 2.1). While there are complex attacks with more than 1,000 lines, they represent only 0.05% of the total number of attacks. The majority of the inject codes (56%) range between 10 and 100 lines. These are illustrated in Figure 6.14. The high deviation in the number of lines gives an idea of the complexity and diversity of the attacks. Looking at the attack codes, we see a wide range of ideas, from a simple rendering of a page element suggestion to the user to install an older (i.e., more vulnerable) web browser, to larger inject codes containing actual scripts to grab personal information.

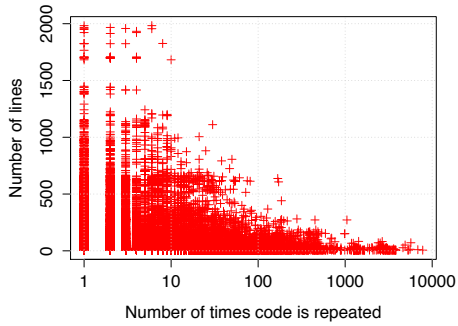


Figure 6.14: Number of lines per attack code

6.8.2 Similarity analysis

To more systematically analyze the level at which attackers re-use existing code, we applied a comparison technique to the whole dataset. Different text comparison techniques can be applied to find discriminating features of different codes. Our approach is an adaptation of text retrieval matching using the so-called Term Frequency-Inverse Document Frequency (*tf-idf*) methods [208]. These techniques have been widely used for comparing different malware (See e.g., [209, 210, 211]), and for detecting plagiarized documents [212]. We also

report the differences between two inject codes, expressed as a minimal list of line changes to bring either file into agreement with the other in relation to the total number of code lines [213].

First, we processed the configuration files to extract the inject code and conform the strings. Then, we tokenized the symbols found using the classic separators (e.g., dot, comma, colon, semi-colon, blank space, tab, etc.). In order to represent a string collection, a common approach in text comparison is to use a Vector Space Model, which represents documents algebraically, as vectors in a multidimensional space. This space consists of only positive axis intercepts. After that, we constructed a text representation of an inject code, which is formed by words s_i, such that $\overrightarrow{C} = (s_1, s_2, \cdots, s_n)$, n being the number of words within the code. We defined the weight $w_{i,j}$ as the number of times the word s_i appears in the inject code $\overrightarrow{C}_j$; if s_i is not present in $\overrightarrow{C}_j$, $w_{i,j} = 0$. Therefore, any attack code $\overrightarrow{C}_j$ can be represented as the vector of $\overrightarrow{C}_j = (w_{1,j}s_1, w_{2,j}s_2, \cdots, w_{n,j}s_n)$. Finally, we used *tf-idf* weighting schema, where the weight of the ith word in the jth injection code, denoted by $w_{i,j}$, is defined by:

$$w_{i,j}{=}tf_{i,j}{\cdot}idf_i{=}\frac{n_{i,j}}{\sum_k n_{k,j}} \cdot \log\left(\frac{\beta}{\gamma}\right)$$

where $n_{i,j}$ is the number of times the word s_i is not present in $\overrightarrow{C}_j$, $\sum_k n_{k,j}$ is the total number of words in $\overrightarrow{C}_j$, β is the number of codes being compared and γ is the number of codes under comparison that contain the word s_i.

Tf-idf method is based on vector similarity over dampened and discriminatively weighted term frequencies. In our case, we chose the cosine similarity that has proven to be a robust metric for scoring the similarity between two strings [214]. The basic idea behind cosine similarity is to transform each string into a vector in some high dimensional space such that similar strings are close to each other. The cosine of the angle between two vectors is a measure of how "similar" they are, which in turn, is a measure of the similarity of these strings. If the vectors are of unit length, the cosine of the angle between them is simply the dot product of the vectors. Thus, two attacks codes are more similar if they contain many of the same terms with the same relative number of occurrences of each.

Having defined the similarity metric, we used it to compare consecutive attack codes per URL. Figure 6.15 shows the average similarity per URL. More than 83% of the inject codes targeting a particular URL are more than 90%

similar, and only 1.71% of the inject codes are very different (less than 50% similar). On average, across all Zeus botnets and attackers, code similarity is over 90% from one attack to the next. This suggests some mechanism of code sharing or stealing among the attackers.

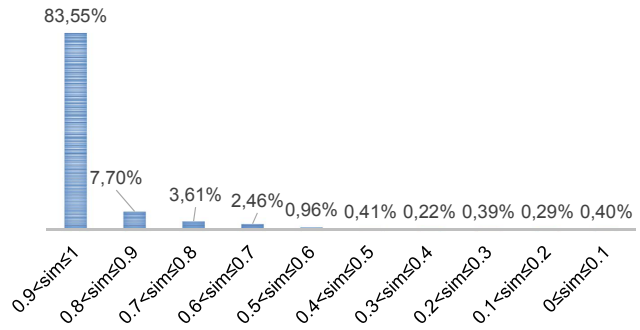


Figure 6.15: Average code similarity per URL

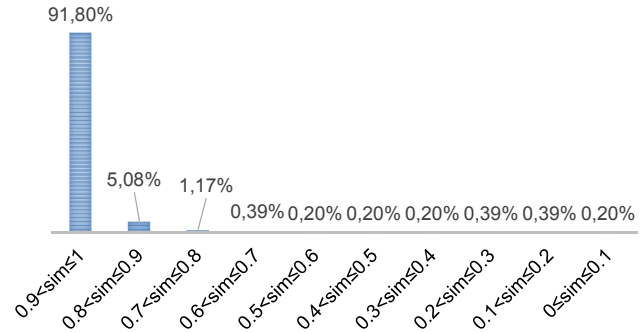


Figure 6.16: Average code similarity per botnet and URL

If we also take a look a particular URL per a particular botnet, we see that the similarity between consecutive codes increases even more, reaching 97% in average (see Figure 6.16). A botnet attacking a particular URL rarely changes the inject code between consecutive attacks.

The high similarity between consecutive attacks could be due to (i) incentives of attackers to not change the code substantially if unnecessary, and (ii) operations related to essential characteristics of the targeted URL.

Next, we analyze the impact of code length on similarity. In general, a clear trend does not appear between the two. As Figure 6.17 shows, large codes are repeated less than small codes, but when repeated, the inject code is more than 97% similar – though this partially reflects the size of the inject, of course. In smaller injects, changing a few lines will drive down the similarity. In any case, consecutive code attacks to the same URL are more than 90% similar in average no matter their length.

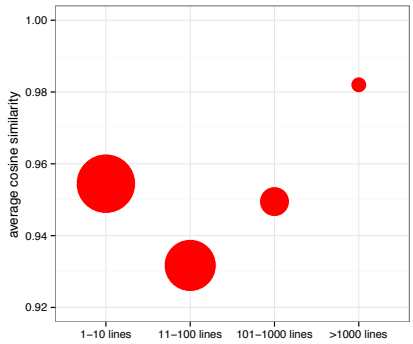


Figure 6.17: Average code similarity versus code length and repetition (size of the bubbles)

We took a closer look at the evolution of attack code for four URLs among the top attacked domains (Figure 6.18). These URLs are from PayPal, HSBC, Bank of America, and Alliance Leicester. Besides the cosine similarity, we also compare the number of different code lines between consecutive inject codes. Both metrics follow the same pattern for all the URLs. As one would expect from the analysis so far, the inject codes in most of the instances is the exact copy of their predecessors. However, we can observe that the similarity metric drops at certain points in case of some URLs. These drops most probably reflect changes in the domain's webpage and defense measures by domain owners (financial service providers and other industries) that forced the attackers to adapt their code. In either case, it can be seen that after each drop in similarity, the next codes again become similar. Similarity drops vary in different levels for each of the URLs, reflecting the amount of change. Among these examples, a particular PayPal URL suffers from the most abrupt changes in the similarity metric, while the HSBC inject code's similarity only drops below 65% one time, with most of the consecutive attacks remaining identical.

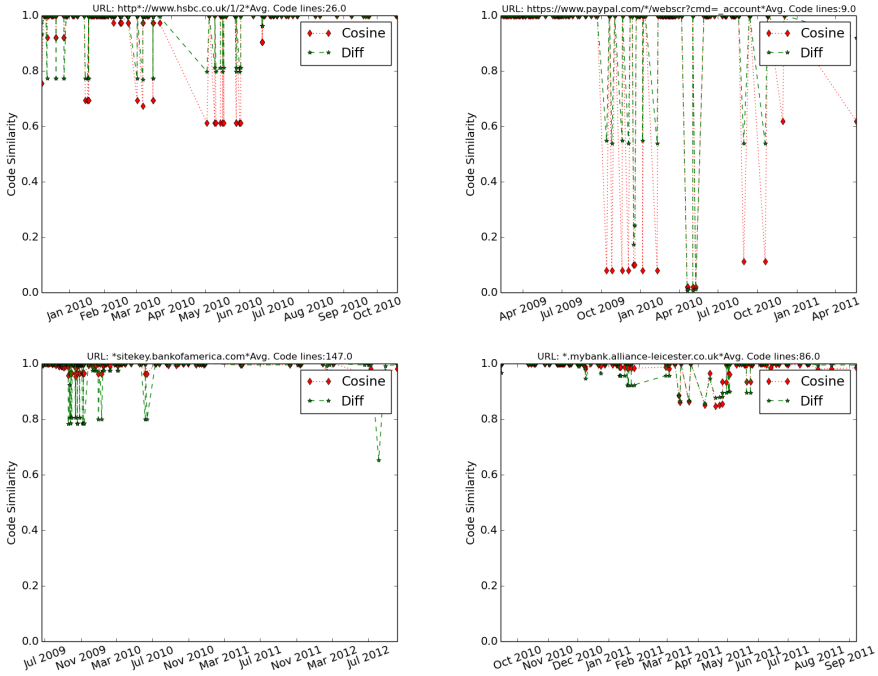


Figure 6.18: Code similarity evolution for 4 URLs corresponding to the top attacked domains

In summary, inject code is not updated with high frequency. This somewhat contradicts the anecdotal descriptions from the industry of highly dynamic cat-and-mouse games with constant adaptations between attackers and defenders. The attack activity observed in this period clearly shows that copies of a previous used inject code and also inject code that only marginally differs from previous ones are present in most configuration files.

In addition, if we take a closer look to the inject code similarity between botnets with different lifetime against targets in each persistence group (6.8), we realize that in most of the cases code similarity increases in line with botnet lifetime, i.e., botnets that are active longer tend to have smaller code changes. It is unclear what this means exactly. If we assume that more rapid code evolution is a sign of attacker competencies, then this finding provides evidence against our

earlier hypothesis that the longer-lived botnets are run by the more competent criminals. An alternative interpretation, however, is that these attackers write more robust code that requires fewer changes to stay functional.

Table 6.8: Cross table of inject code similarity for botnets with different lifetime in different attack persistence categories (code similarity between 0 and 1)

Code similarity	**Attack persistence**			
	briefly attacked	occasionally attacked	often attacked	always attacked
Lifetime 1	0.893	0.897	0.951	0.926
Lifetime 2	0.959	0.968	0.970	0.968
Lifetime 3	0.965	0.970	0.968	0.969
Lifetime 4	0.966	0.973	0.971	0.972

In general, the overall pattern of high code reuse indicates that financial providers are not implementing changes that require rapid adaptation on the side of the attackers. The lack of code development also suggests that the cost of continuing these attacks is limited. Less skilled attackers could enter the market and survive on minor modifications of existing inject code. That we haven't seen an increase in the volume of attacks is further evidence for the earlier finding that low entry barriers to the criminal market do not translate into attack volume.

6.9 Limitations

As with all studies of real-world applications and implementations, we should reflect on the potential impact of possible measurement errors in our data and other limitations. We already discussed the limitation of using RC4 keys as a proxy for botnets and attackers. Here, we focus on the possibility of biases in the collection method. Perhaps the honeypot network is less likely to capture attacks against certain domains or countries, or the captured and decrypted files are biased towards the less competent attackers.

To check how representative our sample of Zeus configuration files is, we crosschecked our data with Zeus Tracker data from Abuse.ch, a well-known anti-Zeus initiative. We compared the C&C domains from which the configuration files were present in both of the datasets for the specific period that Zeus Tracker published this data (Sept 2010 to March 2013). First of all, there is a difference in volume. Where Zeus Tracker published 18500 C&C domains, our dataset identified around 27,400 C&C domains for the 11,000 files (and more than 30,000

C&C domains for the overall set of around 144,000 captured files). It does seem that the Fox-IT collection method is more comprehensive. Out of the 18500 C&C domains listed by Zeus Tracker (live or removed), 4550 of them overlap with C&C domains in our dataset.

In addition, as mentioned in section 6.3.1, the data was gathered via honeypots, predominantly located in European countries. This might have introduced a geographical bias. Indeed, in the next section we see that the European countries are more often targeted than those in the U.S. or Asian targets. This does not per se imply, however, that there is a bias in the data. Our industry partner argues that the attackers did not differentiate among the infected clients based on their geography directly. The dominance of Europe targets may reflect the fact that online banking services have been offered more widely by European providers and have been adopted more comprehensively by consumers. Also, many European banking systems have near real-time transaction processing, which makes it easier for criminals to move money out of the account before anti-fraud operations can stop it.

The situation in the U.S. is rather different. The market is less consolidated, with thousands of smaller banks, not all of them offering online banking facilities. Furthermore, of the banks that offer online services, not all of them provide cross-bank transferring capabilities, making it harder for attackers to cash out funds. Sometimes there is a need for in-person validation of the receiving account beforehand.

We did notice that Asian countries seemed underrepresented. That might indicate a sampling bias or it might reflect a different attack history. For example, the Zeus-variant Citadel has been reported to have started targeting Japanese banks only late in 2013, which is outside our observation period [215]. Another explanation is that Asian attacks were predominantly executed via other malware families, such as KRBanker. Finally, we should also mention that only a small percentage of configuration files in our data belong to the newer Zeus variants, which is understandable since many of the new variants have become active in the course of 2012.

6.10 Conclusions and Discussions

Financial malware on home computers and mobile devices causes millions of Euro in damages each year. Not every financial service provider is equally popular among cybercriminals. Why are some financial service providers targeted more often than others? There is very little comparative empirical research

across providers and countries identifying the factors that contribute to the selection of financial service providers as targets.

This chapter sets out to explore the incentives and strategies of attackers from the instructions – contained in configuration files – sent to the machines that were infected with Zeus malware from 2009-2013Q1, during which period it was one of the dominant financial malware families. We investigated around 11,000 configuration files targeting 1.2 million URLs, which consisted of 14,870 unique URLs on 2,412 unique domains. We identified the attacked domains, which include financial services as well as other targets, and developed metrics to rank their relative attractiveness as a target, taking into account how criminals update the instructions for the bots under their control.

The attacks were concentrated: around 15% of the domains attracted 90% of the attacks. The concentration is not driven just by target size. Using financial data from FDIC and traffic rankings from Alexa as proxies for the size of the payment service provider, we observe that size is a threshold for getting attacked, but does not predict the intensity of attack. Attack persistence varies widely, with around half of the domains targeted briefly (4 weeks or less), and 88 domains targeted during the whole period (216 weeks). We believe the brief attacks are part of a process of trial-and-error of attackers seeking new targets. Looking into it from the perspective of botnets, we realized that long-lived botnets are more probable to attack domains is this category comparing to the short-lived ones, which again supports the idea of trial-and-error.

Surprisingly enough, even though new domains are being tried over the whole period, there seems to be a ceiling in the overall number of domains being attacked simultaneously. This suggests bottlenecks elsewhere in the criminal value chain, for example, in the recruitment of money mules or in the involvement of the attackers in other stages of the attack (e.g., the need to take over banking sessions in real time). Despite what is expected, the ceiling remained in place both in terms of number of domains that were attacked and in terms of number of new botnets that entered the market even after when Zeus source code was leaked and became widely available. This suggests that in this market, low entry barriers do not translate into more crime.

We also studied the evolution of inject code over time. Using a cosine similarity metric, we compared the 1.2 million inject codes in the dataset. In short, the vast majority of the inject codes were merely modifications of previous codes. In fact, only 3,664 attacks were not exact copies of a previously seen code. In any case, consecutive code attacks to a same URL are more than 90% similar regardless of the length of the attack code. This suggests that attacks are much less dynamic than often presumed.

Code gets re-used to a remarkable degree: just 1% of the inject code is never repeated, and 226 different inject codes are repeated over one thousand times without any modifications. On average, across all Zeus botnets, code similarity is well over 90% from one attack to the next. This suggests some mechanism of code sharing or stealing. Within a same botnet, similarity goes up to 97%. Overall, it seems that cost of code development for attackers is limited. This could lower entry barriers and increase attacks, but as we found earlier, this does not occur. Entry barriers are not the factor that is keeping attack levels in check.

At a more general level, the implication of these findings might be that the underground markets for infected machines, malware-as-a-service, which have been portrayed as making attacks cheaper to execute and even as opening up cybercrime to the unschooled masses, are not main force in driving the attack volume, nor the selection of targets. This suggests that there is a need for more investigation on other parts of online banking fraud value chain such as money mules, or banks' money transferring policies. If the bottlenecks are not in the malware ecosystem, then investing in disrupting the ecosystem by defenders and law enforcement may not actually be the best allocation of scare resources.

CHAPTER 7

Measuring the Impact of Providers' Proactive Security Efforts on Abuse

The previous chapters studied different parts of the causal model: incidents, exposure, and attacks. This chapter aims to quantify a final causal link, namely between security/vulnerability and incidents, while controlling for exposure. In other words, to what extent do different proactive security measures taken by webmasters and hosting providers impact abuse rates in a shared hosting environment? The study measures security efforts by drawing on a diverse set of security and software features collected at scale from a large sample of domains. In the first part of study we use the features to estimate underlying latent **factors** *that capture different types of security effort. We then estimate which of these factors are influenced by provider efforts versus the efforts of their customers. In the second part of this study, we construct multiple statistical models to quantify the impact of each factor on malware and phishing abuse observations, identifying the control points that providers can influence to fight against abuse. In sum, this study comprehensively models the main relations of the causal model underlying this thesis.*

7.1 Introduction

Global web infrastructure is compromised at scale in support of a myriad of cybercrime business models, from phishing to botnet command and control (C&C) to malware distribution. The responsibility for remediating compromised resources is shared between webmasters and multiple infrastructure operators, notably hosting providers, domain name registrars and internet service providers (ISPs). The important role of hosting providers is codified in best practices from industry organizations such as M3AAWG and SANS [20, 216, 9].

These guidelines encourage providers to take sensible steps, such as keeping customer software updated.

When the defenses fall short and resources are compromised, providers are regularly faulted for not doing enough to forestall compromise (e.g., [10, 34]). This raises the question, however, of what providers can realistically achieve in terms of preventing abuse. Compromise rates are driven by many factors outside the immediate control of providers, not least of which is the security decisions and patching practices of their own clients [217, 218]. It is this joint responsibility between providers and webmasters that makes answering the question so difficult. In this chapter, we provide an answer for the case of *shared* hosting, one of the most prevalent and affordable ways to publish web content in which many websites share the same server.

We focus on shared hosting services for several reasons. First, its customers operate under restricted privileges. Hosting providers maintain administrator privileges and can typically regulate what software is installed and whether it is updated. As acknowledged in M3AAWG's best practices, providers have the most control over, and hence most responsibility for, their resources in shared hosting plans, compared to other hosting services [9]. Second, even when customers can change configurations, shared hosting providers maintain a strong influence by provisioning default configurations that may or may not be the secure.

Put another way, if hosting providers can and do make a difference in improving security, we would expect to find evidence for it in this segment of the market. Third, this segment matters in the overall scheme of web compromise. Shared hosting is associated with especially high concentrations of abuse [122, 49, 142]. In the data examined for this chapter, for example, around 30% of all abused domains were on shared hosting.

Another barrier to assessing provider efforts to prevent abuse is that their efforts cannot be measured directly. We cannot, for example, measure each provider's security budget, abuse team staff levels, or uptake of technologies to mitigate attacks. In economics terms, there is an inherent information asymmetry about the extent and efficacy of the security efforts undertaken by providers.

We overcome this barrier by adopting a new approach, adapted from psychometrics, that constructs an indirect measure of security effort by amalgamating a disparate set of observable features such as patching levels and secure web design practices. There are two key benefits of our approach. First, we do not presume ex ante if it is the webmaster or hosting provider who is responsible for these features. Who drives patching of Content Management Systems (CMSes), for example? Rather than make a priori assumptions, we answer these

questions empirically and thereby deal with the joint responsibility problem. Second, we do not presume a direct causal relationship between the observable features and how the website is ultimately compromised. For example, setting a Content Security Policy may not stop compromise, yet its presence does reflect the security efforts put into increasing website defences.

We make the following contributions:

- We present the first comprehensive measurement study of the population of shared hosting providers, revealing patterns in 15 indicators spanning domain security and software patching efforts, captured from a sample of 442,684 domains across 1,259 providers.
- We find that version hiding is a widespread hardening effort–e.g., 66% of admin panel installations hide version information. By contrast, indicators of domain security, such as `HttpOnly` cookie and `Content-Security-Policy`, are rare (13% and 0.2% of domains, respectively). Out of those with version information, most discovered installations of webservers and admin panels (87%) and (70%) were running unpatched versions. In stark contrast, CMS installations were unpatched in just 35% of cases. This perhaps reflects a difference in the probability of compromise between lower and higher levels of the software stack.
- We demonstrate a new statistical approach to empirically disentangle the contributions of different parties to a joint security outcome. Different from prior research, we do not make ex ante assumptions about the meaning of security indicators (e.g., that their configuration is under the control of the providers and accurately reflect their efforts). Instead, we use the indicators to induce latent factors that can be interpreted and empirically attributed to roles of responsibility. We then regress these factors on measurements of compromise, while controlling for exposure. This approach can be adopted to study other areas of joint responsibility, such as between cloud hosting providers and tenants, or corporate system administrators and end users.
- We find that webmaster and web application security efforts significantly reduce phishing and malware abuse. For example, the best-performing 10% of providers (in terms of web application security effort) experience 4 times fewer phishing incidents than the bottom 10% of providers. Moreover, we find that providers can influence patching levels, even for software running at the application level such as CMSes. The providers that do a

better job of patching their customers see reduced rates of compromise. This provides the first compelling statistical evidence of the security benefits of hosting providers adhering to industry best practices.

