EARLY DETECTION OF SPAM-RELATED ACTIVITY

A Thesis
Presented to
The Academic Faculty

by

Shuang Hao

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology
December 2014

EARLY DETECTION OF SPAM-RELATED ACTIVITY

Approved by:

Professor Nick Feamster, Advisor
School of Computer Science
Georgia Institute of Technology

Professor Mostafa Ammar
School of Computer Science
Georgia Institute of Technology

Professor Vern Paxson
Department of Electrical Engineering
and Computer Sciences
University of California, Berkeley & International Computer Science Institute

Dr. Christian Kreibich
Senior Research Scientist
International Computer Science Institute & Lastline, Inc.

Professor Wenke Lee
School of Computer Science
Georgia Institute of Technology

Date Approved: 22 October 2014

To my dear family,

Thank you for all of your love, support and encouragements.

ACKNOWLEDGEMENTS

This thesis would not have been completed without the invaluable support, guidance, and inspiration I have received from many people during my Ph.D. study.

My sincere gratitude goes to my advisor, Prof. Nick Feamster, for his caring guidance and consistent support. He led me on this exciting and fruitful journey, encouraged me to explore research topics, and provided me with both insightful advice and intellectual help. His boundless enthusiasm always inspires my passion for our work. He taught me indispensable research skills including picking problems, crafting solutions, presenting to audiences, and cultivating research taste. I feel extremely fortunate to have had the opportunity to work with him closely for many years. The example he set up as a wonderful researcher and teacher will continue to light my way in my future career.

I am deeply grateful to Prof. Vern Paxson for mentoring me during my summer internships at ICSI. His valuable knowledge and constructive suggestions helped me to better address the research problems, and guided me to learn to think more critically and creatively as a scientist. The detailed suggestions and comments that he gave on earlier versions of this dissertation greatly helped me to improve my work.

I would like to thank the members of my committee, Prof. Wenke Lee, Prof. Mostafa Ammar, and Dr. Christian Kreibich, for their interest and help in my work. Prof. Wenke Lee has offered insightful suggestions through my study at Georgia Tech. Prof. Mostafa Ammar gave thought-provoking feedback on my thesis. Dr. Christian Kreibich helped me to understand the details of spam campaigns, and provided great advice during my internships at ICSI. I also wish to express my gratitude to all coauthors and collaborators for their support and assitance, which makes the research much stronger and solid.

I have also been very fortunate to be a member of an excellent networking research group at Georgia Tech. I wish to thank Anirudh Ramachandran, Mukarram bin Tariq, Murtaza Motiwala, Valas Valancius, Sam Burnett, Srikanth Sundaresan, Robert Lychev, Hyojoon

Kim, Yogesh Mundada, Bilal Anwer, Maria Konte, Ben Jones, Sean Donovan, Sarthak Grover, and Mi Seon Park for their valuable discussions and feedback on the research, and thank Cong Shi, Samantha Lo, Nan Hua, Tongqing Qiu, Partha Kanuparthy, and Ahmed Mansy for getting the lab life enjoyable and enriching. Their friendship and company has made my years at Georgia Tech full of wonderful memories.

I dedicate this dissertation to my family. My parents, my brother Yang, and my grandparents have always been supportive, loving, and encouraging. I am deeply indebted to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Spam, the distribution of unsolicited bulk email, is a big security threat on the Internet. Recent studies show approximately 70–90% of the worldwide email traffic—about 70 billion messages a day—is spam [76,105,115]. Spam consumes resources on the network and at mail servers, and it is also used to launch other attacks on users, such as distributing malware or phishing. Spammers have increased their virulence and resilience by sending spam from large collections of compromised machines ("botnets"). Spammers also make heavy use of URLs and domains to direct victims to point-of-sale Web sites, and miscreants register large number of domains to evade blacklisting efforts. To mitigate the threat of spam, users and network administrators need proactive techniques to distinguish spammers from legitimate senders and to take down online spam-advertised sites.

In this dissertation, we focus on characterizing spam-related activities and developing systems to detect them early. Our work builds on the observation that spammers need to acquire attack agility to be profitable, which presents differences in how spammers and legitimate users interact with Internet services and such characteristic behavior is detectable during early period of attack. We examine several important components across the spam life cycle, including spam dissemination that aims to reach users' inboxes, the hosting process during which DNS servers and Web servers maintain the operation of spam-advertised sites, and the naming process to acquire domain names via registration services.