The chapter proceeds as follows: Section 7.2 explains the data and methodology used to sample domains, identify shared hosting providers, estimate their size, and measure compromise rates. Section 7.3 outlines the details of our active measurement setup and describes the effort-related features we collected. Section 7.4 presents an empirical view of the web security landscape in shared hosting. Section 7.5 discusses the reasoning behind why the collected features should not be used as direct causal explanations of abuse, highlighting the need for latent variables. Section 7.6 explains the statistical approach to estimate the latent variables and to empirically disentangle the contributions of different parties to a joint security outcome. Section 7.7, we assess the impact of the latent factors on abuse incidents. Section 7.8 discussed the limitations of our study and section 7.9 revisits related work. Finally, we discusses our main conclusions and implications in Section 7.10.

7.2 Data Collection Methodology

Shared hosting providers We start by populating a list of all domain names[1] and their IP addresses that were observed by DNSDB – a large passive DNS database[2] – in March 2016. Adopting the similar methodology explained in Chapter 3 and Chapter 4 leaves us with a set of hosting providers. Next, we mark a provider as a *shared* hosting provider if we observe at least one IP address that hosts more than 10 domains. We adopt the same threshold used in other studies (see Chapter 3). Using an elbow plot of domain density per IP address, we confirmed a sharp increase in density beyond a threshold of 10 to 15 domains per IP address. The result is a global list of 1,350 shared hosting providers.

Domain sample From the total set of 110,710,388 domains on shared hosting, we randomly sampled 500 domain names for each provider. We scanned them to verify these were still operational[3]. If fewer than 100 domains were up and

[1] We define domain name as a second-level or third-level domain, depending on whether the relevant TLD registry provides such registrations, e.g., `example.pl`, `example.com.pl`, `example.gov.pl`, etc.

[2] `https://www.dnsdb.info`

[3] Domains are sampled only from IPs marked as shared, since a provider can have shared servers next to dedicated ones

running, the provider was excluded from the list (91 providers were excluded). It should be noted that before drawing the random selection of domains, we dismissed around 4,000 parked domains, following the detection methodology outlined in [219]. This is specifically because a majority of parked domains are very similar to each other (share the similar content) and typically a single webmaster owns numerous parked domains, as indicated by Vissers et al. [219]. Therefore, if taken into account, the analysis is more likely to be biased towards a handful of website administrators owning a large number of domains. By excluding parked domains, we maintain an unbiased observation of the features that are related to the efforts of the webmaster. Accordingly, our final set contains 442,684 domains distributed over 1,259 hosting providers, located in 82 countries all over the world.

Size of hosting providers Shared hosting providers differ vastly in size, a fact to be controlled for when analyzing abuse with providers as units of analysis. Clearly, a million-site business is likely to observe more abuse than one with a few thousand customers. Unfortunately, there is no authoritative source for provider size. To estimate it from the available data, we use two different size indicators, each capturing a different aspect of the shared hosting providers. *Shared hosting IP space size* is the number of IP addresses hosting at least 10 or more domains. It is calculated by summing up all the IP addresses defined as shared, associated with domain names per provider that have been observed in the passive DNS data. The mean, median and maximum values are 636, 137 and 71,448 respectively, across providers in our sample. *Shared hosting domain space size* is the number of domains hosted on shared IPs by a particular provider. It is calculated as the sum of the domains that are associated with shared IP addresses of the provider, as seen in the DNSDB data. The mean, median and maximum values are 94,118, 10,233 and $3.3 * 10^7$ respectively, across providers in our sample. Note that due to a large variance and skewed distribution of the size variables, a log-transformation of these variables (base 10) is used in the regression analyses of Section 7.7.

Abuse data To estimate the compromise rate for each shared hosting provider, we used two abuse datasets. We extracted all entries that were associated with the shared hosting IP addresses of the providers and counted the number of unique domains per provider.

The **phishing** data is collected from two sources: the Anti-Phishing Working

Group (APWG) [4] and Phishtank [5]. Both datasets contain IP addresses, fully qualified domains, URLs of phishing pages, blacklisting times, and additional meta-data. For the second half of 2016, the data consisted of 62,499 distinct domains, which resolved to 47,324 IP addresses at the time of reporting. 49,065 of these domains were hosted by one of 968 shared providers in our study (The remaining 291 providers did not record any phishing during the period.)

We include drive-by-download **malware** URLs flagged by the Google Safe Browsing program, as reported to StopBadware[6]. For the second half of 2016, there were 362,069 distinct domains newly flagged with malware. Of these, 332,625 resolved to an IP address at the time of reporting. The rest was likely maliciously registered and already suspended. Of all resolvable domains, 97,872 were hosted by one of 1,050 shared providers in our study (The remaining 209 providers did not record any malware during the period.) The high proportion in both datasets underscores the importance of shared hosting in distributing web-based phishing and malware.

7.3 Measurement of Features

We aim to collect a wide range of features, composed of vulnerabilities and weaknesses, security mechanisms, and software patching practices, all of which can help us estimate the amount of effort going into securing domains.

We perform a large-scale measurement to obtain information from the 442,684 sampled domains. More precisely, we instructed our crawler, which is based on the headless browser PhantomJS[7], to visit up to 20 web pages for each domain. The list of web pages for a certain domain were obtained by following links starting from the home page, until either the maximum number of page visits was reached, or no further links could be discovered.

In order to restrict the feature collection process to the target domains, the crawler only considered web pages with the same second-level domain name. If, for example, the target domain `example.com` immediately redirects users to `website.com`, only a limited set of features could be obtained, i.e., server-level features and those based on response headers sent out by `example.com`. This was done to ensure that only information related to the website hosted in the shared hosting environment was considered. In total, it took our crawler, which

[4] `http://www.antiphishing.org`
[5] `https://www.phishtank.com`
[6] `https://www.stopbadware.org`
[7] `http://phantomjs.org/`

was distributed over 15 virtual machines, each composed of 4 CPUs and 4GB RAM, 7 days to visit and extract information from the 7,463,682 web pages.

We gather information to construct a list of 15 features, which is an extension of the web-based security features explored in prior work [26]. Our features give an indication of both security-related configurations, such as the deployment of `Content-Security-Policy`, and patching practices of various software such as CMSes, admin panels, PHP and SSH. Consequently, the captured features reflect security practices employed by both the shared hosting providers as well as the domain owners (webmasters) themselves. In the following sections, we briefly discuss these two groups. For the extensive list of features, please refer to Table 7.1.

Note that for most of the collected features, we do not expect to observe a direct causal relation on abuse practices. Instead, we consider the features to be proxies of the efforts made by the providers and webmasters. We discuss the limitations of treating these features as direct indicators of effort in greater detail in Section 7.5.

Ethical considerations. We have also assessed our work using the principles outlined in the Menlo report [220]. We do not collect data on persons. We designed our measurement techniques to be as unobtrusive as possible. We collected no more data than necessary and carefully scheduled our requests so that no single server could be overloaded. All features were obtained through passive observation and we added various countermeasures to prevent any irregular interactions with third party websites. Finally, we report the findings in an anonymized manner. We have also assessed our work using the principles outlined in the Menlo report.

7.3.1 Domain security indicators

As domains are prone to a large variety of potential vulnerabilities and weaknesses, the web security community has for a long time supported hosting providers and webmasters with mechanisms that enable them to apply a defense-in-depth approach. In this section, we discuss how we collect a multitude of security-related features to get an approximation of security efforts for domains.

Cross-site scripting (XSS) vulnerabilities are among the most critical security risks according to OWASP [107]. We look for the presence of the Content Security Policy response header, as it can be used to protect against XSS attacks. We consider a domain to have weak browser `XSS` protection if an administrator has disabled the default browser mechanism to detect and block reflected XSS attacks by means of the `X-XSS-Protection` response header. We

also check for the presence of `HttpOnly`, which helps reduce the potential consequences of XSS attacks, and `X-Frame-Options`, which can be used to thwart clickjacking. In addition, we check if the `Secure` cookie attribute and the HTTP `Strict-Transport-Security` response header are present, as they both can effectively improve transport layer security. Properly implemented web applications are also crucial. We define the SSL-stripping vulnerable form feature when a website has a form (e.g., on a login page) pointing to an HTTPS endpoint while being loaded over an insecure connection. Accordingly, the mixed-content inclusions happen when a website's content (e.g., JavaScript code, style-sheets, images etc.) is included over an insecure connection, while the web page was loaded over SSL/TLS.

Note that we indicate the direction of the features by (−) and (+) signs in Table 7.1 since not all features have a positive effect, such as mixed-content inclusions, SSL-stripping vulnerable form, and weak browser `XSS` protection.

7.3.2 Software patching practices

In addition to the security mechanisms discussed in the previous section, the act of patching server software and web applications plays a crucial role in the security posture of websites.

Often, attackers exploit known vulnerabilities present in unpatched software (e.g., vulnerabilities reported in the National Vulnerability Database [221]). Therefore, it is generally considered best practice for providers as well as webmasters to employ patch management mechanisms regularly and extensively.

Content Management Systems (CMSes) have been amongst the most exploited software stacks for many years [222, 43, 105]. Depending on the administration rights in the shared hosting environment, CMSes can be updated either by the webmaster or the shared hosting provider herself. In this chapter, we limit our scope to the CMSes with the majority of market share, namely WordPress, Joomla! and Drupal CMSes [223].

The presence and version number of these three CMSes are determined in two phases: first, a basic scan is performed using our crawler which tried to infer the version number from the `<meta name=″generator″>` HTML tag. However, as many CMSes allow hiding the version number, something that is generally considered a good practice against automated attack scripts, we perform a second, more comprehensive scan. For the comprehensive scan we made use of well-known industry tools such as Sucuri WPScan[8] and WhatWeb[9]. For the

[8] `https://wpscan.org`
[9] `https://whatweb.net`

latter, we updated the original scripts to allow us to incorporate the latest versions of the targeted CMSes.

In addition to the experiment that determines the presence and version number of CMSes, we performed a similar experiment that focused on admin panels, a type of technology that is innate to the shared hosting environment. In this chapter, we focus on the four most popular admin panels, namely cPanel, Plesk, DirectAdmin, and Virtualmin. We instructed our crawlers to visit the domains at the port numbers that are typically associated with the admin panels, e.g., port 2082 and 2083 for cPanel. We then improved our measurements by visiting the endpoints that we found to often be used as a shorthand to link to the admin panel, e.g., `/panel/`. Based on the response headers, HTML contents, and redirection chains that were captured by our crawlers, we tried to determine the presence and, when possible, the version number of the admin panels. This allowed us to obtain the version information for approximately 33% of the domains with admin panel in our sample.

Moreover, other components that contribute to the software stack, such as the HTTP server, SSH server and PHP interface, should also be treated as part of the threat surface. In this chapter we focus on Apache, Microsoft IIS and nginx for the HTTP servers. For the features related to the infrastructure of the web host, we inferred the version information through either the `Server` and `X-Powered-By` response headers (for webserver and PHP), or by the banner that was returned, e.g., the banner on port 22 for SSH.

Lastly, we look into SSL/TLS implementations, as they are important to prevent attacks on the transport layer. To assess weaknesses in the SSL/TLS infrastructure, we used sslyze [224]. The domain's SSL/TLS implementation is considered insecure when it was vulnerable to Heartbleed, supports old protocols (SSLv2, SSLv3), enables compression, or is vulnerable to CCS injection [225].

For all software where the version number could be determined by our scanner, we make the distinction between software that is patched and unpatched. Generally, we consider a software version to be *patched* if it was packaged in one of the supported versions of OSes with larger market share namely, Ubuntu, Debian, and CentOS [226] at the time of our the measurement (November 2016). This approach is relatively generous in considering software patched: patches are often backported to older versions; as we did not undertake any intrusive actions to determine the patch-level of software, no distinction is made between versions with or without these backported patches. Note that all the older versions of software packaged in OSes are deprecated and contain vulnerabilities. For instance, PHP version 5.3.2 had a vulnerability (CVE-2012-2317) that would allow attackers to bypass authentication in applications that would oth-

Table 7.1: Summary of measured domain security and software patching indicators in absolute and relative terms.

Feature	# of domains	% of domains
HTTP server	398,929	90.11
no version information	195,474	44.15
Patched version	58,818	13.28
Unpatched versions	144,637	32.67
SSL	288,018	65.06
Patched version	206,680	46.68
Unpatched versions	81,338	18.37
Admin panel	178,056	40.22
no version information	118,768	26.82
Patched version	17,949	4.05
Unpatched versions	41,600	9.39
PHP	156,756	35.41
Patched version	47,596	10.75
Unpatched versions	109,160	24.65
OpenSSH	130,146	29.39
no version information	716	0.16
Patched version	36,444	8.23
Unpatched versions	92,986	21.00
CMS	103,741	23.43
no version information	10,043	2.26
Patched version	61,457	13.88
Unpatched versions	32,264	7.28
`HttpOnly` cookie (+)	57,696	13.04
`X-Frame-Options` (+)	22,212	5.02
`X-Content-Type-Options` (+)	8,685	1.96
Mixed-content inclusions (−)	2,107	0.47
`Secure` cookie (+)	1,378	0.31
`Content-Security-Policy` (+)	894	0.20
HTTP `Strict-Transport-Security` (+)	847	0.19
SSL-stripping vulnerable form (−)	515	0.11
Weak browser `XSS` protection (−)	376	0.08

erwise be secure. This was then patched in the later versions packaged. A more recent example is CVE-2015-8867 in certain versions of PHP (5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12) [221]. A list of software and their patched versions is included in the Appendix.

Due to the automated nature of our experimental setup, the measurements may be subject to certain limitations. Despite the preventive measures we have taken to make the generated web traffic reflect the browsing behavior of a regular user, there could still be providers who will block our scanning attempts. Moreover, it is possible that certain software was not found within the scanning threshold due to hardening techniques. More specifically for admin

panels, if the software was not located at a default location, we would not be able to detect it. Furthermore, as we focus on a limited set of software, it is possible that a domain makes use of a different software stack, or that it was hand-constructed.

7.4 Descriptive Findings about the Landscape

Previous research has explored individual security features at the domain level. We now extend this approach in two ways: by combining these features with software patching practices and by moving from individual domains to the level of providers. What is the prevalence of security features across domains and providers? How patched are software installations? Do patching rates vary substantially from one provider to the next? Do different portions of the software stack have different updating behavior?

7.4.1 Distribution of security features

Table 7.1 presents a summary of the distribution of all security features, both positive and negative. The security features are presented as boolean variables, with 1 pointing to the direction of the variable. The first column indicates the total number of domains with a particular feature and the second column reports the percentage of all domains with this feature.

The overall pattern is clear. Across the landscape, although crucial, the positive security indicators have low to almost negligible adoption rates. Out of 442,684 scanned domains, `HttpOnly` cookie reaches a somewhat respectable 13%, but after that the prevalence drops quickly. Two features are present in less than 0.3% of all domains. The good news is that the observed negative security features that can result in vulnerabilities are equally sparse: Mixed-content inclusions is the most widespread at 0.5%.

To illustrate, Figure 7.1 displays the percentage of domains at a provider that have `Content-Security-Policy`, `HttpOnly` cookie or `X-Frame-Options`. At most providers, only a small fraction of their domains support these features, hence one sees rarely any large concentration of a feature within a group of providers. In fact, for 1,100 providers (95% of the providers we evaluated), fewer than 20% of their domains in our sample have `HttpOnly` cookie enabled. The exception is a group of 9 providers where 80% of the domains have `HttpOnly` cookie enabled, indicating a provider effort in the form of provisioning default secure configurations. To further validate this assumption, we tried to contact

this set of nine providers manually and check whether they provide certain security features by default. We have been contacted back by three of the providers. Two of the providers confirmed that depending on the customers, they might set `HttpOnly` cookie by default in the cases where they are the responsible entity for the customer's security. Another provider pointed out that the default `HttpOnly` cookie setting is a built-in feature in the `DotNetNuke` CMS they employ.

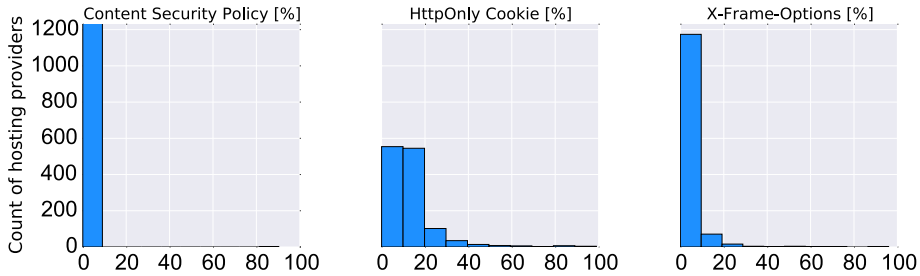


Figure 7.1: Distribution of security features over hosting providers

Note that the median and mean complexity of the webpages in our sample which we measured by the number of endpoints are 11 and 71.68, respectively. Having that in mind, we expect that some of the features under study are only useful in specific configurations, so widespread adoption is not to be expected. Not every page will set a cookie, for example, and not every cookie needs to have the `Secure` or `HttpOnly` attribute. A cookie might set a language preference, it might need to be accessible in JavaScript, and it does not matter if this leaks in a man-in-the-middle attack. Also, for `X-Frame-Options`, it makes sense that this header is only added on pages that are subject to clickjacking attacks. On the other hand, features such as `Content-Security-Policy` would benefit many domains and, as other work has noted [227], adoption is disappointingly low.

Of all the shared hosting providers under study, only 6% has more than a single domain with `Content-Security-Policy` in the sample. That being said, there is an interesting long tail for these scarce features, where the provider seem to play a role. For instance, the managed hosting provider `Netsolus.com`, has more than 92% of its domains in our sample enabled with `Content-Security-Policy` and `HttpOnly` cookie , which again suggests a provider wide setting rather than effort of individual webmasters.

7.4.2 Distribution of software patching features

Regarding software installations, Figure 7.2 provides a visual overview of the data in Table 7.1. The colored area shows the portion of all domains where we were able to discover a certain type of software. This is subdivided in installations where we found the patched version (dark blue), where we found an unpatched version (light blue) and where we could not identify the version (grey).

Manual analysis of the software patching features reveals several interesting patterns. In the rest of this section, we discuss software discoverability by attackers and version hiding efforts by defenders. Then we look at the state of patching across the web stack.

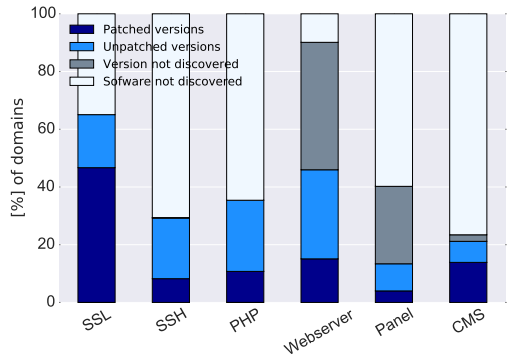


Figure 7.2: Software patching distribution across domains

Hardening practices

Discovering the presence and version of a software installation on a domain is more than a measurement problem. The techniques we deploy can also be used by attackers seeking vulnerable targets, especially if they scale easily. This incentivizes defenders to harden software installations to be less trivially discoverable and to not reveal version information.

Indeed, in the case of the three main CMSes, a basic scan was rarely effective. Figure 7.3 shows that most installations were discovered only through more intrusive industry tools, described in Section 7.3.2. Overall, 23% of the domains had one of the three main CMSes installed. To determine the validity of our results, we manually inspected 40 domains per CMS type, both from domains

for which we discovered an installation and from those for which we did not. We found one false positive where the domain did not have any CMS and no false negatives. Most of the pages that we marked as no CMS pages were either static HTML or featured some custom-made CMS. It is an open question as to what made the discovery more difficult: webmaster action, provider action, or the default configuration provided by the software vendor.

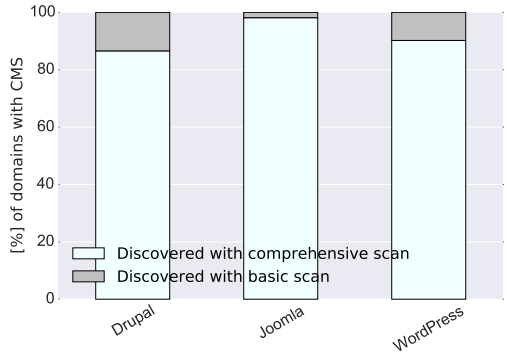


Figure 7.3: Portion of CMS installations discovered via basic vs. comprehensive scans

Similarly to CMS, most of the well-known admin panels were only discovered after a more comprehensive scans. We found them on 40% of the domains. In a shared hosting environment, admin panels seem a necessity, so the actual prevalence is likely to be higher. Many providers, however, appear to shield them from being discovered, even by more comprehensive scans. They are using custom solutions or hide them behind customer portals with restricted access.

Version hiding is also a popular hardening technique. For SSH, all version information is available, as required by the protocol. It is interesting that PHP almost always provides version information, whereas only 50% of HTTP web-servers came with version information. Finding the version information was harder for admin panels. We managed to find it for around 32% of all domains with one of the main admin panel packages installed. For CMSes, version information could be obtained for around 90% of the installations. Given the known hardening techniques such as password-protecting the `/wp-admin/` directory, disabled PHP execution etc., we suspect that this reflects the efficacy of the industry scanning tools, rather than provider or customer practices [228].

We are interested in the difference among providers in version hiding efforts. We looked at the percentage of software installations at a provider for which

version information was available. Figure 7.4 displays where providers are located across a range from where just 0-10% of their installations reveal version information to where 90-100% do. The resulting distributions vary considerably by software type. For CMSes, providers are clustered at the high end of the range. Again, this more likely reflects the efficacy of the scanning tools than of provider practices. For webservers, however, we see a very different pattern; an almost uniform distribution across the range. In some provider networks, nearly all versions are visible. In others, virtually none are. The rest are somewhat evenly distributed across the intermediate intervals. If we assume that shared hosting providers have control over the web server configuration, which seems reasonable, then this distribution suggests that most providers are not consistently applying version hiding in one way or another. The mix of both hidden and visible version information might reflect changes in the provisioning processes over time. As new servers get added, a different default setup might be in use, hiding or not hiding this information. For admin panels, we see yet another distribution. A concentration of providers is on the low end of the range, where version information is mostly hidden across their network. This suggests a consistent practice. But we also see a flat distribution over the rest of the range. Here, again, we might see either changing or ad hoc provisioning processes. It seems unlikely that this reflects customer action.