We first develop a new spam-detection system based on network-level features of spamming bots. These lightweight features allow the system to scale better and to be more robust. Next, we analyze DNS resource records and lookups from top-level domain servers during the initial stage after domain registrations, which provides a global view across the Internet to characterize spam hosting infrastructure. We further examine the domain registration process and illuminate the unique registration behavior of spammers. Finally, we

build an early-warning system to identify spammer domains at time-of-registration rather than later at time-of-use.

We have demonstrated that our detection systems are effective by using real-world datasets. Our work has also had practical impact. Some of the network-level features that we identified have since been incorporated into spam filtering products at Yahoo! and McAfee, and our work on detecting spammer domains at time-of-registration has directly influenced new projects at Verisign to investigate domain registrations.

CHAPTER I

INTRODUCTION

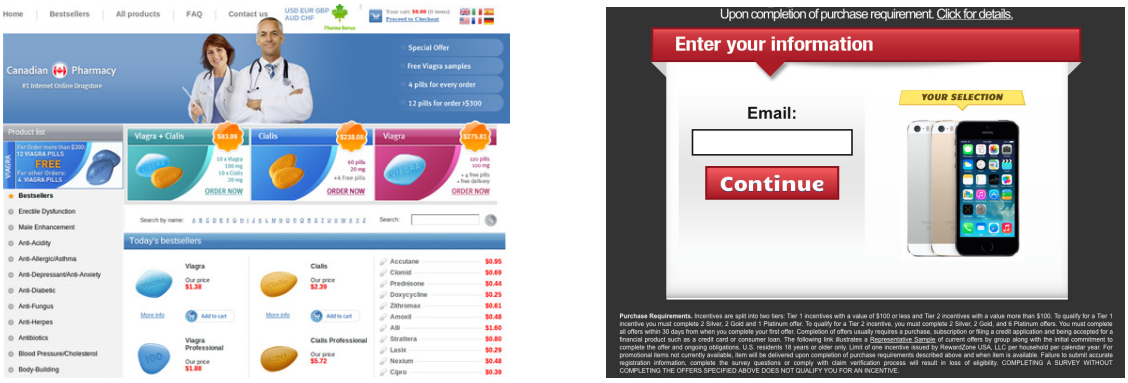
Over the past several decades, we have witnessed the tremendous growth of the Internet, which has nearly two billion users presently [55]. The thriving Internet applications bring great convenience for open communication and electronic business. On the other side, the growing reliance on the Internet also presents many security challenges. Especially, miscreants abuse network services to make illegal profits and consequently jeopardize users experience on the Internet.

A traditional and representative application is email service, which delivers electronic messages from a sender to one or more recipients. Since the protocol used to send and receive email, the Simple Mail Transfer Protocol (SMTP) [65], has no built-in measures for security or accountability, attackers exploit the insecurity and abuse the service to send spam—unsolicited bulk email. Spam not only consumes resources on the network and at mail servers, but also is used to conduct other attacks, such as distributing malware or phishing [115]. There is little cost for email senders, which allows miscreants to dispatch massive volumes of spam messages. If there were no automated spam mitigation methods, users' inboxes would quickly be filled with junk email and become useless. It is a critical task to detect spam in a fast and early manner.

In this chapter, we first introduce the spam problem and show it is a serious security threat. We then explain the research challenges for spam detection, and present the goals that we want to achieve in our work. Next, we provide an overview of our solution to achieve fast and early detection at different stages across the spam life cycle. Finally, we present the contributions of this dissertation and the bibliographic notes of our work.

1.1 Spam: A Widespread Security Threat

Spam refers to unsolicited bulk email, imposing a negative impact on users without the option to opt out in advance. The intent of spam content is to catch eyeballs and conduct



(a) Illicit online pharmacy

(b) Fake Web site with bogus survey

Figure 1: Examples of spam-advertised sites.

users to take some actions. Spam email usually contains links to entice the users to click on those links and visit particular Web sites, which we call *spam-advertised sites*. Figure 1 shows two examples of spam-advertised sites. The first one is a site of illicit online pharmacy, which allows people to buy drugs without an authorized prescription and thus endangers public safety. The second example is a fake Web site with a bogus survey. Users are asked to sign up and fill in personal details to receive gifts. The spammers get a fee for each submitted survey, and there is no gift for the users at the end of the process.

Spam has grown steadily since the early 1990s. Recent reports show that spam accounts for approximately 70–90% of email traffic sent on the Internet–about 70 billion messages a day [76, 105, 115]. Although several spamming botnets were taken down, such as Rustock, Kelihos, and Grum [79, 115], spam remains a serious security threat and attackers keep establishing new spam campaigns. In addition to email service, spammers have targeted many other online communication platforms, including instant messaging to involve real-time interaction [35], social networks (such as Twitter, Facebook, Tumblr, and Instagram) where spammers impose on friend or follower relationship [40, 91, 112], and blogs that experience comment spam [84].

Unlike legitimate online advertising accompanied with valuable services, spam consumes user attention in an intrusive manner without providing compensation or benefit. The negative impact for users includes the time wasted to process the spam email received in

inboxes and the hazard to miss important messages. If there were no automatic spam mitigation systems, users' inboxes would become useless quickly by being full with spam email. With existing anti-spam approaches, the time loss for users to deal with spam exposed in inboxes is still considerable [21]. Another costs imposed on society is the extra network resources and hardware required to handle spam. Spam exposure leads to significantly lower user engagement in the Internet services [21]. Companies, like Google and Yahoo!, invest huge effort to develop mitigation methods. To understand the economic loss caused by spam, recent studies have infiltrated and monitored spammers activity [13, 15, 62, 68]. A conservative estimate of social cost due to spam is 20 billion dollars per year, and every dollar in spammers' revenue costs approximately 100 dollars in society loss [96].

Depending on the miscreants' purpose and monetization means, spam can become a carrier of various attacks, which we summarize as follows.

- *Advertising illicit products.* Spam attempts to advertise products and generate a sale. The advertised products are usually illicit merchandises to gain high profits, including pharmaceuticals, replica luxury goods, and counterfeit software [72]. Once a user puts an order, the merchant acquires a payment and deliver the product. Spammers join affiliate programs and get a share of the profits from merchants [15, 62, 72].
- *Phishing.* Spam is used to steal identity and credential by spoofing messages sent from legitimate organizations, usually banks or other financial organizations, such as PayPal. The links in the spam lead to counterfeit Web sites which appear similar to the legitimate ones and ask users to input sensitive information, such as usernames and passwords [7, 22]. Phishing attacks usually cause direct financial loss to users with the stolen information.
- *Scam.* Criminals typically send out spam messages to defraud people by exploiting human psyche vulnerability, such as greed or gullibility. One class of threats includes fake gift cards and survey scams, where spammers pretend to offer free gifts and lure users to input personal information. The spammers get profits via directing users to specialized ad networks or selling the collected information [16]. Another type of scam

is to require small up-front fee to receive a large amount of money, commonly referred to as Nigerian scam or 419 scam [33, 86]. Scam could cause both financial loss and emotional damage to victim users.

- *Distributing malware.* Spam is widely used to trick users into installing malicious software on their computers [107]. Spam email can directly attach malicious executable files, use social engineering to motivate victims to install malwares, or contain links leading to Web sites with drive-by downloads. Symantec reported one in 291 emails contained a virus in 2012 [115].

Spam has become a serious security threats, in terms of magnitude, social loss, and consequent attacks. The prevalence of spam not only causes a heavy burden on the Internet ecosystem, but also poses a negative effect on people's daily lives.

1.2 Detecting Spam-Related Activity: Why Is It Hard?

To counter the threat of spam, a primary step is to effectively detect spam messages and spam-advertised sites, in order to prevent spam from reaching users' inboxes and take down spam campaigns. Developing effective detection methods is a challenging task due to the following reasons.

Absence of accountability and authentication. The Internet was originally designed for open communication, which has no strong accountability and authentication. This aspect facilitates misuse and makes it difficult to track down spammers. Criminals can spoof sender's email address, and recent study has revealed evidence for spam from hijacked BGP prefixes [93]. Although people have developed security-enhanced mechanisms, such as DomainKeys Identified Mail (DKIM) [19] to validate email senders and secure BGP [71] to authenticate ownership of IP address blocks, it needs enduring time and effort to complete extensive replacement [75]. Miscreants register numerous domains to host spam-advertised domains. Without real-name registration, it is difficult to ascertain the identities of domain owners and hold them accountable for the abuse. Although `WHOIS` data include information of operators or contacts, such information is inaccurate and sometimes even not available due to private registrations. The goal of our work is to design practical systems working on

existing network infrastructure to detect spam-related activities early.