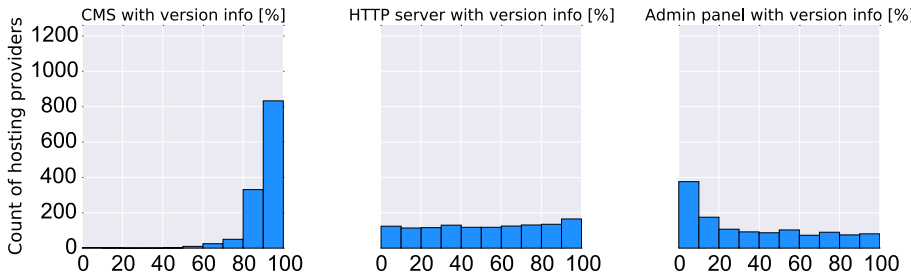


Figure 7.4: Distribution of discoverable software version across providers

Patching practices

More important than hiding version information is to ensure that software is not exploitable in the first place [229]. In this section, we explore patching practices. Figure 7.2 displays the proportion of domains with the patched version

of software, with unptached versions, and installations for which we could not determine the version.

Appendix 7.11 lists the patched versions for each software package and its supported branches. We find that 19% of domains use unpatched SSL. Note that unpatched means SSLv2 and SSLv3 or containing certain vulnerabilities such as Heartbleed, CCS injection, etc. For PHP and SSH, it is clear that fewer domains are running the patched versions relative to the unpatched version. For webservers and admin panels, the majority of installations were running unpatched versions – 87% and 70%, respectively.

In stark contrast to this stand CMS patch levels: less than 35% were not running the latest version. This probably reflects two interlocking mechanisms: a penalty for not updating through higher probability of compromise, as CMSes are known targets for attackers, and increasing support for auto-updating mechanisms, partly in response to these attacks. The fact that lower layers of the software stack such as webserver and SSH do not update as aggressively suggests that the risk of compromise is lower. This might be due to older versions still being patched internally with critical security updates or to the fact that vulnerabilities are harder to exploit remotely than in CMS software.

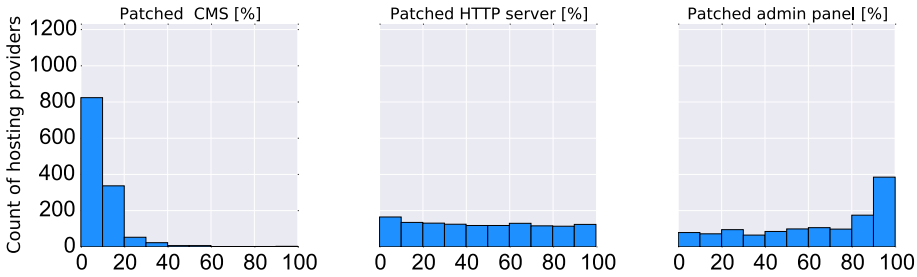


Figure 7.5: Percentage of domains per provider with patched software versions

Figure 7.5 shows the proportion of domains running older versions in each provider. Providers are somewhat normally distributed when it comes to unpatched CMS versions in their network. This is consistent with a natural update cycle over many different customers, each with slightly different time lags. The distribution of providers is more uniform for web servers, which again points to changes in provisioning. We see a positive skew for admin panels, where a significant portion of the providers have almost all installations on the latest version. If we assume that both webserver and admin panel software is under

the provider's control, this difference is remarkable. It might reflect different incentives at work. Since updating incurs cost and can cause compatibility issues, providers might avoid it in the absence of a pressing need. This leaves only changes in provisioning to change the mix of software versions over time, which means the mix of latest and older versions gradually shifts, consistent with the flat distribution of webserver versions. For software that is attacked more often, we would indeed expect a higher concentration of providers running the patched version – which is indeed what we see for the admin panels.

7.5 Direct Relation Between Security Indicators and Abuse

Our main goal in this chapter is to study the relationship between the security indicators we collected and abuse, at the level of shared hosting providers, and eventually *understand* the influence of provider security effort. This justifies the choice of inductive statistical techniques which promise coefficient estimates that lend themselves to meaningful interpretation, as opposed to machine learning, which maximizes predictive power with non-linear methods. Statistical techniques produce exact (up to the arithmetic precision) solutions as well as indicators of confidence, e. g. in the form of significance tests. They can be calculated as a by-product of the estimation, therefore relaxing data requirements compared to heuristic cross-validation typical for machine learning.

Nevertheless, our task is complicated by the fact that each provider hosts a varying number of sites of varying functionality, complexity, exposure, and customer (i.e., webmaster) expertise. The security outcome for each site is a result of joint efforts of provider and webmaster as well as attacker behavior. On the provider level, it is the result of joint efforts of many parties. Therefore, it is convenient and compatible with our statistical approach to model attacker behavior as a random process, which generates counts of incidents observable in our data source.

To explain our method, we contrast it to a naive statistical approach that models the indicators as direct drivers of abuse rates. An example is displayed in Table 7.2. It reports three specifications of a count-data regression model in columns. The units of analysis are providers and the dependent variable is the number of phishing incidents in the provider's shared hosting domains. Model (1) is the baseline, including the two size indicators (cf. Section 7.2). Its Pseudo-R^2 value of 0.68 highlights the importance of size control in this

heterogeneous dataset. Model (2) tries to explain abuse with one technical indicator (of insecurity), namely the number of domains with unpatched CMS. The effect is statistically significant, and in the expected direction: the positive sign of the coefficient means that more domains (log scaled) with outdated CMS coincide with more abuse. However, the more comprehensive Model (3) paints a different picture. The apparent cause of abuse is not the fact that the CMS is unpatched, but the presence of a CMS in the first place. Model (2) missed to control for the fact that websites differ in complexity and thus risk. As a result, it detected a spurious effect in the "unpatched" indicator.

Table 7.2: Quasi-Poisson GLM with Log Link Function

	Dependent variable:		
	Count of phishing domains		
	(1)	(2)	(3)
Number of hosted domains	1.467***	1.539***	1.678***
	(0.083)	(0.085)	(0.078)
Number of IPs hosting domains	0.690***	0.672***	0.472***
	(0.100)	(0.100)	(0.085)
Number of domains with oudated CMS		0.010***	−0.023***
		(0.002)	(0.005)
Number of domains with CMS without version info			−0.019***
			(0.004)
Number of domains with CMS			0.015***
			(0.001)
Constant	−5.596***	−6.150***	−6.743***
	(0.274)	(0.314)	(0.322)
Observations	1,259	1,259	1,259
Dispersion	90	89	68
Pseudo R^2	0.68	0.71	0.78
Note:			*p<0.05; **p<0.01; ***p<0.001
			Standard errors in brackets

These findings confirm previous work at the level of domain names [222]. The authors have concluded that *i)* running popular CMS platforms (WordPress and Joomla) and *ii)* running up-to-date versions of WordPress increases the odds of a domain in getting compromised. Table 7.2 reflects that we find similar relationships on the provider level. In addition, we identify a statistically significant effect of hardening efforts put in place by defenders in hiding the version string.

But does hiding version information really prevent abuse? While plausible in principle, this conclusion is too early and suffers from two issues. The first one is known as ecological fallacy: a relationship at the level of providers might

not hold at the level of domains, i.e., the abuse may not happen at the sites where the security indicator was observed. This fallacy tells us not to interpret aggregate-level analyses as individual causal relationships. As we mainly aim to study the discretion and responsibility of providers, site-level effects need to be isolated, but not necessarily attributed to individual causal relationships. The second issue concerns *unobserved* third variables. There is a plethora of web vulnerabilities and corresponding attack vectors. Any attempt to measure them comprehensively with security indicators is futile, because each indicator may suffer from the issues demonstrated in Table 7.2.

As a way out of this dead end, we first adopt a statistical approach common in psychometrics, where dealing with unobserved constructs has a long tradition. With this lens, hiding the version information should *not* be interpreted as a direct cause of less compromise, but as an indicator of *security effort*, a latent variable indirectly measured by many correlated indicators. The convention to use many indicators reduces the measurement error in each of them. Moreover, latent variables are implicitly defined by the composition of their indicators. The main advantage of using security effort as a latent variable is that we do not need to fully understand the causal relationship of attack and defense mechanisms throughout the global shared hosting space. Instead, it is sufficient to assume that if someone makes above-average effort to, e.g., hide version information, he also takes other steps against attacks, which are not directly captured with indicators. This way, our results become more generalizable and robust at the same time.

In the following, we will infer from data not only one, but several latent variables measuring the security effort of different parties. This allows us to disentangle the joint responsibility using empirical data, without the need to a priori assume and impose a responsible party for each security indicator.

7.6 Security Effort as a Latent Variable

As argued above, constructing latent factors from the security indicators we collected is superior in terms of measuring *security effort* than using the indicators on their own. This approach also allows us to better empirically disentangle provider vs. webmaster influence over these features.

Given the restricted administrative rights in a shared hosting environment, among the features we collected, we assume that features such as `HttpOnly` cookie can be modified by webmasters as well as providers, whereas other

features such as HTTP web server, are more likely to be modified only by the provider itself.

However, this statement is speculative and is not necessarily an accurate reflection of the reality for the following reasons: First, as earlier work also points out, the hosting market is very heterogeneous, meaning that even shared hosting services can be offered in different variations [142]. This essentially means that different providers give different administrative rights to their customers (i.e., webmasters). Second, even if in principle, shared hosting providers leave certain options open to be modified by webmasters, due to the power of default settings, several customers never change those options, even if they can. Our manual analysis shows that even if providers do not directly set up a security feature, they can still trigger security measures via "recommended settings" or regularly nudging their customers towards a more secure environment. The same could hold for software vendors: we have noticed that for instance, from the latest version onwards, cPanel admin panel removed the server type and version parameter from its default server header. Third, there is an interaction between some of the features discussed in the Section 7.3 and content and other applications running alongside a domain, which might require the webmaster to setup certain features such as `Secure` cookie or `HttpOnly` cookie.

To better capture the role of shared hosting providers in securing their domains while accounting for such interactions, we suggest a different methodology than directly using the features that we have collected. We examine the role of shared hosting providers, by empirically and systematically deducing groups of provider features that correlate strongly together yet vary considerably between providers. The results of such an approach would then be an empirical recovery of the effects that are throughout the market more dominant, in the realm of shared hosting providers and are either due to the fact that webmasters have no choice or due to default effects, either of which matters significantly.

We do this in two steps: we first use exploratory factor analysis to define latent variables or *factors*. Empirically inducing factors from data confirms (or denies) whether the hypothesized division of responsibility is actually present in the population. We then quantify to what extent each factor is under the control of shared hosting providers or their customers. Note that we purposefully do not use abuse data in this section in order to avoid circular arguments.

7.6.1 Exploratory factor analysis

Factor analysis uses the correlation matrix of all studied variables and reduces its dimensionality by "looking for variables that correlate highly with

a group of other variables, but correlate very badly with variables outside of that group" [230]. The variables with high inter-correlations then shape a factor. For the factor analysis, we use the security and software features discussed in Section 7.3. Among all our features, the security features are boolean variables with 1 pointing to the direction of the variable. The software features are ordinal from least to most secure with the following order: 0 unpatched versions, 1 patched versions, 2 no software. Note that in order to simplify the input data, from this section onwards, we consider software with 'no version information' as 'patched' software with the latest packaged version. Since our variables are a mix of binary and ordinal, we use Polychoric factor analysis appropriate for ordinal scales.

The input of the factor analysis is an $n \times p$ data matrix with n being the number of measurements (in this case our domains) and p being the number of variables (in this case our features) [231]. The factor analysis generates a set of factors, their corresponding factor loadings and factor scores. Factor loadings express the relationship (correlation) of each original variable with each factor. Factor scores are the estimated values of the factors per measurement (domain). We use parallel analysis for selecting the number of factors, which turns out to be 4. After applying Varimax factor rotation, we obtain the factor loadings in Table 7.3. Each row of the table corresponds to a variable, MR1 to MR4 are the factors, and each number indicates the loading of a variable per factor. The highest loading per variable is shown in bold. Stevens et al. suggest a cut-off point of 0.4 for factor loadings [232].

The results in Table 7.3 indicate all of the 15 features have a medium to high correlation with corresponding factors and hence play a significant role in shaping the factors. Factors MR1 to MR4 each explain a part of the total variance. The cumulative variance explained in Table 7.3 shows that the four factors together are able to explain 62% of the variance observed in all the 15 features. This further confirms our earlier call for having four factors, as the majority of variance is captured by them.

From the results it is clear that these four factors (latent variables) capture different aspects of web security. In other words, the factor analysis not only reduces the complexity of our data, but also control for unobserved third variables, as most of the the collected security features do not directly cause abuse. In the following sections we further use these factors to (a) study the respective role of providers and webmasters and (b) assess their impact on abuse.

Table 7.3: Output of factor analysis

	MR1	MR2	MR3	MR4
`X-Content-Type-Options`	**0.87**	0.11	0.14	-0.01
`Content-Security-Policy`	**0.80**	0.23	-0.01	0.37
`X-Frame-Options`	**0.83**	0.09	0.10	-0.16
HTTP `Strict-Transport-Security`	**0.61**	0.50	0.04	0.03
Mixed-content inclusions	0.26	**0.76**	-0.01	-0.24
Weak browser `XSS` protection	-0.39	**0.68**	0.24	0.29
SSL-stripping vulnerable form	0.08	**0.60**	-0.05	-0.38
`HttpOnly` cookie	0.13	**0.65**	0.14	0.12
`Secure` cookie	0.36	**0.86**	0.03	0.11
Patched HTTP*	0.09	0.05	**0.74**	-0.11
Secure SSL implementation*	-0.15	-0.09	**0.74**	-0.10
Patched SSH*	-0.07	0.04	**0.42**	0.35
Patched PHP*	0.09	-0.12	0.13	**0.55**
Patched CMS*	-0.14	0.01	-0.23	**0.78**
Patched Admin panel*	0.08	0.08	0.10	**0.58**
Loadings' sum of squares	2.90	2.92	1.48	1.90
Proportion of variance explained	0.19	0.19	0.10	0.13
Cumulative variance explained	0.19	0.39	0.49	0.62

* Scale from least to most secure: 0 unpatched, 1 patched or no version, 2 no software

7.6.2 Role of providers in securing domains

The combination of the features per factor and their relative loadings (i.e. how much they correlate with different factors) in Table 7.3 suggest that each of the factors capture a different set of web security efforts. MR1 consist of features that are partially capturing **content security practices**. Features in the MR2 factor seem to capture more **webmaster security practices**. Given the high loadings on variables such as unpatched HTTP server and insecure SSL implementation, MR3 clearly captures more **infrastructure security practices** whereas MR4 seems to capture **web application security practices**. In other words, the factor analysis shows that features which one might assume to be related, such as CMS and admin panel, do indeed covary with each other in practice, as they correlate with the same underlying factor.

This leads us to the following hypothesis: we expect MR1 and MR2 to be be less affected by providers' security efforts than MR3 and MR4. We examine the relation between the factors and the effort of providers by constructing four linear models.

To construct these models, we first calculate the factor scores (the estimated values of the factors) from the factor analysis, in a way that a score is assigned

Table 7.4: Linear Regression Model

	Response Variable: Security Factor(s)			
	MR1	MR2	MR3	MR4
	(1)	(2)	(3)	(4)
Hosting provider fixed effect	yes	yes	yes	yes
Constant	-0.250^{***}	-0.300^{***}	0.100^{*}	0.420^{***}
	(0.064)	(0.066)	(0.043)	(0.051)
Observations	442,075	442,075	442,075	442,075
R^2	0.077	0.066	0.270	0.200
Adjusted R^2	0.075	0.064	0.270	0.200
Residual Std. Error (df = 440801)	1.400	1.400	0.920	1.100
Note:				*p<0.05; **p<0.01; ***p<0.001
				Standard errors in brackets

to each data point (domain). We then construct a linear regression model per factor, with the factor score as the dependent variable and provider fixed effects as the independent variable. The provider fixed effect consists of fitting a separate dummy variable as a predictor for each of the hosting providers in our sample. We are interested to see how much of the variance in each of the factors (dependent variables) can be explained by provider efforts, as opposed to individual webmaster efforts. The relative difference between the amount of variance explained by each model indicates the extent that shared hosting providers influence the security indicators associated with these factors.

Table 7.4 shows the four models and their R^2 and adjusted R^2 values. To simplify presentation, we omit the estimated coefficient for each hosting provider. The findings confirm our hypothesis: hosting provider fixed effects explain at least three times more variance in MR3 and partially in MR4 than MR2 and MR1. MR1 and MR2, as we earlier hypothesized, with the lowest amount of explained variance, seem to be more a compound of webmaster level efforts rather than provider level influence. Disregarding the effect of measurement noise, one should note that the R^2 value cannot be expected to be close to 1 in MR3 and MR4, because there are differences between hosting packages offered by different providers. Similarly, MR1 and MR2 are above zero because customers with specific requirements self-select their provider.

Using the regression results, we are able to empirically confirm our assumptions regarding the role of hosting providers in influencing each of the latent factors constructed using factor analysis. In the following section, we use these

results to examine which of the factors have a higher impact on abuse prevalence and which party, provider or webmaster, can influence it more.

7.7 Impact of Security Efforts on Abuse

Having empirically determined the relationship between provider and website security by constructing latent factors, we now compare the incidence of abuse at providers to the factors. The objective is to test the extent to which the actions of hosting companies and individual webmasters influence the prevalence of abuse, using malware and phishing sites as case studies.

We define our dependent variable Y_i as the number of blacklisted domains in our abuse datasets for $i = 1, \ldots, n$, with n being the total number of hosting providers, where Y_i follows a Quasi-Poisson distribution[10]. We construct separate regression models for phishing and malware.

The regression results for phishing and malware abuse are shown in Tables 7.5 and 7.6, respectively. In order to be able to observe the effect of all variables on abuse, we construct one model per variable (models 3-6), together with a final model that includes all variables (model 7). We report the dispersion parameter for each of the models. Note that the Quasi-Poisson models are estimated using a Quasi Maximum Likelihood and are adjusted via the reported estimated dispersion peremeter. Therefore, the Log Likelihood values are reported from the original Poisson fitted models, as recommended in practice [233].

Moreover, since previous research already established the strong relationship between provider size and abuse prevalence [159, 38], we use model 2 with only size variables as our base model, and study the extent to which our four factors further explain the variance in abuse, on top of the R^2=0.71 of model (2). Hence, in addition to the normal pseudo R^2 value used as a goodness of fit measure for the Quasi-Poisson models [149], we report the pseudo R^2 value with respect to model (2) for each table.

We include both phishing and malware data because while we see some similarities in how abuse type relates to the security characteristics, we also anticipate that there will be differences. Given the specialization in cybercriminal activity, the actors themselves and their preferred methods of compromise are likely different, as is the effectiveness of different security efforts on the side of defenders [38].

[10]We choose Quasi-Poisson over Poisson due to the over-dispersion (unequal mean and variance) in our data.

Table 7.5: Generalized Linear Regression Model (GLM) for count of phishing domains observed per provider

	Response Variable: Count of **phishing** domains						
	Quasi-Poisson with Log Link Function						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
# domains on shared hosting		1.500***	1.400***	1.400***	1.500***	1.800***	1.800***
		(0.083)	(0.081)	(0.093)	(0.082)	(0.080)	(0.110)
# IPs on shared hosting		0.690***	0.780***	0.750***	0.700***	0.620***	0.660***
		(0.100)	(0.100)	(0.110)	(0.100)	(0.086)	(0.120)
MR1			−0.570***				−0.570*
Content security			(0.140)				(0.240)
MR2				−1.100***			−1.100**
Webmaster security				(0.270)			(0.390)
MR3					−0.360**		0.170
Web infrastructure security					(0.110)		(0.150)
MR4						−1.100***	−1.200***
Web application security						(0.110)	(0.160)
Constant	3.300***	−5.600***	−5.700***	−5.500***	−5.600***	−7.100***	−7.100***
	(0.250)	(0.270)	(0.270)	(0.300)	(0.270)	(0.320)	(0.440)
Observations	1,259	1,259	1,259	1,259	1,259	1,259	1,259
Log Likelihood	-99,401	-30,094	-29,152	-28,160	-29,516	-26,173	-24,637
Dispersion	2103	90	88	112	91	75	129
Pseudo R^2	-	0.71	0.72	0.73	0.71	0.75	0.76
Pseudo R^2 with regards to model 2	-	-	0.032	0.066	0.015	0.14	0.19
Note:							*p<0.05; **p<0.01; ***p<0.001

For phishing, three out of four factors are statistically significant when included together in model (7). Webmaster security (MR2) and web application security (MR4) play a statistically significant role in reducing phishing abuse: for each one unit increase in each of these factors, keeping all other factors constant, phishing abuse drops by $e^{1.100} = 3$ and $e^{1.200} = 3.32$, respectively.

The most prevalent individual indicator that makes up MR2 is the presence of an `HTTPOnly` cookie, which is a standard XSS defense. To reiterate, we interpret these features as indicators of latent factors measuring security effort (see Section 7.5). For example, the results suggest that when individual webmasters harden the cookie properties of their websites, this is an indication that they also take other (unobservable) measures to inhibit abuse. The results form

Table 7.6: Generalized Linear Regression Model (GLM) for count of malware domains observed per provider

	Response Variable: Count of **malware** domains						
	Quasi-Poisson with Log Link Function						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
# IPs on shared hosting		1.600***	1.600***	1.600***	1.600***	1.600***	1.400***
		(0.090)	(0.090)	(0.087)	(0.089)	(0.098)	(0.095)
# domains on shared hosting		0.460***	0.560***	0.520***	0.470***	0.480***	0.600***
		(0.110)	(0.110)	(0.110)	(0.110)	(0.110)	(0.110)
MR1			−0.700***				−0.310
Content security			(0.170)				(0.190)
MR2				−1.300***			−1.300***
Webmaster security measures				(0.290)			(0.300)
MR3					−0.380**		−0.130
Web infrastructure security					(0.130)		(0.130)
MR4						-0.170	-0.360*
Web application security						(0.140)	(0.140)
Constant	4.300***	−4.800***	−4.900***	−4.600***	−4.700***	−4.600***	−4.300***
	(0.240)	(0.310)	(0.310)	(0.290)	(0.300)	(0.330)	(0.300)
Observations	1,259	1,259	1,259	1,259	1,259	1,259	1,259
Log Likelihood	-273,893	-79,646	-75,986	-73,496	-78,181	-79,392	-71,461
Dispersion	5800	330	334	298	332	320	288
Pseudo R^2	-	0.71	0.73	0.74	0.72	0.71	0.74
Pseudo R^2 with regards to model2	-	-	0.044	0.077	0.017	0.001	0.098

Note: *p<0.05; **p<0.01; ***p<0.001
Standard errors in brackets

MR4 indicates that running up to date versions of CMS and admin panel, or hiding the version information, or running no software, is negatively associated with compromise. We suspect this is due to the fact that certain providers administer CMSes themselves, to make themselves and their customers less prone to compromise, given the vulnerabilities imposed by CMSes and admin panels [229, 234]. It also shows that these are the areas that providers' effort can be very effective.