Attack agility. To remain profitable, spammers are creative to use various techniques to increase the resilience of spam campaigns. For example, misspelled words can increase difficulty for text-based classifiers, and the spam content is even in the attached JPEG images or PDF files. Spammers deploy botnets (large sets of compromised machines) for spam sending to acquire IP dynamics. Spammers also rely on victims clicking on embedded URLs and being redirected to point-of-sale Web sites, which serves as a layer of proxy and allows to flexibly change the IP addresses associated with the sites [126]. Miscreants register large number of domains to evade blacklisting effort In this dissertation, we aim to characterize the fundamental invariants of spammer behavior, and develop accurate detection techniques over multiple stages.

Large scale of data. Nowadays network services and devices face massive traffic to process. For example, a campus or a large enterprise usually has hundreds of thousands of email addresses, and gets millions of incoming email messages each day [59]; ISP routers forward hundreds of gigabits of traffic per second; Top-level DNS servers receive billions of daily queries worldwide. The large data traffic on the Internet requires that we adopt lightweight features and fast algorithms when detecting spam-related activity.

1.3 Approach Overview

Spam is a multi-facet business to gain illicit profits. The spam business has many components and support services, which involves a range of different players and service providers. The most manifest perception to users is that they receive spam email in inboxes. However, there are many other activities undergoing before spam messages actually reach user inboxes. In Figure 2 we show three important stages in the spam life cycle. First, the domain name is a necessary component of a URL used to access Web sites. Miscreants purchase domains via registrars to establish spam-advertised sites. Next, a registered domain in use must have DNS nameservers to resolve IP addresses and deploy Web servers to host page content. Spammers need the hosting infrastructure to operate spam campaigns.

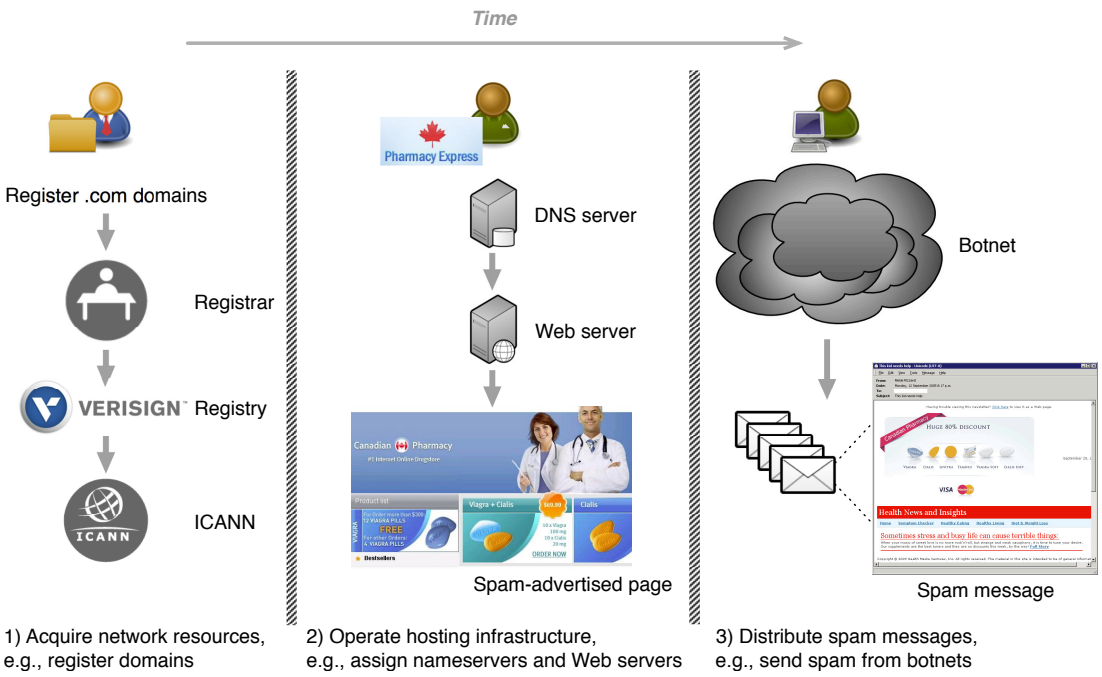


Figure 2: Stages of the spam life cycle.

Last, attackers send spam, usually from botnets, and entice users into clicking on the embedded URLs to visit spam-advertised sites. We provide a more detailed description about the spam ecosystem in Section 2.1. Most previous detection mechanisms are triggered by attack evidence and incur high overhead, such as parsing email content or crawling suspect URLs.