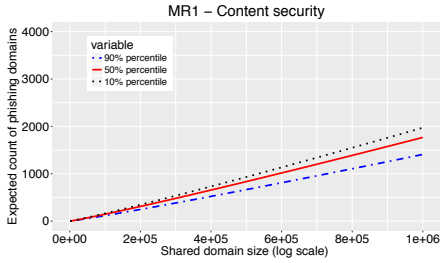
For malware, only MR2 (webmaster security) and MR4 (Web application security) are significant in model (7). From the two, webmaster security (MR2) explains most variance in the malware abuse, both when modeled alone (model

(4)), and when modeled with other factors (model (7)). Again, given that `HTTPOnly` cookie and `Secure` cookie dominantly shape webmaster security factor (MR2), their significant relation with reducing malware abuse is therefore very intuitive. MR4 plays a less significant role in explaining malware abuse. We suspect this is due to the differences in the nature of phishing and malware attacks, attack techniques, and exploited resources.

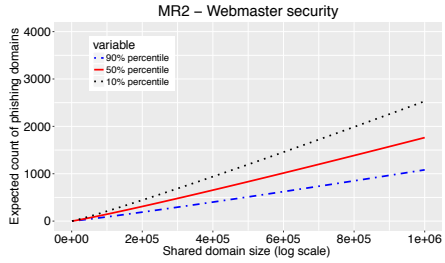
Moreover, MR1 (Content security) and MR3 (Web infrastructure security) show a statistical significant relation with malware and phishing abuse only when considered alone (model (3) and (5), respectively). By inspecting regressions including other combinations of factors (not included for space considerations), it appears that MR1 is the more robust indicator than MR3 for the malware regression.

Overall, the combined model explains 19% of the variance for phishing prevalence and 10% for malware prevalence among providers, beyond the baseline of 71% explained variance, showing that both webmaster and provider efforts influence abuse prevalence. The influence of these efforts on abuse rates, for disparate types of abuse (in our case web-based malware and phishing), is consistent in direction and somewhat varying in magnitude. Finally, we note that while we have explained some of the variation in abuse prevalence among shared hosting providers, much remains unexplained. This should in turn motivate the collection of additional discriminating features in follow-up studies.

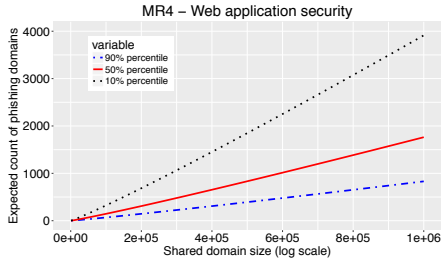
Figure 7.6 uses the model to demonstrate how the factors influence abuse prevalence. Figure 7.6 (a) plots the expected number of phishing incidents as a function of provider size while varying the value of MR1 (content security) and holding other factors at their median value. Note that we plotted one figure for each of the factors that showed a significant relation with phishing abuse in model (7) of table 7.5. We can see that the bottom 10% of providers (with the least effort as measured by MR1) should experience less than one and half as many phishing attacks as the top 10%. In the case of MR2 (webmaster security), the bottom 10% of providers experience more than twice as many phishing attacks as the top 10%. For MR4 (web-application security), the difference is even more pronounced: the best-performing 10% providers by this measure should experience more than 4 times less phishing than the bottom 10% of providers. These findings provide reliable empirical evidence regarding the security benefits of providers adopting industry best practices, most notably proactive patching practices. Given that patching is costly, such evidence is critical to move the industry in the right direction.



(a) MR1



(b) MR2



(c) MR4

Figure 7.6: Plot of expected phishing abuse counts against shared domain size for MR1, MR2 and MR4 (from model (7) of Table 7.5)

7.8 Limitations

As with all large-scale studies of real-world applications and implementations, we should reflect on the potential impact of measurement errors and other limitations. Potential errors in our measurements are caused by the fact that we scan only for the main software packages across the web stack. Also, the collected data can be misinterpreted. One Dutch provider, for example, rolls out its own back-ported security patches for CMSes, without updating the version number. Another limitation stems from the use of a rather crude metric for patching (patched/unpatched). An alternative would be looking, for example, at the distance in time between the installed version and the patched versions.

We captured information on 15 different features, associated with security and patching practices. Some of these features were very biased, mostly because of their extremely low prevalence. Ideally, we would include features with more variance across the population. The features were not interpreted as direct defenses against web compromise, but rather as latent factors that signals effort. However, these feature might also reflect other latent factors in addition to security effort, such as website functionality, popularity, complexity and exposure.

Finally, the reader should bear in mind that our study aggregates abuse at the provider level, while features are collected on a separate sample of uncompromised domains in order to increase sample size. Future work could collect features on compromised websites directly to establish stronger differentiation between individual and provider efforts.

7.9 Related Work

Because our work seeks to measure web security in shared hosting environments, identify the role of the hosting providers and its impact on abuse, we build upon several aspects of the literature.

Measuring vulnerabilities of websites and webservers: There are numerous measurement studies aiming to detect web vulnerabilities across domains (e.g., [98, 67, 108, 235, 96, 97, 236, 26, 107, 227]). For example, Van Goethem et al. assessed 22,000 websites and studied the severity of certain vulnerability and security features [26]. SecuBat developed by Kals et al. automatically detects XSS and SQL injection vulnerabilities [96]. Lekies et al. analyzed the 5,000 most popular domains and found that 9.6% of the examined websites carry at least one DOM-based XSS problem [97]. Weichselbaum et al. detected domains adopting CSP and studied how effective the policies were in protecting against XSS attacks [107]. Calzavara et al. also studied CSP adoption via a large scale measurement study and concluded that in addition to limited deployment, existing policies are frequently misconfigured [227]. Van Acker et al. performed a systematic examination of login-page security and found that many login pages are vulnerable to leaking passwords to third parties and to eavesdropping attacks. They also observed that a few login pages deploy advanced security measures [98]. Lastly, Aviram et al. introduced two different attack techniques against SSLv2 and concluded that SSLv2 weaknesses are a significant threat against SSL ecosystem [67].

Threats against shared hosting: In addition to general domain vulnerabilities, there are certain threats specific to domains hosted on a shared server. In shared hosting, a physical server is shared among multiple clients, ranging from a few to over a thousand. Customers are allocated a fraction of a machine's overall resources and given limited user privileges. Server-side software must be managed by the provider. Canali et al. examined security performance of a small group of shared hosting providers and concluded that the majority were unable to detect even basic attacks on their infrastructure [10]. The Anti-Phishing Working Group reported that some attackers would compromise shared hosting servers and load phishing pages on each of the hosted websites [122]. Tajalizadehkhoob et al. investigated the security performance of different hosting provider types in terms of phishing abuse take-down times and concluded that phishing domains in shared hosting providers often last longer than other group of providers [142]. The potential for compromise on a shared environment abuse was first pointed out by Nikiforakis et al. [39] and Mirheidari et al. [31], who noted that the lack of enforced session isolation leaves shared web hosts open to mass compromise. Perhaps reflecting this strategy, Vasek et al. found that phishing websites were disproportionately likely to be hosted in a shared environment [222].

Relationship between vulnerabilities and abuse: A few studies empirically investigated the relationship between the vulnerabilities of a domain and the likelihood of being compromised. Vasek and Moore found that Apache and nginx server software and popular CMS platforms, most notably WordPress, Joomla! and Drupal, are positive risk factors for webserver compromise [222]. In fact, a key counterintuitive finding was that fully patched installations have a higher likelihood of compromise than unpatched ones. Soska and Christin developed an approach that predicts whether websites will be compromised in the near future. The prediction is done via a classifier that is trained on features that are extracted from a set of both malicious and benign websites. They found CMS type and version to be predictive features, suggesting that many websites could be compromised through a vulnerability in their CMS [43].

Role of intermediary in dealing with abuse: A number of studies focused on different types of intervention done by intermediaries (e.g., [135, 129, 237, 10, 238, 134, 239, 218, 240, 241]). Moore and Clayton, for example, examined the effectiveness of phishing websites take-down by web hosting providers and concluded that website removal is not yet fast enough to completely mitigate the phishing problem [129]. Stock et al. performed a large-scale notification

campaign of website owners using a set of over 44,000 vulnerable websites and concluded that there are no reliable notification channels that would significantly inhibited the success of notifications [239]. Li et al. examined the life cycle of 760,935 hijacking incidents identified by Google Safe Browsing and Search Quality, and found that direct communication with webmasters increased the cleanup rate by 51%. They concluded that in order to decrease the number of security incidents, one could increase the webmaster coverage of notification while also equipping hosting providers with tools alerting webmasters to update software [217].

We build on the existing work in several ways. First, we extend the measurement approach developed by Van Goethem et al. [26] to collect a broader set of features. Next, we move the level of analysis from individual domains to providers. In areas beyond shared hosting, researchers have repeatedly found that the intermediaries can play a key role in improving security [6, 237, 242, 243, 244, 159, 245].

In chapter 4 of this dissertation, we studied the different factors at work in the abuse data generation process of hosting providers. We identified structural properties and security efforts of hosting providers, behavior of attackers, and measurement errors, as factors that can influence concentrations of abuse. Further, we showed that the structural properties of hosting providers alone – such as different size, price, and business model variables – can explain more than 84% of the variance in abuse concentration of hosting providers [159]. Noroozian et al. investigated the closely related question of how provider security practices impact abuse concentration and whether the outcome of provider security practices can be indirectly inferred (as a latent variable) from multiple sources of abuse data employing Item-Response Theory [38]. Their results quantified the impact of security practices (without knowledge of what those practices may be), demonstrating predictive and explanatory power. Finally, Sarabi et al. studied the implications of end-user behavior in applying patches. They observed that although both end-users' patching speed and vendors' facilitating policies help in improving the overall security posture of a host, they were also overshadowed by other factors, such as frequency of vulnerability disclosures and the vendors' speed in deploying patches [246].

In our study, the hosting company's role is critical, since many domain owners will not be willing or able to adequately secure their site. Our data collection is not based on a random sample from all domains, but on a sampling strategy that covers all shared hosting providers. We present a new approach to disentangle the role of providers and customers in protecting domains. This also allows us to extend the work on the relationship between vulnerability and

compromise from the level of individual webmasters to that of providers. Last, but not least, we provide the first estimate of the potential gains of such efforts for lowering compromise levels.

7.10 Conclusions and Discussions

We have undertaken an extensive study of web security efforts. The purpose of this work is (i) to study the state and landscape of security hygiene at the level of domains and shared hosting providers, (ii) to disentangle the defensive efforts of providers and their customers, and (iii) to assess their impact on web compromise.

Our descriptive findings regarding the web-security landscape show that most domain security features occur sparsely across the domain and provider space. Even here, though, we see the potential influence of providers. A tiny fraction of providers has very high adoption rates of certain features like `Content-Security-Policy` and `HttpOnly` cookie. They appear to offer more managed forms of shared hosting, which might enable them to exert more control over feature configurations of their customers.

Regarding software patching, higher levels of the web stack such as CMS and admin panels are updated more than infrastructure software like SSH and PHP. This might reflect the fact that CMSes and admin panel are attacked more aggressively. Interestingly, even though infrastructure software is typically under the control of the provider, we see a lot of heterogeneity of versions within the same provider. We suspect this is due to changes in provisioning processes over time. Since patching is costly, earlier default configurations might not get updated unless there is an urgent need.

The individual features should not be interpreted as being directly causing web compromise, for reasons that we laid out in Section 7.5. It is more valid and informative to interpret them as indicators of a latent factor that is the actual causal driver, namely security effort. Using exploratory factor analysis, we uncovered four such latent factors: content security practices, webmaster security practices, web application security practices and infrastructure security practices. The fixed-effect regression analysis uncovered that providers have control over infrastructure and application security, as we expected. Regarding CMSes specifically, however, the influence of providers is more surprising. This software can run client-side, but still providers influence patch levels. This might mean that a subset of providers administer these installations themselves, or that they found ways of getting their customers to patch in a timely fashion.

Finally, we model the impact of the four security factors on the compromise rate of providers, as observed in phishing and malware incidents, using Quasi-Poisson GLM regression. Taken together, the results suggest that both webmaster and provider efforts influence abuse prevalence. While provider security efforts play a more significant role in fighting phishing abuse, webmasters are also effective in reducing abuse rates. Most of the four factors play a statistically significant role in reducing abuse, either when modeled alone or with other factors. More specifically, the factor that captures web-master security efforts such as `Secure` and `HTTPOnly` cookies, shows a negative relation with both malware and phishing abuse, highlighting the effectiveness of webmasters' efforts in fighting abuse. The regression results have also shown that web-application security, a factor associated with provider efforts, has a strong significant negative relation with malware and phishing abuse. To illustrate the relative impact, we show that the best-performing 10% of providers by this measure experience 4 times fewer phishing incidents than the bottom 10% providers.

In short, our study shows that providers have influence over patch levels–even higher in the stack, where CMSes can run as client-side software–and that this influence is tied to a substantial reduction in abuse levels. Our study has provided the first rigorous evidence of the security benefits of provider efforts to increase patching levels. This is a critical finding for the dialogue, with and within the industry community, about the merits, costs and benefits of the proposed best practices–e.g., [9]. The takeaway for providers is that improving patch levels pays off. They can do this by administering themselves more of the software installations across the web stack, by securely provisioning default installations or by deploying some other mechanisms that enable them to get their customers to collectively reach higher patch levels.

Beyond the area of shared hosting and web compromise, our study provides a new methodological approach to disentangle the impact of different actors on security. This approach can be adopted to study other areas of joint responsibility, such as between cloud hosting providers and tenants, or corporate system administrators and end users.

Measuring effort in a heterogeneous environment with different requirements is hard. Future work could measure feature use before (or together with) security. Measuring security alone is vulnerable to spurious correlations and inferences, when not controlling for the differences in website functionality, complexity, exposure, et cetera. Another future direction is to make this approach longitudinal, in order to tell apart which fraction of security effort is reactive (i.e., reacting to compromise) and to better detect the direction of causality. In

the end, we hope to provide better empirical support for industry best practices focused on hosting providers.

7.11 Version Information Details

Table 7.7: The list of versions per software that are considered patched (patched = latest packaged version in Ubuntu, Debian or CentOS)

Software	Version considered patched
Apache	[2.2.15 - 2.2.22 - 2.4.7 - 2.4.10 - 2.4.18 - 2.4.20 - 2.4.23]
SSH	[5.3p1 - 5.9p1 - 6.0p1 - 6.6p1 - 6.6.1p1 - 6.7p1 - 7.1p2 - 7.2p2 - 7.3 - 7.3p1]
WordPress	[4.7 - 4.6.1 - 4.5.4 - 4.4.5 - 4.3.6 4.2.10 - 4.1.13 - 4.0.13 3.9.14 - 3.8.16 - 3.7.16]
Joomla!	[3.6.4]
Drupal	[7.52 - 8.2.3]
cPanel	[7.52]
DirectAdmin	[1.50.1]
Virtualmin	[1.820]
Plesk	[12.5.30 - 17.0.16]
Microsoft IIS	[12 - 10 - 9 - 8.5]
Nginx	[1.2.1 - 1.4.6 - 1.10.0 - 1.10.1 - 1.10.3 - 1.11.5]
PHP	[5.3.10 - 5.3.3 - 5.4.45 - 5.5.9 - 5.6.27 - 5.6.28 - 6.6.30 - 7.0.11 - 7.0.12 -7.0.13]

CHAPTER 8

Conclusions

This research sought to understand and improve the role of hosting providers in cybersecurity provision. We presented five peer-reviewed empirical studies. These explored the hosting space and the role hosting providers can play in security provision, with the ultimate aim of identifying areas for intervention. Together with the literature review (chapter 2), the five studies set out to answer the following research question:

How can the security performance of hosting providers be measured and improved?

8.1 Summary of the Empirical Findings

Each chapter answered part of the research question. The conceptual framework presented in chapter 2 disentangled factors associated with abuse. That framework then guided our review of the state of the art in hosting security and helped us to pinpoint gaps in existing scientific work, which this dissertation sought to fill.

We identified a variety of initiatives to improve the security of hosting services. None, however, considered the most basic characteristics of the market, such as how many providers there are, their distribution worldwide, and the types of services they offer. The lack of such information has hindered development of reliable best practices and impeded performance evaluation in this market. A practical solution, in this case, was to go back to the root of the problem and study the context in which cybersecurity problems occur, namely, the hosting market.

Chapter 3 developed an approach to uncover and grasp the complexity of

the hosting market. The method we proposed for identifying hosting providers was to map technical identifiers – IP addresses and domain names captured in passive DNS data – to the economic agents behind the hosting services, identified by organizational data in the WHOIS database. With this method, we identified the organizations that *are* responsible for the security of hosting services and actually *can* take action to improve it.

We also surveyed the hosting landscape, empirically identifying a diverse set of business profiles. The landscape revealed was very *heterogeneous*. Some providers owned only a single IP address, which they used to offer shared hosting services. In terms of global distribution, hosting providers were located in more than 150 countries. Several hosting providers had infrastructure in multiple countries.

Following the mapping in chapter 3 of the hosting space, **chapter 4** developed an analytical and statistical method to infer information about hosting providers' security performance from noisy abuse data. Our analytical model decomposed the different sources of variance present in abuse data, such as defender properties, attacker behavior, and measurement and attribution errors. For defender properties (in this case regarding the hosting provider), we distinguished two main types: inherent structural properties, such as the size of their customer base and infrastructure, and security efforts, particularly reactive and proactive measures taken by providers to secure their networks. With these factors in mind, we sought to advance on previous methods for assessing hosting providers' security performance based on incident data. We developed a new approach that draws on concentrations of abuse at hosting providers after controlling for other characteristics, such as hosting providers' structural properties.

To quantify the impact of providers' structural properties on abuse concentrations, we empirically modeled the concentration of phishing domains in the network of hosting providers. Our results showed that a handful of provider structural properties – such as number of domain names, number of IP addresses used for web hosting, and the size of their shared hosting business – accounted for 84% of the variation in phishing abuse concentrations. These variables were easily measurable on a large scale (for all of the 45,000 providers), and captured providers' exposure – sometimes called their 'attack surface'. These factors are associated with features of providers' business models. In short, we found hosting providers with a large customer base and a larger shared hosting business to be more exposed to phishing attacks.

We constructed additional models to measure the impact of factors that were difficult to observe on a large scale (that could not be assessed for all

of the 45,000 providers). In short, we found that providers' pricing strategy, website popularity, time in business (years of experience), and use of applications known to be vulnerable played a significant role in abuse concentrations. These explained an additional 77% of the remaining variance in phishing abuse. The level of ICT development in the countries where phishing domains were hosted was also a significant factor in abuse concentrations, after other differences between countries were controlled for. In addition, as more than 85% of the variance in phishing abuse was explained by providers' structural properties, our results suggest, though indirectly, that providers' security efforts have less explanatory and predictive power than inherent properties and business model when it comes to concentrations of phishing abuse. However, this relation requires testing for different types of abuse. Moreover, the impact of providers' security efforts on abuse concentrations merits quantification through direct measurement.

Chapter 5 examined providers' security performance. This was measured by abuse concentrations, using an approach similar to that developed in chapter 4. We assumed that attack concentrations and attackers' preferences would vary according to the nature of the attack and the centrality of the abuse to the attacker's operation. For example, we suggested that from the attacker's perspective it is more critical when a command-and-control (C&C) domain – a domain in charge of communicating commands to other infected machines – is taken down than when, say, a phishing site is taken down. We therefore expected attackers to prefer hosting their C&C infrastructure with providers that are slow to take down C&C domains. More specifically, we expected a relation between the concentration of C&C domains within providers and their reactive security efforts, measured by the uptime of C&C domains.

This was tested in the study presented in chapter 5. Here, we modeled the distribution of C&C domains across providers. Four measures of provider size and business model were found to explain some 71% of the variance in C&C counts. Given that the amount of variance explained by provider structural properties was 13% more for phishing data (84%), we conclude that the frequency of C&C incidents is determined less by providers' structural properties than by their security efforts, compared to phishing incidents. Further, concentration of C&C domains was negatively related to a rule of law indicator for the countries where the domains were hosted. Finally, the providers' take-down speed was only weakly related to C&C concentrations, explaining just an additional 1% of the variance. Thus, attackers appear to show little preference for providers that allow long-lived C&C domains. On a more general level, these results suggest that providers' structural properties, such as size and pric-

ing strategy, play a much more economically and statistically significant role in driving C&C concentrations, compared to any reactive security measures providers take, such as the effort they put into taking down abused websites.

Chapter 6 shifted the focus from defender properties to attackers' behavior and strategies. Here, we confined our investigation to one case study: Zeus malware, which is a leading family of malware used for attacking financial institutions. We looked only at the targets of Zeus attack in an effort to better understand attackers' behavior, regardless of the providers that hosted the victimized domains.

We transformed four years of noisy data from Zeus configuration files into structured data on attack targets and attackers' instructions sent to the machines infected with Zeus malware. Our explanatory analysis produced several findings. For example, the attacks were very concentrated. 90% of the attacks were aimed at only 15% of the overall targets. Surprisingly, we observed that this concentration was not driven by size of the target financial institution; nor did size predict the intensity of attack. We also observed wide variation in attack persistence. Some institutions were attacked very briefly, while others underwent attacks during the entire observation period (216 weeks). We speculate that the brief attacks were part of a trial-and-error process in which attackers sought new targets. Strengthening this speculation, we discovered that long-lived botnets had more trial-and-error attempts than short-lived ones.

Attackers, furthermore, copied one another's target list. Code reuse and code similarity rates were very high. One would expect code sharing or code stealing to lead to low code development costs, low market entry barriers for attackers and newbies, and ultimately to a rise in the number of attacks. The data contradicts this conjecture, however. Although attackers tried new targets over the whole observation period, there seemed to be a ceiling in the overall number of targets attacked at any one time. Taken together, these results suggest that what drove the Zeus attack volume was neither target-list-as-a-service nor code-as-a-service. The determining factors likely lie elsewhere in the criminal value chain, such as in the recruitment of money mules or the money transferring policies of financial institutions, or in the cash-out segment of the value chain.

Chapter 7 investigated providers' proactive security efforts and sought to quantify their impact on abuse concentrations for the specific case of shared hosting. Although hosting providers are a key actor in fighting website compromises, we found that their ability to prevent abuse is constrained by the security practices of their own customers. In shared hosting, customers operate under restricted privileges. Providers thus retain more control over system con-

figurations. Our study constituted the first comprehensive empirical analysis of proactive security practices of shared hosting providers. We examined 15 proxy features, from which we distilled four major latent factors capturing security efforts: content security, webmaster security, web infrastructure security, and web application security.

Providers and webmasters employed various soft techniques, such as hardening the software discovery process and hiding version information, to make it harder for criminals to exploit vulnerabilities in their applications. Our results confirmed that providers exert significant influence over web infrastructure and web application security related to the software stack in their hosting environment. We also observed that content and web application security played a significant positive role in reducing website compromises, after controlling for size. Our findings suggest that when a provider moves from the bottom 10% to the best-performing 10% in the market in terms of web application security, it experiences four times fewer phishing incidents. Thus, providers' efforts at the software patching level – even for client-side software like content management systems (CMS) – can eventually lead to a substantial reduction in web compromises.

Taken together, the findings of these studies provide a deeper and more detailed view of the hosting market overall and of the security performance of hosting providers and the factors that influence it in particular. The section below expands on the implications of these findings for practice.

8.2 Implications for Practice

The introduction of this dissertation pointed out the particular challenges faced in governance of the hosting market, especially when it comes to improving cybersecurity. The hosting market, after all, is globally distributed and encompasses a multitude of actors. Actors' roles are intertwined and not clearly distinguishable, and empirical knowledge about hosting providers is limited. Furthermore, there are as yet no regulations in place specific to the hosting market, and no governance mechanism has been fully effective in improving the security of this market, given its known characteristics.

Negative externalities caused by insecurity in this market affect not only hosting providers, but also end-users, software vendors, economies, and societies as a whole. Yet, achieving adequate and stable levels of hosting service security requires efforts by both providers and users of these services. It there-

fore constitutes a collective action problem. Overcoming such a problem requires collaboration between the different actors involved.

Our research results provide a better understanding of how this market functions, particularly in terms of investing in security and countering abuse. We learned about the geographical distribution of providers, the types of services on offer, and particularly the security practices employed to reduce abuse. We also came across a variety of ways in which the market seems to regulate itself. Some providers appeared to have a *market* incentive to strive towards higher security performance, but there were failures as well.

This section explores lessons for practice from the findings of this research. We revisit the different instruments associated with each of the four governance modes introduced in chapter 1 [247]. The paragraphs below discuss implications of our findings for using governance to tackle the collective problem of security provision in the hosting market. Recall that governance here is defined as processes and structures for coordination, steering, and decision-making among the variety of actors involved.

8.2.1 The market

Our results confirmed that hosting providers can function as control points, and therefore exert influence on the amount of abuse in their networks. However, providers vary widely in their security performance, as indicated by the concentration of abuse in their networks. Some hosting providers can influence the security of their services, and many of them do. They take measures that eventually reduce the number of abuse incidents, perhaps leading to lower costs for reactive security activities. However, there are still failures in this market. It is thinkable that the market itself could stimulate these failures to do better. But they may be more effectively swayed by other governance means, such as network, community, or hierarchical approaches. Before looking at these, we first discuss the implications of our results for the players in the hosting market.

We found that security outcomes in terms of concentrations of abuse in the networks of hosting providers can be explained and predicted by two main factors: providers' inherent *structural* properties and providers' reactive and proactive security *efforts*. By providers' structural properties, we mean the nature of providers' business and their customer base. Our results confirmed that these properties have a major impact on abuse concentrations and can explain a large portion of variance in abuse concentrations. In our study, more than 84% of the variance in phishing abuse was explained by the structural properties of the providers. Providers with a very large customer base and those

offering shared hosting as a service were by default more exposed to attacks than providers with a smaller customer base and less vulnerable types of business. Providers' reactive and proactive security efforts refer to the measures providers' take to secure their services. In our study, some 12% of the variance in abuse was explained by providers' security efforts.

The implication is clear: hosting providers need to be cognizant of the structural properties of their services that expose them more to cyberabuse, while investing in reactive and proactive security efforts. Practically speaking, a hosting provider might adjust the services it offers in order to reduce its vulnerability to abuse. For instance, providers that offer shared hosting services could internalize all security decisions regarding client-side applications, to retain better control over patching levels. Or, limits could be placed on the applications that clients may install.

Moreover, our results suggest that consideration of a provider's inherent structural properties and the services it offers can render even reactive and proactive security measures more effective. For instance, a shared hosting provider that, as typical, has rather low margins, and hence invests little in security, could utilize affordable means to keep its services secure. An example is nudging customers via notifications concerning regular software updates. Such a provider could also use default settings to impose certain measures on their customers, such as flags and headers, to improve the security performance of a website regardless of its content. That said, in practice numerous providers still do not take even these simple steps.

We shared some of our results in brief with a few Dutch hosting providers. Some of these turned out to be quite uninformed regarding the amount of criminal abuse in their own networks. One way the market can regulate itself is by making blacklist data readily available to the hosting providers, so that they can monitor how often resources in their network are reported. Once the providers have such a monitoring system in place, they can adjust their abuse handling given the types of abuse noted. Our analysis found that hosting providers' structural properties or exposure had greatest effect regarding phishing abuse compared to C&C abuse.

Reactive security measures taken by hosting providers, such as the takedown time of abused domains (in our case domains used to host C&C), seemed to play a less significant role in reducing abuse than proactive measures taken by providers, such as patching. With this in mind, our results suggest that providers would suffer much less abuse in their networks if they invested time and effort in proactive measures, such as patching server-side and client-side applications and setting up domain level security mechanisms such as CSP or

`HTTPOnly` cookie policy. Motivating hosting providers to adopt such mechanisms remains the biggest challenge. This aspect can be further addressed through the use of hierarchy.

8.2.2 Hierarchy

The classic hierarchical governance mechanism is the tightening of state regimes. State-specific instruments can play a crucial role in giving hosting providers incentive to act. Yet, we identified some 45,000 hosting providers operating in more than 150 states worldwide. Regulating the entire population of hosting providers, some with infrastructure in multiple countries with different policies, would therefore be very onerous indeed. Instead, state interventions could focus on incentivizing a targeted subset of providers to act to improve security. These need not be formal regulations; rather, they could take the form of stimulation mechanisms. Below are a few examples:

(i) *Enforcing criminal law.* Criminal law enforcement could actively pursue the crime facilitating segment of the hosting market. Our research distinguished between 'best performers' and 'worst performers' in terms of both the abuse concentrations in their networks and their proactive and reactive security measures. The police could use the special instruments they have to punish or take down providers within the worst performing group, found to consistently facilitate criminal activities, wittingly or unwittingly.

(ii) *Rewarding.* Fiscal measures or financial awards, such as tax cuts and subsidies, could create economic incentives for providers willing to implement the required security measures. Subsidies could be offered for proactive security measures or for reactive measures, such as abuse handling units (e.g., abuse.io). Similarly, tax cuts could be granted to providers willing to put certain security measures in place, such as contributing data to abuse.io, or complying with particular security recommendations. The current study's results provide a starting point for such measures.

(iii) *Information transparency.* Law enforcement and administrative regulators could use information as an instrument to motivate hosting providers to increase their network hygiene. By promoting transparency regarding the actual security levels in the networks of hosting providers, governments can stimulate improved security levels while raising awareness of the need to reduce the overall harm that insecurity inflicts on society. This could be done, for example, by utilizing performance metrics or by developing websites containing price/security comparisons of providers. The methodologies presented in this thesis provide a starting point for generating such measures.

(iv) *Hybrid.* National law enforcement could adopt the so-called throffr approach. This is when a unilateral threat is made in parallel with an offer to negotiate [248]. For example, the police might threaten the worst performing providers with legal action, while at the same time offering them an opportunity to use a third-party research center to monitor abuse concentrations in their networks. One such center is AbuseHub of the Dutch Abuse Information Exchange. Law enforcement could instigate a flying start in cybersecurity solely by announcing to all hosting providers that it will be employing a third party to measure providers' performance based on abuse and identify providers that facilitate cybercrime, wittingly or unwittingly. This would give hosting providers a concrete reason to improve their security practices and start collaborating with the regulator.

8.2.3 Network

Network governance is based on interdependence and repeated interactions among network members such as public-private partnerships, or unilateral actions in the absence of an overarching authority [22, 249]. Yet, it would be very challenging to maintain the trust needed for reciprocal interactions across the huge network of hosting providers scattered around the globe.

Our results suggest that the majority of abuse is concentrated in a minority of providers. Therefore, measures that require collaboration and unilateral action targeting the hosting market could focus on strategic subsets of providers. These can then effectuate substantial improvement in the overall security of the market. Meaningful subsets would be, for example, the 1,485 providers found to own up to 80% of the hosting infrastructure, or the 8,400 providers found to host up to 80% of all phishing abuse in their networks.

Focusing on such narrower groups, national law enforcement entities, such as police high-tech crime units, could get involved indirectly in building security routines, using the so-called *shadow of hierarchy* [248]. That is, when a regulator uses a credible threat of unilateral intervention, without actually implementing it, to change actors' perception of their general gain and loss. Although these threats might be imposed based on generic or ambiguous existing laws, the possibility of being punished through reputation damage vis-a-vis peers and customers could push hosting providers towards self-regulation of the market and change their position in the game. Anecdotal evidence from ISPs suggests that such approach could be effective.

We found that identifying 'culprits' or bad performers and comparing security performances based on abuse observations was informative only if providers'

structural properties and effort-related differences were accounted for. This provides useful input for regulators aiming to identify bad performers.

8.2.4 Community

Communities are normally groups of actors with a common identity (geography, culture, or common interest), which makes it easier for them to develop joint norms. In governance of the hosting market, communities have so far been most active in areas such as development of best practices and performance of security measurements. The Messaging, Malware and Mobile Anti-Abuse Working Group (M3AAWG) is an example of a well-established community related to the hosting market [9]. However, best practices, such as those formulated by M3AAWG, have not been very effective, due to the lack of empirically-based knowledge about hosting providers and the market they operate in. Using our results regarding salient properties of the hosting market to inform best practices could evoke more effective implementation.

On a more general level, our quantitative models could increase understanding among regulators, law enforcement, and providers regarding the distribution of security threats and factors that may influence it.

8.3 Future Work

Each analytic chapter in this dissertation concluded by enumerating study limitations and potential research directions. Here, we discuss them briefly in their broader context.

The limitations of this research can be categorized into three general groups: those related to the data used, those related to the methodological choices made, and those related to the theoretical background employed.

8.3.1 Data limitations

While the data limitations encountered were varied, all could ultimately be categorized as measurement errors. Data obtained from third-party sources invariably brought corresponding data quality constraints. Some foremost examples were encountered in (i) WHOIS records, (ii) DNSDB passive DNS data, (iii) data regarding uptime of malicious domains, (iv) data collected using black-box third-party resources such as WPscan, and (v) passive and active measurements taken ourselves using our own tools. Future research could build a feedback cycle to improve the quality of such data. Data quality could also be improved via

collaboration between researchers and the industry partners that maintain data sources. Examples of such sources are the Regional Internet Registries (RIRs), Farsight Security, and the various third-party organizations that maintain abuse blacklists and blocklists.

In specific cases, we lacked reliable information about the data collection methodologies used for third-party datasets. More information on such data and the ways they are collected would help researchers establish a greater degree of accuracy and better understand the patterns observed in them. This, of course, would be much easier if researchers and abuse data providers worked more closely with one another. In addition, the thesis used several data sources that are not fully open to public, such as DNSDB, APWG, and the Zeus data. Although this is a limitation, one should note that these datasets are being used by many other researchers in the field of cybersecurity. Therefore, there is a lot of triangulation or corroboration of people using the same data or similarly collected data, which gives some check and balances to overly confident claims based on a black box dataset.

8.3.2 Methodological limitations

These include issues related to statistical methods, sampling data points, independent variables, metric definitions, aggregation levels, and data sources, among others. During the course of this dissertation, we proposed use of a diverse set of statistical approaches such as *Generalized Linear Models*, *Statistical Twins* and *Latent Factor Analysis*. Future work, can however, extent these methods to even more complex approaches.

Moreover, we were limited to snapshot data for measuring the security levels of hosting providers. For future research, longitudinal measurements are recommended regarding the patching levels of providers, to provide a better indicator for providers' security efforts. Longitudinal measurements are also recommended for direct reactive and proactive security efforts, such as setting up certain technical measures and establishing better abuse handling units. They are also advised for indirect measures such as user awareness campaigns and nudging efforts

Our use of specific abuse data sources could, to some extent, limit the generalizability of our results. Our methodology, however, is independent of the abuse data sources, and hence can be generalized to other abuse types. A natural progression of our studies would be to expand our methodology using other and more specific abuse data types, such as materials showing child sexual abuse and botnet data.

This research was based on a quantitative analysis of empirical data. Thus, the data provided the starting point for our observations regarding incentives and the behavior of hosting providers, as well as regarding the security measures taken by providers. These, however, were constrained by our own interpretations of the data, which could also be subject to error. Although we shared our results with a few well-known hosting providers as well as with law enforcement representatives during the course of the research, we did not systematically include these interactions as a part of our methodology. To reduce interpretation errors and improve insights regarding the hosting providers and the market they operate in, future research could utilize mixed methods. The quantitative results would thus be combined with qualitative observations regarding behavioral, social, and regulatory aspects of the hosting market.

On a more general level, future work could extend our research approach by including perspectives and disciplines touched upon only briefly in this dissertation. For instance, the hosting market could be investigated from a multi-actor perspective. That is, beyond the hosting providers, other key actors within this economic space could also be studied, including end-users, regulators, and criminals alongside their interactions. Another interesting dimension that could be further explored in future work is the effects of interventions by different regulators, such as law enforcement entities and private companies, on abuse concentrations.

8.3.3 Theoretical limitations

We focused on theories related to web security and Internet measurement, rather than concepts of economics and governance. This was mainly because technical measurements were required as an essential first step, to shed light on the nature and the scope of the security problem in the hosting market. The aim here was to initiate a fine-tuning of not only technical solutions, but also of solutions related to governance in this market.

In sum, the research presented in this dissertation contributes new insights regarding hosting providers and the market they operate in. At the same time, it offers a methodology for measuring hosting providers' security levels. Finally, it provides approaches and recommendations for improving the security practices of hosting providers. These results, alongside future research, hold the promise of effectively reducing the frequency of online incidents such as those cited at the start of this dissertation, ultimately diminishing their effects on society.

References

[1] The Telegraph, "How to avoid the Google Docs phishing attack and what to do if you're a victim," http://www.telegraph.co.uk/technology/2017/05/04/avoid-google-docs-phishing-attack/, 2017.

[2] E. Kovacs, "Hackers Target Prominent Chinese-Language News Sites," http://www.securityweek.com/hackers-target-prominent-chinese-language-news-sites, 2017.

[3] B. Krebs, "Who is the GovRAT Author and Mirai Botmaster 'Bestbuy'?" https://krebsonsecurity.com/2017/07/who-is-the-govrat-author-and-mirai-botmaster-bestbuy/, 2017.

[4] CVE Details, "Browse Vulnerabilities By Date," https://www.cvedetails.com/browse-by-date.php, 2017.

[5] M. van Eeten, H. Asghari, J. M. Bauer, and S. Tabatabaie, "Internet service providers and botnet mitigation: A fact-finding study on the dutch market," Delft University of Technology, 2011.

[6] M. Van Eeten, J. M. Bauer, H. Asghari, S. Tabatabaie, and D. Rand, "The role of internet service providers in botnet mitigation an empirical analysis based on spam data," TPRC, 2010.

[7] K. Perset, "The Economic and Social Role of Internet Intermediaries," http://www.oecd.org/dataoecd/49/4/44949023.pdf, 2010.

[8] European Network and Information Security Agency (ENISA), "Involving Intermediaries in Cyber-security Awareness Raising," goo.gl/ogxxDH, 2012.

[9] M3AAWG, "M3AAWG Anti-Abuse Best Common Practices for Hosting and Cloud Service Providers," https://www.m3aawg.org/sites/default/files/document/M3AAWG_Hosting_Abuse_BCPs-2015-03.pdf, 2015.

[10] D. Canali, D. Balzarotti, and A. Francillon, "The role of web hosting providers in detecting compromised websites," in Proceedings of the 22nd international conference on World Wide Web (WWW), 2013, pp. 177–188.

[11] D. Bradbury, "Testing the defences of bulletproof hosting companies," Network Security, vol. 2014, no. 6, pp. 8–12, 2014.

[12] D. Mahjoub, "Behaviors and patterns of bulletproof and anonymous hosting providers." USENIX Association, 2017.

[13] R. Anderson, "Why information security is hard-an economic perspective," in proceedings 17th Annual Computer Security Applications Conference (ACSAC). IEEE, 2001, pp. 358–365.

[14] R. Anderson and T. Moore, "The economics of information security," Science, vol. 314, no. 5799, pp. 610–613, 2006.

[15] T. Moore, R. Clayton, and R. Anderson, "The economics of online crime," The Journal of Economic Perspectives, vol. 23, no. 3, pp. 3–20, 2009.

[16] L. J. Camp and C. Wolfram, "Pricing security," in Proceedings of the CERT Information Survivability Workshop, 2000, pp. 31–39.

[17] G. A. Akerlof, "The market for" lemons": Quality uncertainty and the market mechanism," The quarterly journal of economics, pp. 488–500, 1970.

[18] D. Wood and B. Rowe, "Assessing home internet users demand for security: Will they pay isps?" 2011.

[19] StopBadware. (2011) Best practices for web hosting providers. [Online]. Available: https://www.stopbadware.org/files/best-practices-responding-to-badware-reports.pdf

[20] SANS, "A practical methodology for implementing a patch management process," https://www.sans. org/reading-room/whitepapers/bestprac/practical-methodology-implementing-patch-management-process-1206, 2003.

[21] T. Tenbensel, "Multiple modes of governance: Disentangling the alternatives to hierarchies and markets," Public Management Review, vol. 7, no. 2, pp. 267–288, 2005.

[22] A. Kuerbis, "Mapping the cybersecurity institutional landscape," Digital Policy, Regulation and Governance, vol. forthcoming, 2017.

[23] M. van Eeten and M. van Eeten, "Patching security governance: an empirical view of emergent governance mechanisms for cybersecurity," Digital Policy, Regulation and Governance, vol. 19, no. 6, pp. 429–448, 2017.

[24] M. Howlett, "Governance modes, policy regimes and operational plans: A multi-level nested model of policy instrument choice and policy design," Policy Sciences, vol. 42, no. 1, pp. 73–89, 2009.

[25] T. Tenbensel, "Multiple modes of governance," Public Management Review, vol. 7, no. 2, pp. 267–288, 2005. [Online]. Available: http://dx.doi.org/10.1080/14719030500091566

[26] T. sch, P. Chen, N. Nikiforakis, L. Desmet, and W. Joosen, "Large-scale security analysis of the web: Challenges and findings," in Trust and Trustworthy Computing. Springer, 2014, pp. 110–126.

[27] E. Bursztein, B. Benko, D. Margolis, T. Pietraszek, A. Archer, A. Aquino, A. Pitsillidis, and S. Savage, "Handcrafted fraud and extortion: Manual account hijacking in the wild," in Proceedings of the 2014 Conference on Internet Measurement Conference (IMC). ACM, 2014, pp. 347–358.

[28] V. Garg and L. J. Camp, "Macroeconomic analysis of malware," in Network & Distributed System Security Symposium (NDSS). The Internet Society, 2013, pp. 1–3.

[29] C. Wilcox, C. Papadopoulos, and J. Heidemann, "Correlating spam activity with IP address characteristics," in INFOCOM IEEE Conference on Computer Communications Workshops, 2010. IEEE, 2010, pp. 1–6.

[30] S. Hao, N. Feamster, and R. Pandrangi, "Monitoring the initial dns behavior of malicious domains," in Proceedings of the 2011 ACM conference on Internet measurement conference (IMC). ACM, 2011, pp. 269–278.

[31] S. A. Mirheidari, S. Arshad, S. Khoshkdahan, and R. Jalili, "Two novel server-side attacks against log file in shared web hosting servers," in 2012 International Conference for Internet Technology And Secured Transactions. IEEE, 2012, pp. 318–323.

[32] C. Wagner, J. François, R. State, A. Dulaunoy, T. Engel, and G. Massen, "ASMATRA: Ranking ASs providing transit service to malware hosters," in IFIP/IEEE International Symposium on Integrated Network Management (IM). IFIP/IEEE, 2013, pp. 260–268.

[33] M. Konte, R. Perdisci, and . N. Feamster, "ASwatch: An AS Reputation System to Expose Bulletproof Hosting ASes," in Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, ser. SIGCOMM '15. ACM, 2015, pp. 625–638. [Online]. Available: http://doi.acm.org/10.1145/2785956.2787494

[34] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, and E. Kirda, "Fire: Finding rogue networks," in Computer Security Applications Conference. IEEE, 2009, pp. 231–240.

[35] Y. Liu, A. Sarabi, J. Zhang, P. Naghizadeh, M. Karir, M. Bailey, and M. Liu, "Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents," in USENIX Security Symposium, 2015, pp. 1009–1024.

[36] HostExploit, http://hostexploit.com, 2017.

[37] Netcraft, https://www.netcraft.com/internet-data-mining/hosting-analysis/, 2017.

[38] A. Noroozian, M. Ciere, M. Korczyński, S. Tajalizadehkhoob, and M. Eeten, "Inferring the Security Performance of Providers from Noisy and Heterogenous Abuse Datasets," in 16th Workshop on the Economics of Information Security, June 2017. [Online]. Available: http://weis2017.econinfosec.org/wp-content/uploads/sites/3/2017/05/WEIS_2017_paper_60.pdf

[39] N. Nikiforakis, W. Joosen, and M. Johns, "Abusing locality in shared web hosting," in Proceedings of the Fourth European Workshop on System Security. ACM, 2011, p. 2.

[40] D. Molnar and S. E. Schechter, "Self hosting vs. cloud hosting: Accounting for the security impact of hosting in the cloud." in the Annual Workshop on the Economics of Information Security (WEIS), 2010.

[41] (2017) VPS Hosting: How It Works, How-to Choose, Recommendations and Discounts. https://www.webhostingsecretrevealed.net/vps-hosting-guide/.

[42] X. Cai, J. Heidemann, B. Krishnamurthy, and W. Willinger, "Towards an AS-to-organization map," in Proceedings of the 2010 ACM conference on Internet Measurement Conference (IMC). ACM, 2010, pp. 199–205.

[43] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in USENIX Security Symposium. USENIX Association, 2014, pp. 625–640.