Thesis statement. In this dissertation, we posit that spammers need to acquire attack agility to be profitable, which exhibits characteristic behavior. We show the way in which spammers and legitimate users interact with Internet services presents differences and such characteristic behavior is detectable during early period of attack.

We build effective classifiers to detect spam-related activities across different life-cycle stages.

1.4 Contributions

In this dissertation, we make the following contributions:

1. **Characterization.** We understand and characterize the behavior of spammers to interact with Internet services. First, we perform a detailed analysis of the unusual

email sending patterns from spamming bots, and identify a set of network-level features. Next, we monitor the DNS behavior of spam-advertised domains, shortly after the domains are registered, to understand their hosting infrastructure and DNS lookup patterns. Further, we examine domain registration process and reveal the distinct behavior how spammers acquire domains.

2. **Early detection.** We develop two practical detection systems, *SNARE* and PREDATOR, based on lightweight features and fast algorithms. These systems are evaluated on real-world data and can accurately detect spam-related activities with a low false positive rate. Our work has started to make practical impact. Some network-level features from *SNARE* have since been successfully applied by Yahoo! and McAfee, and PREDATOR has directly influenced and created new projects at Verisign to investigate domain registrations.

We now describe our contributions in more detail.

Detecting spammers with network-level features. Chapter 3 presents a spam-filtering system, *SNARE*, based solely on lightweight network-level features, which could allow it to scale better and to operate on higher traffic rates. *SNARE* relies on the fact that about 75–95% of spam email originates from botnets, which often exhibit unusual sending patterns that differ from those of legitimate email senders. We define network-level features as observations and measurements that can be made cheaply and efficiently, usually at the network layer of the TCP/IP protocol stack, and from anywhere in the network (*i.e.*, not necessarily at the recipient's machine). Examples of such features include the Autonomous System (AS) of the sender, the geographic distance between sender and receiver, the density of email senders in the surrounding IP address space, and the time of day the message is sent. We demonstrate that *SNARE* can achieve comparable accuracy to existing static IP blacklists: about a 70% detection rate for less than a 0.3% false positive rate. *SNARE* improves on prior detection techniques because it automatically infers the reputations of spammers from their sending behavior, which tends to be more invariant than the contents

of the message or the IP address from which they are sending.

Monitoring the initial DNS behavior and hosting of spammer domains. In Chapter 4, we monitor the DNS behavior of malicious domains, as identified by appearance in a spam trap, shortly after the domains are registered. We study two behaviors: the DNS infrastructure associated with the domains, as is observable from resource records; and DNS lookup patterns from networks that look up these domains initially. In particular, we analyze DNS information of `.com` and `.net` domains collected from the Verisign top-level domain servers, which allows a *global view* across the Internet, as opposed to a view from any single network. We identify a few regions of IP addresses and tainted ASes that are used heavily by bad domains to host resource records. We also discover that miscreant domains clearly cluster by querying networks. Those features are often evident before any attack even takes place, which can serve as the basis for a DNS-based early warning system for attacks.

Understanding the domain registration behavior of spammers. In Chapter 5, we examine the registration process of domains in `.com`, the largest top-level domain, in conjunction with several large blacklist feeds as indicators of spammer domains. Miscreants expose unusual registration behavior, due to economic concerns (price of registration) as well as the ease of management. We explore the characteristics of various aspects of domain registration, such as registrars, domain life cycles, registration bursts, and naming conventions. Our findings suggest steps that registries or registrars could use to frustrate the efforts of miscreants to acquire domains in bulk.

Proactive detection of spammer domains at time-of-registration. In Chapter 6, we present PREDATOR, a proactive detection system that can accurately and automatically identify spammer domains at time-of-registration, rather than later at time-of-use. PREDATOR relies on the observation that miscreants register large volumes of domains to maintain revenue and ensure agility, which exhibits characteristic registration behavior. We derive temporal, lexical, and hosting features about domain registrations, and incorporate these features into a fast classification method. PREDATOR can achieve high accuracy

with a low false positive, and raise early warnings in advance of domains being actually used in spam activity.

1.5 Lessons Learned

This dissertation characterizes spam-related activity and develops new detection systems. We discuss several lessons during our research and hope they are useful for future study. We further introduce how prior research handled those problems in Section 2.2.