[44] M. Vasek, J. Wadleigh, and T. Moore, "Hacking Is Not Random: A Case-Control Study of Webserver-Compromise Risk," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 206–219, 2016.

[45] A. P. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi, "A layout-similarity-based approach for detecting phishing pages," in Third International SecureComm. IEEE, 2007, pp. 454–463.

[46] C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," in Network & Distributed System Security Symposium (NDSS). The Internet Society, 2010, pp. 1–5.

[47] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," in Network & Distributed System Security Symposium (NDSS). The Internet Society, 2011, pp. 1–17.

[48] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Félegyházi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu et al., "Click trajectories: End-to-end analysis of the spam value chain," in IEEE Symposium on Security and Privacy (SP). IEEE, 2011, pp. 431–446.

[49] G. Aaron and R. Rasmussen. (2015) Anti-Phishing Working Group (APWG) Global Phishing Survey: Trends and Domain Name Use in 2H2014. http://docs.apwg.org/reports/APWG_Global_Phishing_Report_2H_2014.pdf.

[50] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck, "Understanding the domain registration behavior of spammers," in Proceedings of the 2013 ACM conference on Internet Measurement Conference (IMC). ACM, 2013, pp. 63–76.

[51] H. Liu, K. Levchenko, M. Félegyházi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage, "On the Effects of Registrar-level Intervention," in LEET, 2011.

[52] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, "Highly resilient peer-to-peer botnets are here: An analysis of gameover zeus," in Malicious and Unwanted Software:" The Americas"(MALWARE), 2013 8th International Conference on. IEEE, 2013, pp. 116–123.

[53] C. Rossow, D. Andriesse, T. Werner, B. Stone-Gross, D. Plohmann, C. J. Dietrich, and H. Bos, "Sok: P2pwned-modeling and evaluating the resilience of peer-to-peer botnets," in Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013, pp. 97–111.

[54] D. Y. Wang, S. Savage, and G. M. Voelker, "Juice: A longitudinal study of an seo botnet." in Network & Distributed System Security Symposium (NDSS), 2013.

[55] Z. Durumeric, M. Bailey, and J. A. Halderman, "An internet-wide view of internet-wide scanning." in USENIX Security Symposium. USENIX Association, 2014, pp. 65–78.

[56] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 pound gorilla: The rise and decline of ntp ddos attacks," in Proceedings of the 2014 Conference on Internet Measurement Conference (IMC). ACM, 2014, pp. 435–448.

[57] M. Karami, Y. Park, and D. McCoy, "Stress testing the booters: understanding and undermining the business of ddos services," in Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2016, pp. 1033–1043.

[58] C. Rossow, "Amplification hell: Revisiting network protocols for ddos abuse." in Network & Distributed System Security Symposium (NDSS), 2014.

[59] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from hell? reducing the impact of amplification ddos attacks." in USENIX Security Symposium. USENIX Association, 2014, pp. 111–125.

[60] J. J. Santanna, R. Durban, A. Sperotto, and A. Pras, "Inside booters: an analysis on operational databases," in IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015, pp. 432–440.

[61] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM, 2006, pp. 581–590.

[62] T. Moore and R. Clayton, "An empirical analysis of the current state of phishing attack and defence." in WEIS, 2007.

[63] ——, "Evil searching: Compromise and recompromise of internet hosts for phishing," in Financial Cryptography and Data Security. Springer, 2009, pp. 256–272.

[64] N. P. P. Mavrommatis and M. A. R. F. Monrose, "All your iframes point to us," in Proceedings of the 17th USENIX Security Symposium (SEC'08). USENIX Association, 2008, pp. 1–15.

[65] N. Leontiadis, T. Moore, and N. Christin, "Measuring and Analyzing Search-Redirection Attacks in the Illicit Online Prescription Drug Trade," in USENIX Security Symposium. USENIX Association, 2011.

[66] S. Alrwais, K. Yuan, E. Alowaisheq, X. Liao, A. Oprea, X. Wang, and Z. Li, "Catching predators at watering holes: finding and understanding strategically compromised websites," in Proceedings of the 32nd Annual Conference on Computer Security Applications. ACM, 2016, pp. 153–166.

[67] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni et al., "DROWN: breaking TLS using SSLv2," in USENIX Security Symposium. USENIX Association, 2016.

[68] N. AlFardan, D. J. Bernstein, K. G. Paterson, B. Poettering, and J. C. N. Schuldt, "On the security of rc4 in tls," in USENIX Security Symposium. USENIX Association, 2013, pp. 305–320.

[69] M. Vanhoef and F. Piessens, "All your biases belong to us: Breaking rc4 in wpa-tkip and tls," in USENIX Security Symposium. USENIX Association, 2015, pp. 97–112.

[70] N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. Van Acker, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "You are what you include: large-scale evaluation of remote javascript inclusions," in Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012, pp. 736–747.

[71] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, N. Modadugu et al., "The ghost in the browser: Analysis of web-based malware." HotBots, vol. 7, pp. 4–4, 2007.

[72] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: a fast filter for the large-scale detection of malicious web pages," in Proceedings of the 20th international conference on World wide web (WWW). ACM, 2011, pp. 197–206.

[73] L. Invernizzi and P. M. Comparetti, "Evilseed: A guided approach to finding malicious web pages," in Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012, pp. 428–442.

[74] K. Borgolte, C. Kruegel, and G. Vigna, "Delta: automatic identification of unknown web-based infection campaigns," in Proceedings of the 2013 ACM conference on Computer & communications security. ACM, 2013, pp. 109–120.

[75] Anti-Phishing Working Group, http://www.antiphishing.org, 2016.

[76] PishTank, https://www.phishtank.com, 2017.

[77] ZeusTracker, https://www.abuse.ch, 2017.

[78] MalwareDomains, http://www.malwaredomains.com, 2017.

[79] Dshield, https://secure.dshield.org, 2017.

[80] A. Pitsillidis, C. Kanich, G. M. Voelker, K. Levchenko, and S. Savage, "Taster's choice: a comparative analysis of spam feeds," in Proceedings of the 2012 ACM conference on Internet Measurement Conference (IMC). ACM, 2012, pp. 427–440.

[81] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in Research in Attacks, Intrusions and Defenses. Springer, 2014, pp. 1–21.

[82] L. Metcalf and J. M. Spring, "Everything You Wanted to Know About Blacklists But Were Afraid to Ask," CERT Network Situational Awareness Group, Tech. Rep., 2013.

[83] VERIS Community Database, http://veriscommunity.net/index.html., 2017.

[84] Data Breach Investigations Reports, http://www.verizonenterprise.com/DBIR/, 2017.

[85] A. J. Kalafut, C. A. Shue, and M. Gupta, "Malicious hubs: detecting abnormally malicious autonomous systems," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–5.

[86] C. Grier, L. Ballard, J. Caballero, N. Chachra, C. J. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis et al., "Manufacturing compromise: the emergence of exploit-as-a-service," in Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012, pp. 821–832.

[87] D. McCoy, A. Pitsillidis, G. Jordan, N. Weaver, C. Kreibich, B. Krebs, G. M. Voelker, S. Savage, and K. Levchenko, "Pharmaleaks: Understanding the business of online pharmaceutical affiliate programs," in USENIX Security symposium. USENIX Association, 2012, pp. 1–1.

[88] M. A. Kuypers, T. Maillart, and E. Pate-Cornell, "An empirical analysis of cyber security incidents at a large organization," Department of Management Science and Engineering, Stanford University, School of Information, UC Berkeley, vol. 30, 2016.

[89] H. Asghari, M. Ciere, and M. J. Van Eeten, "Post-mortem of a zombie: conficker cleanup after six years," in USENIX Security Symposium. USENIX Association, 2015, pp. 1–16.

[90] J. Zhang, Z. Durumeric, M. Bailey, M. Liu, and M. Karir, "On the Mismanagement and Maliciousness of Networks," in Network & Distributed System Security Symposium (NDSS). The Internet Society, 2014, pp. 1–12.

[91] WhiteHat. (2016) Website Security Statistics Report. https://www.whitehatsec.com/resource/stats.html.

[92] Symantec. (2016) Internet Security Threat Report. https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf.

[93] Edgescan. (2016) 2016 Vulnerability Statistics Report. https://www.edgescan.com/assets/docs/reports/2016-edgescan-stats-report.pdf.

[94] The Web Application Security Consortium. (2007) Web Application Security Statistics. http://projects.webappsec.org/w/page/13246989/Web%20Application%20Security%20Statistics.

[95] A. Alarifi, M. Alsaleh, and A. Al-Salman, "Security analysis of top visited arabic web sites," in Advanced Communication Technology (ICACT), 2013 15th International Conference on. IEEE, 2013, pp. 173–178.

[96] S. Kals, E. Kirda, C. Kruegel, and N. Jovanovic, "Secubat: a web vulnerability scanner," in Proceedings of the 15th international conference on World Wide Web. ACM, 2006, pp. 247–256.

[97] S. Lekies, B. Stock, and M. Johns, "25 million flows later: large-scale detection of dom-based xss," in Proceedings of the 2013 ACM Conference on Computer and Communications Security. ACM, 2013, pp. 1193–1204.

[98] S. Van Acker, D. Hausknecht, and A. Sabelfeld, "Measuring login webpage security," in Proceedings of the 32st Annual ACM Symposium on Applied Computing, ser. SAC'17. ACM, 2017.

[99] A. Doupé, L. Cavedon, C. Kruegel, and G. Vigna, "Enemy of the state: A state-aware black-box web vulnerability scanner." in USENIX Security Symposium, vol. 14. USENIX Association, 2012.

[100] M. Yar, "The novelty of 'cybercrime' an assessment in light of routine activity theory," European Journal of Criminology, vol. 2, no. 4, pp. 407–427, 2005.

[101] G. C. Moura, Internet bad neighborhoods. Giovane Cesar Moreira Moura, 2013, no. 12.

[102] A. Sarabi, P. Naghizadeh, Y. Liu, and M. Liu, "Prioritizing Security Spending: A Quantitative Analysis of Risk Distributions for Different Business Profiles," in the Annual Workshop on the Economics of Information Security (WEIS)(June 2015), 2015.

[103] Y.-G. Kim, S. Cho, J.-S. Lee, M.-S. Lee, I. H. Kim, and S. H. Kim, "Method for evaluating the security risk of a website against phishing attacks," in International Conference on Intelligence and Security Informatics. Springer, 2008, pp. 21–31.

[104] M. Zhao, J. Grossklags, and K. Chen, "An exploratory study of white hat behaviors in a web vulnerability disclosure program," in Proceedings of the 2014 ACM Workshop on Security Information Workers. ACM, 2014, pp. 51–58.

[105] Usher,Mark and Kessem, Limor and Steigemann,Martin, "Relying on Data to Mitigate the Risk of WordPress Website Hijacking," https://securityintelligence.com/relying-on-data-to-mitigate-the-risk-of-wordpress-website-hijacking/, 2017.

[106] J. P. John, F. Yu, Y. Xie, M. Abadi, and A. Krishnamurthy, "Searching the searchers with searchaudit." in USENIX Security Symposium. USENIX Association, 2010, pp. 127–142.

[107] L. Weichselbaum, M. Spagnuolo, S. Lekies, and A. Janc, "CSP is dead, long live CSP! on the insecurity of whitelists and the future of content security policy," in Proceedings of the 2016 ACM Conference on Computer and Communications Security (CCS). ACM, 2016, pp. 1376–1387.

[108] X. Pan, Y. Cao, S. Liu, Y. Zhou, Y. Chen, and T. Zhou, "Cspautogen: Black-box enforcement of content security policy upon real-world websites," in Proceedings of the 2016 ACM Conference on Computer and Communications Security (CCS). ACM, 2016, pp. 653–665.

[109] R. Anderson, C. Barton, R. Böhme, R. Clayton, M. J. Van Eeten, M. Levi, T. Moore, and S. Savage, "Measuring the cost of cybercrime," in The economics of information security and privacy. Springer, 2013, pp. 265–300.

[110] V. Subrahmanian, M. Ovelgonne, T. Dumitras, and B. A. Prakash, The Global Cyber-Vulnerability Report. Springer, 2015.

[111] G. Mezzour, K. M. Carley, and L. R. Carley, "Global variation in attack encounters and hosting," in Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp. ACM, 2017, pp. 62–73.

[112] V. Garg, T. Koster, and L. J. Camp, "Cross-country analysis of spambots," EURASIP Journal on Information Security, vol. 2013, no. 1, p. 3, 2013.

[113] V. Garg, L. J. Camp, and C. Kanich, "Analysis of ecrime in crowd-sourced labor markets: Mechanical turk vs. freelancer," in The economics of information security and privacy. Springer, 2013, pp. 301–321.

[114] A. Noroozian, M. Korczyński, S. TajalizadehKhoob, and M. van Eeten, "Developing Security Reputation Metrics for Hosting Providers," in 8th Workshop on Cyber Security Experimentation and Test (CSET 15). USENIX Association, 2015.

[115] C. A. Shue, A. J. Kalafut, and M. Gupta, "Abnormally malicious autonomous systems and their internet connectivity," IEEE/ACM Transactions on Networking (TON), vol. 20, no. 1, pp. 220–230, 2012.

[116] Netcraft. (2015) Hosting provider server count. [Online]. Available: http://www.netcraft.com/internet-data-mining/hosting-provider-server-count/

[117] DNSDB, https://www.dnsdb.info, 2016.

[118] M. Korczyński, S. Tajalizadehkhoob, A. Noroozian, M. Wullink, C. Hesselman, and M. van Eeten, "Reputation metrics design to improve intermediary incentives for security of tlds," in Proceedings of IEEE European Symposium on Security and Privacy. IEEE, 2017, pp. 1–15.

[119] K. Elliott, "Who, What, Where, When, and Why of WHOIS: Privacy and Accuracy Concerns of the WHOIS Database," SMU Science & Technology Law Review, vol. 12, p. 141, 2008.

[120] MaxMind, https://www.maxmind.com, 2016.

[121] H. Asghari, M. J. van Eeten, and J. M. Bauer, "Economics of fighting botnets: Lessons from a decade of mitigation," IEEE Security & Privacy, vol. 13, no. 5, pp. 16–23, 2015.

[122] G. Aaron and R. Rasmussen. (2015) Anti-Phishing Working Group (APWG) Global Phishing Survey: Trends and Domain Name Use in 1H2014. http://docs.apwg.org/reports/APWG_Global_Phishing_Report_1H_2014.pdf.

[123] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, pp. 100–108, 1979.

[124] L. Kaufman and P. J. Rousseeuw, Partitioning Around Medoids (Program PAM). John Wiley & Sons, Inc., 2008, pp. 68–125.

[125] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," Journal of the American Statistical Association, vol. 97, no. 458, pp. pp. 611–631, 2002.

[126] F. Murtagh, Multidimensional clustering algorithms, 1985.

[127] G. Brock, V. Pihur, S. Datta, and S. Datta, "clValid, an R package for cluster validation," Journal of Statistical Software, 2011.

[128] C. GMBH., http://www.cyscon.de, 2016.

[129] T. Moore and R. Clayton, "Examining the impact of website take-down on phishing," in Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit. ACM, 2007, pp. 1–13.

[130] J. Nazario and T. Holz, "As the net churns: Fast-flux botnet observations," in International Conference on Malicious and Unwanted Software. IEEE, 2008, pp. 24–31.

[131] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," Journal of the American statistical association, vol. 53, no. 282, pp. 457–481, 1958.

[132] R. Peto and J. Peto, "Asymptotically efficient rank invariant test procedures," Journal of the Royal Statistical Society. Series A (General), pp. 185–207, 1972.

[133] D. Mahjoub, "Sweeping the IP space: The hunt for evil on the internet." Virus Bulletin Conference, 2014. [Online]. Available: https://www.virusbtn.com/pdf/conference/vb2014/VB2014-Mahjoub.pdf

[134] O. Cetin, M. H. Jhaveri, C. Gañán, M. van Eeten, and T. Moore, "Understanding the Role of Sender Reputation in Abuse Reporting and Cleanup," in Proceedings of the 14th WEIS. WEIS, 2015, pp. 1–15.

[135] A. Nappa, M. Z. Rafique, and J. Caballero, "Driving in the cloud: An analysis of drive-by download operations and abuse reporting," in Proceedings of the 10th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, ser. DIMVA'13. Springer-Verlag, 2013, pp. 1–20.

[136] C. Gañán, O. Cetin, and M. van Eeten, "An empirical analysis of Zeus C&C lifetime," in Proceedings of the 10th ACM Symposium (ASIA CCS). ACM, 2015, pp. 97–108.

[137] M. Goncharov, "Criminal Hideouts for Lease: Bulletproof Hosting Services," http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-criminal-hideouts-for-lease.pdf, 2015.

[138] McAfee Intel Security, "Botnet Control Servers Span the Globe," https://blogs.mcafee.com/mcafee-labs/botnet-control-servers-span-the-globe, 2013.

[139] Nederlandse Omroep Stichting, "Nederland paradijs cybercriminelen," http://nos.nl/artikel/469969-nederland-paradijs-cybercriminelen.html, 2013.

[140] Dutch Hosting Provider Association, "Nederland paradijs voor internet criminelen?" https://www.dhpa.nl/nederland-paradijs-voor-internet-criminelen.html, 2013.

[141] R. Clayton, T. Moore, and N. Christin, "Concentrating Correctly on Cybercrime Concentration," in Proceedings of the 14th Annual Workshop on the Economics of Information Security. WEIS, 2015, pp. 1–16.

[142] S. Tajalizadehkhoob, M. Korczynski, A. Noroozian, C. Ganán, and M. van Eeten, "Apples, oranges and hosting providers: Heterogeneity and security in the hosting market," in Network Operations and Management Symposium (NOMS). IEEE/IFIP, 2016, pp. 289–297.

[143] S. Liu, I. Foster, S. Savage, G. M. Voelker, and L. K. Saul, "Who is. com? Learning to Parse WHOIS Records," in Proceedings of the 2015 ACM conference on Internet Measurement Conference (IMC). ACM, 2015, pp. 369–380.

[144] X. Dimitropoulos, D. Krioukov, G. Riley, and k. claffy, "Revealing the Autonomous System Taxonomy: The Machine Learning Approach," in Passive and Active Network Measurement Workshop (PAM), 2006, pp. 91–100.

[145] International Telecommunication Union (ITU), "Measuring the Information Society Report 2014," https://www.itu.int/en/ITU-D/Statistics/Documents/publications/mis2014/MIS2014_without_Annex_4.pdf, 2014.

[146] Web-Archive, http://archive.org/web, 2016.

[147] W. Team, "Succuri WPscan," https://wpscan.org, 2016.

[148] A. C. Cameron and P. K. Trivedi, "Regression-based tests for overdispersion in the Poisson model," Journal of Econometrics, vol. 46, no. 3, pp. 347–364, 1990.

[149] H. Heinzl and M. Mittlböck, "Pseudo R-squared measures for Poisson regression models with over-or underdispersion," Computational statistics & data analysis, vol. 44, no. 1, pp. 253–271, 2003.

[150] M. Mittlböck, "Calculating adjusted r(2) measures for Poisson regression models," Computer Methods and Programs in Biomedicine, vol. 68, no. 3, pp. 205–214, 2002.

[151] A. C. Cameron and P. K. Trivedi, Regression analysis of count data. Cambridge university press, 2013, vol. 53.

[152] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," ACM SIGCOMM Computer Communication Review, vol. 36, no. 4, pp. 291–302, 2006.

[153] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. De Shon, and J. Kadane, "Using uncleanliness to predict future botnet addresses," in Proceedings of the 2007 ACM conference on Internet Measurement Conference (IMC). ACM, 2007, pp. 93–104.

[154] A. Kleiner, P. Nicholas, and K. Sullivan, "Linking Cybersecurity Policy and Performance," Microsoft Trustworthy Computing, vol. 1, no. 1, pp. 1–20, 2013.

[155] S. He, G. M. Lee, J. S. Quarterman, Q. Creations, and A. B. Whinston, "Cybersecurity Policies Design and Evaluation: Evidence from a Large-Scale Randomized Field Experiment," in Proceedings of the 14th Annual Workshop on the Economics of Information Security. WEIS, 2015, pp. 1–50.

[156] A. Welzel, C. Rossow, and H. Bos, "On measuring the impact of DDOS botnets," in Proceedings of the Seventh European Workshop on System Security. ACM, 2014, p. 3.

[157] W. Chang, A. Mohaisen, A. Wang, and S. Chen, "Measuring botnets in the wild: Some new trends," in Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (CCS). ACM, 2015, pp. 645–650.

[158] "Microsoft Security Intelligence Report," https://www.microsoft.com/security/sir/default.aspx, 2015.

[159] S. Tajalizadehkhoob, R. Böhme, C. Gañán, M. Korczyński, and M. van Eeten, "Rotten Apples or Bad Harvest? What We Are Measuring When We Are Measuring Abuse," ACM Transactions on Internet Technology (TOIT), 2017.

[160] J. C. Botero and A. Ponce, "Rule of law index," The World Justice Project, 2010.

[161] S. Tajalizadehkhoob, H. Asghari, C. Gañán, and M. van Eeten, "Why them? extracting intelligence about target selection from zeus financial malware," in Proceedings of the 13th Annual Workshop on the Economics of Information Security. WEIS, 2014.

[162] Z. Li, A. Goyal, Y. Chen, and V. Paxson, "Automating analysis of large-scale botnet probing events," in Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS). ACM, 2009, pp. 11–22. [Online]. Available: http://doi.acm.org/10.1145/1533057.1533063

[163] W. Lu, M. Tavallaee, and A. A. Ghorbani, "Automatic discovery of botnet communities on large-scale communication networks," in Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS). ACM, 2009, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/1533057.1533062

[164] B. B. Kang, E. Chan-Tin, C. P. Lee, J. Tyra, H. J. Kang, C. Nunnery, Z. Wadler, G. Sinclair, N. Hopper, D. Dagon, and Y. Kim, "Towards complete node enumeration in a peer-to-peer botnet," in Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS). ACM, 2009, pp. 23–34. [Online]. Available: http://doi.acm.org/10.1145/1533057.1533064

[165] C. Rossow, C. Dietrich, and H. Bos, "Large-scale analysis of malware downloaders," in Proceedings of the 9th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA). Berlin, Heidelberg: Springer-Verlag, 2013, pp. 42–61.

[166] X. Han, N. Kheir, and D. Balzarotti, "The role of cloud services in malicious software: Trends and insights," in Proceedings of the 12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), vol. 9148. New York, NY, USA: Springer-Verlag New York, Inc., 2015, pp. 187–204.