Identifying effective and robust features is challenging. To stay undetected and remain profitable, spammers are creative to increase attack agility on various technical and business components. In particular, spammers deploy botnets to effectively transmit spam emails and obtain IP dynamics. Miscreants also register large number of domains to host spam-advertised sites to evade blacklisting effort. A necessary step of developing better mitigation approaches is to first characterize the operation of spam campaigns and identify useful features. The characterization process is essential to identify the bottleneck of spam-related activity. After miscreants know the existence of the detection mechanisms, they will try to evade them, which forms the arms race between cybercriminals and security researchers. We aim to extract features that spammers can not easily change without considerably reducing the negative impact that they cause. In our work, we perform measurement-based analysis to understand the attributes of spammers, which becomes the foundation to design detection systems. For example, we investigate network-level features of spam from botnets, combine them in *SNARE* to achieve satisfactory detection, and further evaluate feature importance and relationship; We also make a holistic analysis of domain registration process and demonstrate unusual behavior of spammers, which leads to the development of PREDATOR, a registration-based detection system. Identifying effective and robust features is challenging, but is critical to establish better detection and mitigation approaches against spam-related activity.

Obtaining large-scale ground truth is hard. We need labelled data to indicate what activities are spam-related to study spammer characteristics or to evaluate the performance of detection systems ("ground truth"). Obtaining large-scale ground truth is a paradox:

If there exists perfect ground truth in a timely manner, we could have already greatly mitigated spam-related activities; On the other hand, without good-quality labelled data, we have more difficulty to design and evaluate new defense approaches. A solution is to sample and manually validate the data, but it can only provide limited amount of data. We acquire satisfactory ground truth by collaborating with companies or collecting from public services, but privacy is a concern and sometimes only meta data are available. How to obtain and share ground truth remains an open problem in anti-spam research.

1.6 Bibliographic Notes

The work of *SNARE* in Chapter 3 appeared at USENIX Security 2009 [44]. The work on monitoring initial DNS behavior of spammer domains, presented in Chapter 4, was published at 2011 ACM Internet Measurement Conference [43]. The work on analyzing spammer domain registrations, presented in Chapter 5, appeared at 2013 ACM Internet Measurement Conference [45]. Chapter 6 contains the work of PREDATOR that is under submission.

CHAPTER II

BACKGROUND AND RELATED WORK

This chapter introduces spam components in more detail, and discusses related work in detecting spam-related activity.

2.1 The Evolution and Components of Spam

Spam first emerged in the early 1990s, and now has become a multi-faceted business to gain illicit profits or benefit. We provide a overview of the technical and business components within spam ecosystem, which is essential for further spam mitigation and defense. In this dissertation, our work focuses on the problem within the components of *spam delivery* and *naming and hosting*.

Spam delivery. The delivery of spam is the most visible part to users within spam ecosystem, and has evolved considerably over the past two decades. Early spam relied on their own servers or open relays to deliver outgoing email [87]. However, blacklists made it ineffective to send spam from fixed IP addresses. To bypass this countermeasure, spammers started to use botnets, large networks of compromised computers, to send spam. Botnets supply fresh IP addresses with newly infected machines. Either spam is directly sent from bots, or attackers use bots to login accounts in large email providers, such as Yahoo! Mail, and send spam email through these accounts. Nowadays, botnets are the major carrier to deliver spam, responsible for 81% of email spam on the Internet in 2011 [114]. Although the takedowns of botnets have considerably reduced worldwide spam volume, spammers quickly re-established their spam operation by assembling new botnets [108]. In early 2014, some spam messages have been observed to originate from compromised home devices, such as refrigerators or smart TVs [106], which indicates that spammers are aggressively targeting more platforms and devices for spam delivery.

Naming and hosting. Spammers usually rely on potential victims clicking on embedded

URLs in the spam messages and being redirected to point-of-sale or malicious Web sites. A URL includes a domain name to locate the corresponding Web site, which is supported by Domain Name System (DNS) and Web hosting. DNS provides a critical service to map humanly memorable names to IP addresses. The name space of domains is organized in hierarchical levels, and the names on each level contain up to 63 characters [81]. The registrations are through ICANN accredited registrars or some resellers, and managed by registries [54]. The domain space provides higher attacking resilience for spammers compared to IP addresses—spammers can register any names they want as long as the names have not been taken by others. DNS nameservers are responsible to receive queries to domains and resolve them to IP addresses. A visitor then can make connection to the IP address of the designated Web server and access the Web page. Spammers either run hosting infrastructure by themselves or rely on third-party services to support the spam-advertised sites. Another technique that attackers use to increase resilience against blacklisting or takedown effort is fast flux, which changes the IP addresses assigned to domains with high frequency [49, 67].

Affiliate program. As the spam business becomes larger and more sophisticated, its operation has started to involve affiliates. Spammers often join affiliate programs of online stores. Affiliate programs provide links or page templates to spammers, and spammers send spam through their own email delivery infrastructure. For every purchase attracted to the spam-advertised sites, spammers receive a cut of the final revenue [15, 72, 101]. The estimate of the gross revenue of a pharmaceutical affiliate program (such as GlavMed and EvaPharmacy) ranges from 300,000 to 1 million dollars per month [62], and affiliates get a share of 30–50% in the pharmaceutical market. The market scheme of affiliate programs frees spammers from handling the back end of monetizing victims, and allows them to focus on generating more effective spam techniques.

Realization. A prevalent category of spam is to promote and sell merchandise, most of which are illicit products, such as pharmaceuticals, replica luxury goods, or counterfeit software. After users are enticed to the spam-advertised sites by affiliates and put items into shopping carts, the process is handed over to the operators of the affiliate program.

The merchants acquire money from users through payment systems, such as credit card payment service. A recent study shows that just a small number of banks take care of the payment processing for spam-related merchants [72], which is a bottleneck in the spam ecosystem. To fulfill the order, the product suppliers will package the items and ship them to the customers if the order is physical products. Even though the customers finally receive the orders that they purchase, the product quality is not guaranteed as in the branded store and can endanger public safety [123].

This dissertation focuses on the problem of email spam, but the similar technical and business components occur in other advertising vectors as spammers target more platforms, including instant messaging, social networks, and search engines.

2.2 Detection Methods

The anti-spam community has spent much effort in mitigating spam-related activity. We introduce prior research on detection approaches.

2.2.1 Content-based Detection

A traditional spam mitigation approach is to examine message body and headers, whether the email content is anomalous. Content-based detection was first based on heuristic rules of matching illicit words, and evolved into systems based on supervised learning approaches including Bayesian inference [2, 78, 100] and Support Vector Machines [27, 128]. These techniques use labeled examples of spam and non-spam to train a classifier and determine how likely a message is spam. Content-based detection has successfully reduced spam amount, but miscreants responded with circumvention methods. Spammers use deliberately misspelled words or include sentences that are common in benign email to confuse content-based systems [50]. In addition to text, spammers can attach other media, such as images or PDF files, to contain advertising content. The content-based systems need to inspect the whole message for detection and require constant maintenance from operators to keep up with the new spam templates.

Recent content-based spam detection techniques have involved analyzing URLs embedded in spam messages. The overhead to process URLs is lower and the signatures are

simpler. Xie *et al.* analyzed human classified Hotmail email messages and designed regular expression to identify spam-related URLs sent by botnets [126]. Unfortunately, detection by parsing URLs has the similar problem as detection based on analyzing message content. Spammers keeps generating new URLs and registering more domains for evasion.

2.2.2 Sender-based Detection

A sender-based detection is to track the IP addresses of hosts being used to send spam. Such blacklists are commonly referred to as DNS-based IP blacklists (DNSBLs) [73], such as Spamhaus or URI [110, 117], since the major method to check against the blacklists is through DNS queries. An email being sent from a blacklisted IP address is considered as spam. DNSBLs are very useful to block open relays and proxies which transmit spam [60]. However, botnets have been widely used for spam sending, which makes IP blacklists less effective [95]. Spam can come from previously unseen IP addresses, due to dynamic IP addresses or new infections of bots.

A related approach is to inspect the sending behavior of spammers, as opposed to directly checking the IP addresses of the senders. Ramachandran *et al.* studied the data collected at large spam sinkholes and analyzed the *network-level* behavior of spamming botnets [93]. They found that most spam was being sent from a few regions of IP address space and some spammers hijacked large BGP prefixes for short periods to send spam. In Chapter 3, we perform a detailed characterization of network-level features and incorporate them with a supervised learning technique to build a detection system.