[167] A. Nappa, Z. Xu, M. Z. Rafique, J. Caballero, and G. Gu, "Cyberprobe: Towards internet-scale active detection of malicious servers," in Network & Distributed System Security Symposium (NDSS), 2014, pp. 1–15.

[168] A. Caglayan, M. Toothaker, D. Drapeau, D. Burke, and G. Eaton, "Behavioral analysis of botnets for threat intelligence," Information Systems and e-Business Management, vol. 10, no. 4, pp. 491–519, 2012.

[169] W. Xu, X. Wang, and H. Xie, "New trends in fastflux networks," in Proceedings of the 16th BlackHat USA.

[170] M. H. Jhaveri, O. Cetin, C. Gañán, T. Moore, and M. V. Eeten, "Abuse reporting and the fight against cybercrime," ACM Computing Surveys (CSUR), vol. 49, no. 4, p. 68, 2017.

[171] X. Liao, C. Liu, D. McCoy, E. Shi, S. Hao, and R. Beyah, "Characterizing Long-tail SEO Spam on Cloud Web Hosting Services," in Proceedings of the 25th International Conference on World Wide Web (WWW), 2016, pp. 321–332.

[172] G. Stringhini, O. Hohlfeld, C. Kruegel, and G. Vigna, "The Harvester, the Botmaster, and the Spammer: On the Relations Between the Different Actors in the Spam Landscape," in Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIACCS). ACM, 2014, pp. 353–364. [Online]. Available: http://doi.acm.org/10.1145/2590296.2590302

[173] W. de Vries, "Hosting provider Antagonist automatically fixes vulnerabilities in customers' websites," https://www.antagonist.nl, 2012.

[174] TrendMicro, "Looking Into a Cyber-Attack Facilitator in the Netherlands," http://blog.trendmicro.com/trendlabs-security-intelligence/looking-into-a-cyber-attack-facilitator-in-the-netherlands/.

[175] N. Premchaiswadi, J. G. Williams, and W. Premchaiswadi, "A Study of an On-Line Credit Card Payment Processing and Fraud Prevention for e-Business," in World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, vol. 2009, 2009, pp. 2199–2206.

[176] E. C. Bank. (2014) Third report on card fraud. http://www.ecb.europa.eu/press/pr/date/2014/html/pr140225.en.html.

[177] Financial Fraud Action UK. (2013) Fraud the Facts 2013. http://www.financialfraudaction.org.uk/.

[178] (2013) Banque de France Rapport Annuel 2012. http://www.banque-france.fr/observatoire/telechar/2013/Rapport-annuel-2012.pdf.

[179] Nederlandse Vereniging van Banken, "Scherpe Daling Fraude Internetbankieren," 2013.

[180] R. J. Sullivan, "The changing nature of us card payment fraud: industry and public policy options," Economic Review-Federal Reserve Bank of Kansas City, vol. 95, no. 2, p. 101, 2010.

[181] J. Claessens, V. Dem, D. De Cock, B. Preneel, and J. Vandewalle, "On the security of today's online electronic banking systems," Computers & Security, vol. 21, no. 3, pp. 253–265, 2002.

[182] R. P. Jaleshgari, "Document trading online," Information Week, vol. 755, p. 136, 1999.

[183] M. Adham, A. Azodi, Y. Desmedt, and I. Karaolis, "How to attack two-factor authentication internet banking," in International Conference on Financial Cryptography and Data Security. Springer, 2013, pp. 322–328.

[184] N. Utakrit, "Review of browser extensions, a man-in-the-browser phishing techniques targeting bank customers," 2009.

[185] J. Wyke, "What is zeus?" Sophos, May, 2011.

[186] P. Kruse. (2011) Csis: Complete zeus sourcecode has been leaked to the masses. https://http://www.csis.dk/en/csis/blog/3229/.

[187] N. Falliere and E. Chien. (2009) Zeus: King of the bots.

[188] D. Macdonald. (2014) Zeus, God of Diy Botnets. http://www.fortiguard.com/legacy/analysis/zeusanalysis.html.

[189] S. Li, A.-R. Sadeghi, S. Heisrath, R. Schmitz, and J. J. Ahmad, "hpin/htan: A lightweight and low-cost e-banking solution against untrusted computers," in International Conference on Financial Cryptography and Data Security. Springer, 2011, pp. 235–249.

[190] L. F. Cranor, K. Idouchi, P. G. Leon, M. Sleeper, and B. Ur, "Are they actually any different? comparing thousands of financial institutions' privacy practices," in Proc. WEIS, vol. 13, 2013.

[191] S. J. Murdoch, S. Drimer, R. Anderson, and M. Bond, "Chip and pin is broken," in Security and Privacy (SP), 2010 IEEE Symposium on. IEEE, 2010, pp. 433–446.

[192] S. J. Murdoch and R. Anderson, "Verified by visa and mastercard securecode: or, how not to design authentication," in International Conference on Financial Cryptography and Data Security. Springer, 2010, pp. 336–342.

[193] N. Christin, S. S. Yanagihara, and K. Kamataki, "Dissecting one click frauds," in Proceedings of the 17th ACM conference on Computer and communications security. ACM, 2010, pp. 15–26.

[194] Sherstobitoff, R. (2014) Inside the World of the Citadel Trojan. http://www.mcafee.com/uk/resources/white-papers/wp-citadel-trojan.pdf.

[195] TrendMicro. (2014) Security Threats to Business, the Digital Lifestyle, and the Cloud. http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/spotlight-articles/sp-trend-micro-predictions-for-2013-and-beyond.pdf.

[196] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural change as informational cascades," Journal of political Economy, vol. 100, no. 5, pp. 992–1026, 1992.

[197] MaxMind, "MaxMind (2014) Geoip | Ip Address Location Database," http://www.maxmind.com/en/geolocation_landing, 2014.

[198] Alexa - the web information company. http://www.alexa.com.

[199] Federal Deposit Insurance Corporation, "Federal Deposit Insurance Corporation: Institution Directory," http://www2.fdic.gov/idasp/main.asp, 2014.

[200] "Open Directory Project," http://www.dmoz.org, 2014.

[201] Domingues Boscovich, R, "Microsoft and Financial Services Industry Leaders Target Cybercriminal Operations from Zeus Botnets," 2012.

[202] F-Secure, "F-Secure Threat Report H1 2012," https://http://www.f-secure.com/static/doc/labs_global/Research/Threat_Report_H2_2012.pdf, 2012.

[203] Krebs, B, "Thieves Replacing Money Mules with Prepaid Cards?" http://krebsonsecurity.com/2012/04/thieves-replacing-money-mules-with-prepaid-cards/, 2014.

[204] Rashid, F Y, "Zeus Source Code Leak Means Even More Banking Malware to Hit the Web," http://www.eweek.com/c/a/Security/Zeus-Source-Code-Leak-Means-Even-More-Banking-\Malware-to-Hit-the-Web-253343/, 2011.

[205] RSA FraudAction Research Labs , "Fraud News Flash – the Downfall of the Mighty – Zeus Trojan's Source Code Leaked and Now Available Everywhere," https://blogs.rsa.com/fraud-news-flash-\T1\textendash-the-downfall-of-the-mighty-\T1\textendash-zeus-trojan\T1\textquoterights\-source-code-leaked-and-now-available-everywhere/, 2015.

[206] D. Florêncio and C. Herley, "Where do all the attacks go?" in Economics of Information Security and Privacy III. Springer, 2013, pp. 13–33.

[207] D. Mitropoulos, V. Karakoidas, P. Louridas, and D. Spinellis, "Countering code injection attacks: a unified approach," Information Management & Computer Security, vol. 19, no. 3, pp. 177–194, 2011.

[208] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of documentation, vol. 28, no. 1, pp. 11–21, 1972.

[209] A. Karnik, S. Goswami, and R. Guha, "Detecting obfuscated viruses using cosine similarity analysis," in Modelling & Simulation, 2007. AMS'07. First Asia International Conference on. IEEE, 2007, pp. 165–170.

[210] R. K. Shahzad and N. Lavesson, "Veto-based malware detection," in Availability, Reliability and Security (ARES), 2012 Seventh International Conference on. IEEE, 2012, pp. 47–54.

[211] A. Suebsing and N. Hiransakolwong, "Feature selection using euclidean distance and cosine similarity for intrusion detection model," in Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on. IEEE, 2009, pp. 86–91.

[212] T. C. Hoad and J. Zobel, "Methods for identifying versioned and plagiarized documents," Journal of the Association for Information Science and Technology, vol. 54, no. 3, pp. 203–215, 2003.

[213] J. W. Hunt and M. MacIlroy, An algorithm for differential file comparison. Bell Laboratories New Jersey, 1976.

[214] A. Singhal, "Modern information retrieval: A brief overview," IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35–43, 2001.

[215] Baylor, K, "The Cutting Edge Is Honed: NSSLabs," https://www.nsslabs.com/reports/cutting-edge-honed-financial-malware-update, 2014.

[216] hosting.com. (2012) Best practices for architecting your hosted systems for 100% application availability. http://www.hosting.com/wp-content/uploads/2013/11/Hosting_2012-04-WP-Architect-Availability.pdf.

[217] F. Li, G. Ho, E. Kuan, Y. Niu, L. Ballard, K. Thomas, E. Bursztein, and V. Paxson, "Remedying Web Hijacking: Notification Effectiveness and Webmaster Comprehension," in Proceedings of the 25th International Conference (WWW), 2016, pp. 1009–1019.

[218] F. Li, Z. Durumeric, J. Czyz, M. Karami, M. Bailey, D. McCoy, S. Savage, and V. Paxson, "You've got vulnerability: Exploring effective vulnerability notifications," in USENIX Security Symposium. USENIX Association, 2016, pp. 1033–1050.

[219] T. Vissers, W. Joosen, and N. Nikiforakis, "Parking sensors: Analyzing and detecting parked domains," in Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS). Internet Society, 2015, pp. 53–53.

[220] D. Dittrich, E. Kenneally et al., "The menlo report: Ethical principles guiding information and communication technology research," US Department of Homeland Security, 2012.

[221] National Institute of Standards and Technology (NIST. (2017) National Vulnerability Database. https://nvd.nist.gov/vuln/search/results?adv_search=false&form_type=basic&results_type=overview&search_type=all&query=PHP5.

[222] M. Vasek, J. Wadleigh, and T. Moore, "Hacking is not random: A case-control study of webserver-compromise risk," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 206–219, March 2016.

[223] Web Technology Survays. (2016) Market share trends for content management systems for websites. https://w3techs.com/technologies/history_overview/content_management.

[224] A. Diquet, "SSLyze," https://github.com/nabla-c0d3/sslyze, 2016.

[225] RedHat, "OpenSSL CCS injection vulnerability (CVE-2014-0224)," https://access.redhat.com/articles/904433, 2014.

[226] Web Technology Survays. (2017) Usage statistics and market share of Linux for websites. https://w3techs.com/technologies/details/os-linux/all/all.

[227] S. Calzavara, A. Rabitti, and M. Bugliesi, "Content security problems?: Evaluating the effectiveness of content security policy in the wild," in Proceedings of the 2016 ACM Conference on Computer and Communications Security (CCS). ACM, 2016, pp. 1365–1375.

[228] WPBeginner, "Sucuri Review – How Sucuri Helped us Block 450,000 WordPress Attacks in 3 Months," http://www.wpbeginner.com/opinion/sucuri-review-how-sucuri-helped-us-block-450000-wordpress-attacks-in-3-months/, 2016.

[229] I. S. Intelligence. New Year, New Problems: CMS Vulnerabilites Take on 2016.

[230] B. Habing, "Exploratory factor analysis," University of South Carolina-October, vol. 15, p. 2003, 2003.

[231] S. Cheung and A. Valdes, "Malware characterization through alert pattern discovery." in 2th Usenix Workshop on Large-scale Exploits and Emergent Threats (LEET 09), 2009.

[232] J. P. Stevens, Applied multivariate statistics for the social sciences. Routledge, 2012.

[233] B. Balker. (2017) Dealing with quasi-models in R. https://cran.r-project.org/web/packages/bbmle/vignettes/quasi.pdf.

[234] cPanel. cPanel TSR-2017-0002 Full Disclosure.

[235] A. Alhuzali, B. Eshete, R. Gjomemo, and V. Venkatakrishnan, "Chainsaw: Chained automated workflow-based exploit generation," in Proceedings of the 2016 ACM Conference on Computer and Communications Security (CCS). ACM, 2016, pp. 641–652.

[236] F. Cangialosi, T. Chung, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "Measurement and analysis of private key sharing in the https ecosystem," in Proceedings of the 2016 ACM Conference on Computer and Communications Security (CCS). ACM, 2016, pp. 628–640.

[237] H. Liu, K. Levchenko, M. Félegyházi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage, "On the effects of registrar-level intervention," in Proceedings of the 4th USENIX Conference on Large-scale Exploits and Emergent Threats, ser. LEET'11. USENIX Association, 2011, pp. 1–8.

[238] M. Vasek and T. Moore, "Do malware reports expedite cleanup? An experimental study," in USENIX Workshop on Cyber Security Experimentation and Test (CSET). USENIX Association, 2012.

[239] B. Stock, G. Pellegrino, C. Rossow, M. Johns, and M. Backes, "Hey, you have a problem: On the feasibility of large-scale web vulnerability notification," in USENIX Security Symposium. USENIX Association, 2016, pp. 1015–1032.

[240] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from hell? Reducing the impact of amplification DDoS attacks," in USENIX Security Symposium. USENIX Association, 2014, pp. 111–125.

[241] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, and V. Paxson, "The matter of heartbleed," in Proceedings of the 2014 Conference on Internet Measurement Conference (IMC). ACM, 2014, pp. 475–488.

[242] D. McCoy, H. Dharmdasani, C. Kreibich, G. M. Voelker, and S. Savage, "Priceless: The role of payments in abuse-advertised goods," in Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS). ACM, 2012, pp. 845–856.

[243] R. Clayton, T. Moore, and N. Christin, "Concentrating correctly on cybercrime concentration," in 14th Workshop on the Economics of Information Security, 2015. [Online]. Available: http://www.econinfosec.org/archive/weis2015/papers/WEIS_2015_clayton.pdf

[244] M. Vasek, M. Weeden, and T. Moore, "Measuring the impact of sharing abuse data with web hosting providers," in ACM Workshop on Information Sharing and Collaborative Security. ACM, 2016, pp. 71–80. [Online]. Available: http://tylermoore.ens.utulsa.edu/wiscs16.pdf

[245] S. Tajalizadehkhoob, C. Gañán, A. Noroozian, and M. van Eeten, "The Role of Hosting Providers in Fighting Command and Control Infrastructure of Financial Malware," in Proceedings of the 12th ACM Symposium on Computer and Communications Security (ASIACCS). ACM, 2017.

[246] A. Sarabi, Z. Zhu, C. Xiao, M. Liu, and T. Dumitraş, "Patch me if you can: A study on the effects of individual user behavior on the end-host vulnerability state," in International Conference on Passive and Active Network Measurement. Springer, 2017, pp. 113–125.

[247] K. G. Provan and P. Kenis, "Modes of network governance: Structure, management, and effectiveness," Journal of public administration research and theory, vol. 18, no. 2, pp. 229–252, 2008.

[248] J. De Bruijn, Management in networks: on multi-actor decision making. Routledge, 2008.

[249] A. Schmidt, "Secrecy versus openness: Internet security and the limits of open source and peer production," 2014.

Summary

Internet infrastructure, in addition to facilitating communication and data sharing for users around the world, also serves as a platform for fraud and misuse. Cybercriminals exploit the global web infrastructure for personal and financial gain. They devise ways to compromise servers and web domains via technical vulnerabilities in systems or human mistakes. The consequences not only harm individuals, but also generate wider economic impacts, hurting society as a whole.

Hosting providers are a key Internet intermediary. They provide and facilitate the infrastructure used for storing and hosting online content. In theory, hosting providers can play an important role in fighting cybercrime and misuse. This is because many online threats, be they high-profile or mundane, use hosting infrastructure at the core of their criminal operations. Selling stolen credit cards, publishing materials showing child sexual abuse, running C&C servers for botnets, and phishing for personal information are all crimes that use online storage maintained and offered by hosting providers. Sometimes existing legitimate websites are compromised for illicit purposes, or new websites may be registered solely for criminal gain.

In practice, thousands of providers enable online crime on a daily basis, wittingly or unwittingly. Yet, we see large differences in the security measures taken by hosting providers. Some providers implement an array of actions to protect their customers. Others lack even the capacity to detect cybercrime, are negligent of cybercrime, or even willfully facilitate it.

Ensuring and improving security in the hosting market constitutes a collective action problem. This is because the consequences of online crime affect not only providers, but also users, the economy, and society as a whole. Moreover, while multiple actors would benefit from a solution to this problem, it is implausible that any individual actor could solve the problem alone, due to all the associated factors and costs.

So far, the hosting provision market itself has not taken steps to ensure adequate online security. Providers lack incentive to do so, due to negative

externalities in the market and information asymmetry regarding the security levels of the services on offer. Hierarchal or network governance mechanisms or a combination of these could be effective, if they were designed taking into account the properties of hosting providers and the market they operate in. However, much about this market is still unclear. For instance, there exists no comprehensive empirical understanding of how many providers operate worldwide, how their services are distributed, and what hosting types predominate.

Therefore, tackling hosting providers' security problem and improving their security performance first requires a better understanding of the hosting market itself. That is, we need to know more about the structure and operations of hosting providers, the current security levels of their infrastructure, and the mechanisms and factors that shape their security decisions and security outcomes. This leads to the main research question of the current thesis:

How can the security performance of hosting providers be measured and improved?

This main research question is divided into multiple areas of inquiry. These are explored in five chapters presenting empirical peer-reviewed articles which form the core of this dissertation. All of the studies have been well received by both academia and the industry, and their findings have became starting points for policy discussions and next research steps. The empirical research starts with a mapping of technical identifiers to economic agents. This culminates in an innovative approach for making inferences about providers' security efforts by collecting and analyzing complex real world datasets that reveal hosting providers' security practices and criminal activities in their networks.

Chapter 3 develops an approach for uncovering and grasping the complexity of the hosting market. The method proposed for identifying hosting providers is to map technical identifiers – IP addresses and domain names captured in passive DNS data – to the agents behind the hosting services, identified by organizational data in the WHOIS database. This enables us to distinguish the organizations that are responsible for the security of hosting services and in a position to take action to improve it. We also survey the hosting landscape, empirically identifying a diverse set of business profiles. The landscape revealed is a very heterogeneous one. Some providers owned only a single IP address, which they used to offer shared hosting services. In terms of global distribution, hosting providers were located in more than 150 countries. Several hosting providers had infrastructure in multiple countries.

Chapter 4 develops an analytical and statistical method for inferring infor-

mation about hosting providers' security performance from noisy abuse data. Using this analytical model, we decompose the different sources of variance present in abuse data, such as defender properties (i.e., their structure and security efforts), attacker behavior, and measurement and attribution errors. Among these factors, we quantify the impact of hosting provider structural properties on concentrations of abuse for the whole population of 45,000 hosting providers (defenders). Our results show that a handful of providers' structural characteristics related to their attack surface – such as number of domain names, number of IP addresses used for web hosting, and the size of their shared hosting business – accounted for 84% of the variation in phishing abuse concentrations. We further show that operational factors that are more difficult to measure for the population of hosting providers – such as providers' pricing strategy, website popularity, years in business, and use of applications known to be vulnerable – can explain a further 77% of the variance in phishing abuse concentrations. This, however, leaves little room for providers' security efforts to influence abuse levels.

Chapter 5 takes these results a step further, to examine the impact of both structural properties and the reactive security efforts of hosting providers on abuse concentrations in their networks. The chapter investigates what properties of providers drive attackers' preference for hosting the command-and-control (C&C) domains used to communicate instructions to machines infected with Zeus malware. Results show that structural properties of providers played a less significant role (71%) in explaining C&C abuse concentrations compared to phishing abuse. Further, concentration of C&C domains was negatively related to a rule of law indicator for the countries where the domains were hosted. Finally, providers' speed in taking down C&C domains (a proxy for reactive security efforts) was only weakly related to C&C concentrations, explaining just an additional 1% of the variance. Thus, attackers appear to show little preference for providers that allow long-lived C&C domains. More generally, our results suggest that providers' structural properties, such as size and pricing strategy, play a much more economically and statistically significant role in driving C&C concentrations, compared to any reactive security measures that providers take, such as the effort they put into taking down abused websites.

Chapter 6 focuses on attackers' behavior and strategies as another factor that can drive abuse concentrations. Here, we confine our investigation to one case study, that of Zeus malware, which is a leading family of malware used for attacking financial institutions. We look only at the targets of Zeus attack, in an effort to better understand attackers' behavior, regardless of the providers that hosted the victimized domains. We transform four years of noisy data from

Zeus configuration files into structured data on attack targets and attackers' instructions sent to the machines infected with Zeus malware. Results show that targets were located all over the world and that 90% of the attacks were aimed at only 15% of the overall targets. Surprisingly, the size of the targeted financial institution did not drive or predict attack concentration. Attack persistence varied widely in our sample. Some institutions were attacked very briefly, while others underwent attacks during the entire observation period (216 weeks). We speculate that the brief attacks were part of a trial-and-error process of attackers. Studying the attack code over the course of four years, we observe very high rates of code reuse and code similarity. We expected this prevalence of code sharing to lead to low code development costs, low market entry barriers for attackers and newbies, and ultimately to a rise in the numbers of attacks. However, we found a ceiling in new targets being attacked. Taken together, our results suggest that Zeus attack volumes are driven not by the technology but more by the cash-out segment of the value chain, such as money mules.

Chapter 7 investigates providers' proactive security efforts, seeking to quantify their impact on abuse concentrations for the specific case of shared hosting. The core idea is that although hosting providers are a key actor in fighting website compromises, their ability to prevent abuse is constrained by the security practices of their own customers. In shared hosting, customers operate under restricted privileges. Providers thus retain more control over system configurations. We examine 15 proxy features, from which we distill four major latent factors capturing security efforts: content security, webmaster security, web infrastructure security, and web application security. Our results confirm that providers and webmasters employ various soft techniques, such as hardening the software discovery process and hiding version information, to make it harder for criminals to exploit vulnerabilities in their applications. We observe that content and web application security played a significant positive role in reducing abuse concentrations, after controlling for size. Our findings suggest that providers' efforts at the software patching level – even for client-side software like content management systems (CMS) – can eventually lead to a substantial reduction in abuse concentrations.

Regarding the implications of our results for practice, I conclude that the various insights produced, concerning hosting providers' characteristics, security incentives, and security performance, constitute an essential first step toward improving online security. To effectively influence and elevate providers' security performance, providers' incentives and properties have to be taken into account. Influence can be exerted not only by the market players themselves, but also by government through hierarchical mechanisms and by hosting community peers

through network governance mechanisms. Hosting providers themselves must be cognizant of the structural properties of their services that expose them more to cyberabuse, while investing in reactive and proactive security efforts. Government can employ soft techniques, such as facilitating information transparency regarding providers' security levels and rewarding actions to improve security performance. Communities can influence security practices by developing norms and empirically-based best practices that capture the complexities and difficulties that providers face in achieving and maintaining high security levels.

Samenvatting

Het internet wordt gebruikt voor communicatie en het delen van informatie tussen gebruikers over de hele wereld, maar ook als een platform voor fraude en misbruik. Internetcriminelen misbruiken de infrastructuur van het wereldwijde web voor financieel gewin. Zij misbruiken kwetsbaarheden in de beveiliging van web servers en domeinen om deze vervolgens in te zetten voor criminele doeleinden. De consequenties daarvan benadelen niet alleen eigenaren van die servers en domeinen, maar hebben bredere economische gevolgen voor de maatschappij als geheel.

Hosting providers hebben een sleutelpositie in de bestrijding van deze vormen van cybercrime. Deze bedrijven leveren de infrastructuur die gebruikt wordt voor het online brengen van web domeinen en diensten. Allerlei vormen van cybercrime hebben hostingdiensten voor een deel van de criminele handelingen. Het verkopen van gestolen credit cards, het publiceren van beelden van kindermisbruik, het laten draaien van command-and-control (C&C) servers voor botnets, phishing sites die persoonlijke gegevens proberen buit te maken – het zijn allemaal misdaden die diensten gebruiken die worden beheerd en aangeboden door hosting providers. Vaak worden bestaande legale websites gehackt en gebruikt voor illegale doeleinden, soms worden nieuwe websites geregistreerd door de criminelen zelf.

In de praktijk maken duizenden providers online criminaliteit dagelijks mogelijk, bewust of onbewust. Maar er zijn grote verschillen in de veiligheidsmaatregelen die door hosting providers worden genomen. Sommige providers nemen een scala aan maatregelen om hun klanten te beschermen. Anderen zijn niet in staat of bereid om cybercrime op hun systemen te ontdekken. Een kleine groep faciliteert doelbewust criminele praktijken.

Het verbeteren van veiligheid op de hostingmarkt is een probleem dat om gemeenschappelijke actie vraagt. En wel omdat online criminaliteit niet alleen gevolgen heeft voor providers, maar ook voor haar klanten, en voor de economie en de maatschappij als geheel. Terwijl een veelheid aan betrokkenen baat heeft

bij een oplossing voor dit probleem, kan het niet opgelost worden door een individuele speler, vanwege alle factoren en kosten die er mee verbonden zijn.

Tot nu toe heeft de hostingprovidermarkt zelf weinig stappen ondernomen om voldoende online veiligheid te verhogen. Het ontbreekt providers aan een economische prikkel om dit te doen, vanwege het optreden van negatieve externaliteiten en vanwege informatie asymmetrie: klanten en toezichthouders kunnen niet goed zien welke providers het goed doen en welke niet. Overheidsregulering of zelf-regulering, of een combinatie van deze twee zouden effectief kunnen zijn, als ze zouden worden ontworpen met inachtneming van de karakteristieken van hosting providers en de markt waarin zij opereren. Veel over deze markt is echter nog onduidelijk. Er bestaat bijvoorbeeld geen compleet en goed gedocumenteerd overzicht van hoeveel providers er wereldwijd opereren, hoe hun diensten zijn verdeeld, en welke soorten van hostingdiensten precies worden aangeboden.

Daarom is er eerst een beter begrip van de hostingmarkt nodig om het veiligheidsprobleem van hosting providers aan te kunnen pakken, en om hun prestaties op het gebied van veiligheid te kunnen verbeteren. Dat betekent dat we meer moeten weten over de structuur en de activiteiten van hosting providers, de huidige veiligheidsniveau's van hun infrastructuur, en de mechanismes en factoren die hun beslissingen en resultaten op het gebied van veiligheid bepalen. Dit leidt tot de belangrijkste onderzoeksvraag van dit proefschrift:

Hoe kunnen de veiligheidsprestaties van hosting providers worden gemeten en verbeterd?

Deze hoofdonderzoeksvraag is onderverdeeld in verschillende deelvragen. Na de inleiding in hoofdstuk 1 en het literatuuroverzicht en theoretische model in hoofdstuk 2, worden de deelvragen beantwoord in vijf empirische hoofdstukken die gebaseerd zijn op peer-reviewed artikelen. Deze studies zijn goed ontvangen in zowel de academische wereld als in het bedrijfsleven, en de bevindingen zijn startpunt geworden voor beleidsdiscussies en vragen voor vervolgonderzoek.

Het empirisch onderzoek begint met het in kaart brengen van de technische kenmerken van marktpartijen en culmineert in een vernieuwende benadering waarmee conclusies getrokken kunnen worden over beveiligingsinspanningen van providers door het verzamelen en analyseren van grootschalige datasets uit de praktijk waaruit blijkt hoe hosting providers te werk gaan op het gebied van veiligheid en waarmee criminele activiteiten in hun netwerken aan het licht komen.

Hoofdstuk 3 ontwikkelt een benadering om duidelijkheid te krijgen over

de complexiteit van de hostingmarkt. De voorgestelde methode om hosting providers te identificeren bestaat uit het in kaart brengen van de hostingmarkt door het koppelen van passieve DNS data en WHOIS data van domeinnamen en IP adressen. Dit stelt ons in staat om de aanbieders te identificeren die verantwoordelijk zijn voor de veiligheid van hosting diensten. We verkennen ook het hosting landschap als geheel en treffen daar een empirisch divers geheel aan provider typen aan. Het landschap dat hieruit opdoemt is zeer heterogeen. Ter illustratie: sommige providers blijken eigenaar van slechts een enkel IP adres wat ze gebruikten om zogenaamd "shared hosting" diensten aan te bieden. Andere providers hebben miljoenen adressen in beheer. Kijkend naar wereldwijde verspreiding, blijkt dat hosting providers gevestigd zijn in meer dan 150 landen. Verschillende hosting providers hebben infrastructuur in een veelheid aan landen.

Hoofdstuk 4 ontwikkelt een analytische en statistische methode om informatie over de veiligheidsprestaties van hosting providers af te leiden uit grootschalige incidentdata met veel ruis, zoals de links naar phishing pagina's die worden ontdekt in spam. Met een analytische model ontrafelen we de verschillende bronnen van variantie die zich voordoen in incidentdata, zoals de eigenschappen van verdedigers (d.w.z. hun structuur en beveiligingsinspanningen), het gedrag van aanvallers, maar ook meetfouten en attributiefouten. We kwantificeren de invloed die de structurele eigenschappen van hosting providers hebben op hoeveel phishing domeinen zich bevinden in de netwerken van de totale populatie van 45.000 hosting providers. Met een handvol structurele eigenschappen van de providers kunnen we meten in welke mate de provider blootgesteld is aan deze aanvallen, hun zogenaamde attack surface. Dit betreft indicatoren zoals aantal domeinnamen, aantal IP adressen die gebruikt worden voor web hosting, en de omvang van hun shared hosting business. Gezamenlijk zijn deze structurele eigenschappen verantwoordelijk zijn voor 84% van de variatie in concentraties van phishing domeinen. Van de resterende 16% variatie kunnen we 77% verklaren met operationele factoren die moeilijker meetbaar zijn voor de hele populatie van hosting providers – zoals het prijsniveau van providers, de populariteit van websites, het aantal jaren dat hun onderneming bestaat, en het gebruik van applicaties die bekend zijn vanwege hun kwetsbaarheden. Dit betekent, verrassend genoeg, dat er weinig impact overblijft voor de veiligheidsinspanningen van providers om het niveau van misbruik te beïnvloeden.

Hoofdstuk 5 zet een volgende stap door opnieuw de invloed te meten van de structurele eigenschappen van providers, maar nu in een andere type misbruik: de locatie van de command-and-control (C&C) domeinen die gebruikt worden om instructies te geven aan apparaten die geïnfecteerd zijn met Zeus

malware. Zulke C&C domeinen zijn belangrijk voor criminelen, anders dan bij phishing domeinen, die op grote schaal geproduceerd en vervangen worden. Daarom wordt algemeen aangenomen de criminelen voorkeuren zullen hebben voor providers die C&C domeinen langer online laten staan, oftewel die minder alert hun netwerken beveiligen. Resultaten laten zien dat structurele eigenschappen van providers inderdaad een minder belangrijke rol speelden (71%) om C&C concentraties te verklaren, in vergelijking met phishing domeinen, maar dat nog steeds een heel hoog percentage verklaard wordt uit puur structurele eigenschappen, niet uit het veiligheidsbeleid van providers. Verder bleek de concentratie van C&C domeinen negatief te correleren met een indicator voor sterke wetgeving in de landen waar de domeinen geregistreerd stonden. Tenslotte bleek dat de snelheid waarmee providers C&C domeinen verwijderden (een proxy voor hun reactieve veiligheidsinspanningen) slechts zwak correleerde met concentraties van C&C. Dit verklaarde niet meer dan nog 1% van de variatie. Dus aanvallers lijken weinig voorkeur te hebben voor providers die langdurig C&C domeinen laten voortbestaan. Meer in het algemeen wijzen onze resultaten in de richting dat structurele eigenschappen van providers, zoals omvang en prijsbeleid, een veel grotere economisch en statistisch significante rol spelen in het sturen van C&C concentraties, dan welke reactieve veiligheidsmaatregelen ook die providers nemen, zoals de energie die ze stoppen in het verwijderen van misbruikte websites.

Hoofdstuk 6 richt de aandacht meer direct op de strategieën van aanvallers als factor die concentraties van misbruik kan sturen. We bestuderen data over de doelwitten van criminelen die Zeus malware hebben ingezet, een belangrijke malwarefamilie die gebruikt wordt voor het aanvallen van financiële instituties. We kijken puur naar de doelen van de aanvallen, zoals die zichtbaar worden in de instructies die meegegeven worden aan de Zeus malware, in een poging het gedrag van de aanvaller beter te begrijpen, onafhankelijk van de providers waar de aangevallen domeinen gehost werden. We transformeren vier jaar aan onderschepte Zeus configuratiebestanden naar gestructureerde gegevens over aanvalsdoelen en instructies die de aanvaller gestuurd heeft naar de apparaten die waren geïnfecteerd met Zeus malware. De resultaten laten zien dat de doelen gesitueerd waren over de hele wereld, en dat 90% van de aanvallen gericht was op slechts 15% van het totale aantal doelen. Verrassend is dat de omvang van het aangevallen financiële instituut geen voorspeller is van de concentratie van aanvallen. In ons voorbeeld was er een grote variatie in de hardnekkigheid van aanvallen. Sommige organisaties werden slechts zeer kort aangevallen, terwijl andere aanvallen te verduren kregen gedurende de hele observatieperiode (216 weken). Onze veronderstelling is dat de kortdurende aanvallen onderdeel waren

van een leerproces door de aanvallers. Tijdens de periode van vier jaar dat we de aanvalscode hebben bestudeerd, hebben we uitzonderlijk veelvuldige herhaling en hergebruik van vergelijkbare codes gezien. We hadden verwacht dat het veel voorkomen van het delen van codes zou leiden tot lage kosten voor het ontwikkelen van codes, lage drempels om de markt te betreden voor aanvallers en nieuwkomers, en uiteindelijk zou leiden tot een toename in het aantal aanvallen. We vonden echter een plafond in het aantal nieuwe doelen dat werd aangevallen. Alles bij elkaar genomen wijzen onze resultaten in een richting dat Zeus aanvallen niet zozeer worden gestuurd door de technologie maar eerder door het segment van de waardeketen waar de opbrengst wordt weggesluisd, zoals cashout via geldezels.

Hoofdstuk 7 presenteert een grootschalige directe meting van de proactieve veiligheidsinspanningen van providers en poogt de invloed hiervan op de concentraties van misbruik te kwantificeren. We richten ons specifiek op shared hosting. Hosting providers spelen een sleutelrol bij het bestrijden van misbruik, maar hun mogelijkheden om misbruik tegen te gaan worden beperkt door de veiligheidspraktijken van hun eigen klanten. We willen weten hoeveel invloed providers hebben op die praktijken van klanten. Shared hosting is daarvoor een geschikte casus, omdat in die dienst klanten opereren met beperkte privileges. Daardoor behouden providers meer controle over de configuraties van de systemen. We verzamelen data over 15 veiligheidsgerelateerde voor ongeveer een half miljoen webdomeinen die verspreid zijn over de gehele shared hosting markt. Daaruit destilleren we vier belangrijke latente factoren die veiligheidsinspanningen omvatten: beveiliging van de content, beveiliging van de webmaster, veiligheid van de infrastructuur van het web, en veiligheid van de web toepassingen. Providers hebben vooral invloed op de laatste twee factoren. We constateren dat de beveiliging van inhoud en web applicaties een belangrijke positieve rol heeft gespeeld in het terugdringen van misbruik. Onze bevindingen suggereren dat de inspanningen van providers op het niveau van software patching – zelfs voor software zoals content management systems (CMS) die door klanten zelf beheerd worden – in de toekomst kunnen zorgen voor een aanzienlijke afname van concentraties in misbruik.

Voor wat betreft de gevolgen van onze resultaten voor de praktijk, concluderen we dat we nu meer inzicht hebben in de hostingmarkt en in de factoren die de veiligheidsprestaties van providers bepalen. Dit is een essentiële eerste stap betekenen op weg naar het verbeteren van online veiligheid. Om de beveiligingsprestaties van providers effectief te beïnvloeden en op een hoger niveau te brengen, moet rekening gehouden worden met hun eigenschappen en incentives. Invloed kan niet alleen worden uitgeoefend door de marktspelers

zelf, maar ook door de overheid met hiërarchische mechanismen en met het faciliteren van netwerk governance zoals benchmarking en initiatieven uit de sector zelf. Hosting providers zelf moeten op de hoogte zijn van welke eigenschappen van hun diensten hen meer blootstellen aan misbruik en investeren in reactieve en proactieve tegenmaatregelen. De overheid kan soft regulation toepassen, zoals het faciliteren van openbare informatie over de beveiligingsniveaus van providers (benchmarks). Dit verlaagt de informatie asymmetrie en zorgt ervoor dat de markt betere beveiligingsprestaties kan belonen. De hosting sector zelf kan beveiligingspraktijken beïnvloeden door gedeelde normen te ontwikkelen en door 'best practices' te ontwikkelen die op objectieve metingen van veiligheidsprestaties zijn gebaseerd.

Authorship Contribution

The five empirical chapters of this thesis are based on peer-reviewed publications coming out of collaborative work with a variety of co-authors. I'm the lead author on all five studies. I was fortunate to receive valuable contributions from several colleagues. Below, I summarize the main contributions to each paper.

In the IEEE NOMS study, I conducted the data generation, data modeling and the overall writing. Maciej Korczyński and Carlos Gañán provided a lot of help regarding the approach, visualizing the plots and drafting the text. Arman Noroozian helped with improving the text of the literature review section. Most of the co-authors contributed ideas on the approach and analysis, as well as improving the text of the manuscript.

For the study published in ACM TOIT, I handled empirical data generation, modeling, analysis and the bulk of the writing. The overall methodology is indebted to ideas generated by Rainer Böhme, partially during a research visit at his team in Innsbruck. Maciej Korczyński collected the feature related to the use of WordPress. The analysis of measurement errors was carried out and written up by Carlos Gañán. All co-authors helped a lot with clarifying the argument, also in light of reviewer comments from an earlier submission, and with improving the text.

In the study published in ACM ASIACCS, I conducted the empirical data generation, modeling, analysis and the bulk of the writing. Carlos Gañán drafted the literature review section. Arman Noroozian and Michel van Eeten helped with improving the quality of the text and clarifying the arguments.

The WEIS paper on target selection is an extension of my master thesis. The thesis benefited from hands-on supervision of Hadi Asghari, the daily supervisor of the project. For the WEIS paper, Hadi conducted the analysis of target size and revised a significant part of the final text. Carlos Gañán conducted and wrote the analysis of code re-use. All co-authors contributed greatly to the ideas on the approach and quality of the final text.

For the ACM CCS paper, I conducted all of the analysis, drafted the main text and conducted a minor part of the data collection. Most of the data

collection of the security and vulnerability features was primarily done by Tom Van Goethem. Wouter Joosen supervised the large-scale measurements and helped in improving the infrastructure required for the measurements. Maciej Korczyński collected additional features related to the use of WordPress, drafted the literature review section and helped greatly in finalizing the text of the overall paper. Arman Noroozian helped to extend the scripts for collecting data on one of the CMSes and checked the false positive rate of our CMS measurements on a sample of 50 websites. Tyler Moore and Rainer Böhme both contributed important ideas on fleshing out the methodology needed to answer the central question, as well as extensively helping to clarify the overall argument and drafting and improving the text.

Finally, in all of the studies, Michel van Eeten contributed incredibly in conceptualizing the main ideas and in improving the approach, arguments, and writeup.

List of Publications

- S. Tajalizadehkhoob, H. Asghari, C. Gañán, and M. van Eeten, "Why them? extracting intelligence about target selection from zeus financial malware," in Proceedings of the 13th Annual Workshop on the Economics of Information Security (WEIS). WEIS, 2014. [Online]. Available: http://www. econinfosec.org/archive/weis2014/papers/Tajalizadehkhoob-WEIS2014.pdf

- A. Noroozian, M. Korczyński, S. TajalizadehKhoob, and M. Van Eeten, "Developing security reputation metrics for hosting providers," in Proceedings of the 8th Workshop on Cyber Security Experimentation and Test (CSET). Berkeley, USA: USENIX Association, 2015, pp. 5–5. [Online]. Available: http://dl.acm.org/citation.cfm?id=2831120.2831125

- S. Tajalizadehkhoob, M. Korczyński, A. Noroozian, C. Gañán, and M. van Eeten, "Apples, oranges and hosting providers: Heterogeneity and security in the hosting market," in Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS), Istanbul, Turkey: IEEE, 2016, pp. 289–297. [Online]. Available: http://ieeexplore.ieee.org/document/7502824/

- S. Tajalizadehkhoob, R. Böhme, C. Gañán, M. Korczyński, and M. van Eeten, "Rotten Apples or Bad Harvest? What We Are Measuring When We Are Measuring Abuse," in Forthcoming ACM Transactions on Internet Technology (TOIT), 2017. [Online]. Available: https://arxiv.org/abs/1702.01624

- M. Korczyński, S. Tajalizadehkhoob, A. Noroozian, M. Wullink, C. Hesselman, and M. v. Eeten, "Reputation metrics design to improve intermediary incentives for security of tlds," in Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), 2017, pp. 579–594. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7962004/

- S. Tajalizadehkhoob, C. Gañán, A. Noroozian, and M. v. Eeten, "The Role of Hosting Providers in Fighting Command and Control Infrastructure of Financial Malware," in Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIA CCS). Abu Dhabi, UAE: ACM, 2017, pp. 575–586. [Online]. Available: http://doi.acm.org/10.1145/3052973.3053023

- M. Aertsen, M. Korczyński, G. Moura, S. Tajalizadehkhoob, and J. van den Berg. "No domain left behind: is Let's Encrypt democratizing encryption?." In the proceedings of the Applied Networking Research Workshop (ANRW), ACM, 2017, pp. 48-54. [Online]. Available: https://arxiv.org/pdf/1612.03005.pdf

- A. Noroozian, M. Ciere, M. Korczyński, S. Tajalizadehkhoob, and M. van Eeten, "Inferring the security performance of providers from noisy and heterogenous abuse datasets," in Proceedings of the 16th Annual Workshop on the Economics of Information Security (WEIS), 2017. [Online]. Available: http://weis2017.econinfosec.org/wp-content/uploads/sites/3/2017/ 05/WEIS 2017 paper 60.pdf

- S. Tajalizadehkhoob, T. van Goethem, M. Korczyński, A. Noroozian, R. Böhme, T. Moore, W. Joosen, and M. van Eeten, "Herding Vulnerable Cats: A Statistical Approach to Disentangle Joint Responsibility for Web Security in Shared Hosting," in Proceedings of the ACM Conference on Computer and Communications Security (CCS). ACM, 2017. [Online]. Available: https://arxiv.org/pdf/1708.06693.pdf

About the Author

Samaneh Tajalizadehkhoob was born in Tehran the capital of Iran, in 1987. She obtained a dual degree in Electrical Engineering from University of Tehran (UT), Iran and Purdue University Indianapolis (IUPUI), United States, in 2010. To fulfill the requirements of her dual degree, she spent the first two years of her bachelor's study in UT and the last two years in IUPUI.

In 2011, Samaneh moved to the Netherlands to pursue her master's degree in Engineering and Policy Analysis in the Technology, Policy and Management (TBM) faculty, at Delft University of Technology. She finalized her master thesis about online banking fraud under the supervision of Prof. dr. Michel van Eeten in the Economics of Cybersecurity group.

Shortly after obtaining her master's degree in August 2013, in September 2013 Samaneh started as a PhD candidate in the same group. She worked for the NWO REMEDI3S project funded by NCSC (National Cyber Security Center) and SIDN (.NL registry) on developing reputation metrics for Internet intermediaries. As a part of her PhD project, she visited Prof. Rainer Böhme's security and privacy group at Innsbruck university, Austria, where she learned a lot about statistical modeling. She also visited prof. Wouter Joosen's imec-DistriNet group at KU-Leuven, Belgium, where she collaborated with Tom Van Goethem on vulnerability measurements for shared hosting providers. Both of these collaborations resulted in papers that are included as a part of this dissertation.

During her PhD project she co-authored a number of peer-reviewed papers (see the publication list for the complete list). She supervised several master students working on topics related to her PhD and master thesis project. Samaneh

was also involved with teaching a master's course related to decision making in complex networks in the TBM faculty. Finally, she was a board member of the TBM faculty PhD council for two years during her PhD.

Currently, Samaneh is a postdoctoral researcher at the same group working on the economics of financial malware. As a team consist of another postdoctoral researcher and a PhD student, their goal is to explore, understand, and predict the behavior of malware used to attack financial institutions, using machine learning and data analytics tools. Additionally, she is involved in supervising several master students in collaboration with Fox-IT. Last, she is coordinating and teaching a bachelor course about techniques and methods used in 'Data Mining' and machine learning at TU-Delft.