2.2.3 DNS-based Detection

DNS infrastructure provides rich information to detect botnet, malware, and spam-related activity. Previous studies have used the mechanism of querying the DNS servers to check the zones' resource records. Holz *et al.* found that 30% of the domains advertised in spam were hosted via fast-flux networks, and they investigated the diversity of the A-type records returned in the lookups to identify fast-flux service networks [49]. Konte *et al.* studied the spam-advertised domains collected at a large spam sinkhole, and found that the DNS records of fast-flux domains changed on shorter time intervals than their TTL values

and resolved to more widely distributed IP addresses [66]. The first studies of DNS lookup behavior at a local resolver were performed by Danzig *et al.* [20] and Jung *et al.* [61]; both of these studies examined lookup behavior from the vantage point of lookups to a single local resolver, and did not attempt to characterize how these lookup patterns differed for malicious domains. Notos relied on passive monitoring of recursive DNS traffic, modeled network and zone behaviors, and used automated classification and clustering algorithms to detect malware-related domains [4]. EXPOSURE extracted behavioral features from recursive DNS traffic without relying on large amounts of historical information, and used the decision tree algorithm to detect domain names involved in malicious activity (such as for botnet command and control, spamming, and phishing) [9]. Such a view of DNS lookup behavior below the DNS resolvers is valuable, but this vantage point cannot reveal coordinated behavior across multiple networks, and it relies first on an attack to take place or hosts being compromised before it can detect any malicious domains. Antonakakis *et al.* analyzed passive DNS traffic at the authoritative name servers or top-level domain servers, and used random forest to classify blacklisted malware domains based on the diversity and reputation of the querying IP addresses [5]. In Chapter 4, we study DNS records and lookup traffic from the perspective of top-level domains and focus exclusively on the newly registered domain.

Recently, a number of research efforts have studied domain registration patterns. Kreibich *et al.* infiltrated the Storm botnet, and found that spammers changed domains frequently to avoid detection and the average time from a domain's registration to its use was 21 days [68]. Coull *et al.* studied the registration abuse phenomenon, including domain-name speculation, tasting, and front-running (where registrars abuse insider information to obtain a domain, thus locking out other registrars) [17]. They showed that a significant financial incentive of abusive registrations is to resell popular domains and to use parking services to generate advertising revenue. Liu *et al.* found that registry policy changes and registrar actions had real effect to evict spammers from a top-level domain, but spammers could transition to other registrars or top-level domains [74]. Felegyhazi *et al.* inferred groups of

malicious domains based on the DNS servers and daily registrations associated with known-bad seed domains [30]. In Chapter 5, we provide a significantly more granular analysis of registration patterns of spammer domains. Chapter 6 further presents an early warning system to identify spammer domains at time-of-registration.

2.2.4 Web-based Detection

Spammers mainly rely on embedded URLs to entice victim users to malicious Web sites. A detection technique is through automatic URL crawling tools. Thomas *et al.* built a large-scale system to crawl URLs in email and Twitter feeds to detect malicious messages [116]. Cova *et al.* used a emulated browser to detect malicious JavaScript code [18]. Wu *et al.* proposed detection algorithms based on the link structure leading to the pages [124]. Some others are based on the presence of cloaking and redirection [70, 122, 125]. The problem of URL crawling is the processing load is high, requiring to fetch the Web pages and deal with dynamics. If crawls are from a limited set of hosts, attackers can become aware and take countermeasure actions. The registration-based detection system that we introduce in Chapter 6 do not need to visit the Web pages and can also serve as an early warning system to narrow down the suspect URLs for crawling.

2.2.5 Other Mitigation Methods and Effort

Most majority of spam messages were delivered through botnets. An effective way to mitigate spam is to take down botnets which contain a set of compromised machines to send spam. Rajab *et al.* revealed that 27% of unwanted Internet traffic and malicious activities can be directly attributed to botnets by analyzing bot binaries and tracking IRC botnets [92]. Dispatcher analyzed the memory buffers and reconstructed the command-and-control (C&C) messages to infiltrate botnets [14]. Gu *et al.* monitored network traffic and detected bots based correlation between C&C channels and malicious activity [41, 42, 127]. Ramachandran *et al.* studied the prevalence of DNSBL reconnaissance performed by botmasters to determine their bots' blacklist status, and suggested the possibility of using counter-intelligence to discover likely bots. Prior studies on botnet and its detection have provided useful insight to our work.

Another line of recent research focusing on understanding and disrupting spam from an economic perspective in addition to the technical point of view. Kanich *et al.* infiltrated the Storm botnet and measured the conversion rate of spam [15]. Levchenko *et al.* made a holistic analysis of spam value chain and found only a small number of banks willing to process payment for the spam-related merchants. McCoy *et al.* analyzed customer demand and overheads in spam cost model by using transaction logs of pharmaceutical affiliate programs [77]. Economic intervention is promising to disrupt spam-related activity. Our work on early detection in this dissertation also attempts to raise the cost of spammers and further making spam campaigns less profitable.

CHAPTER III

SNARE: FILTERING SPAM WITH NETWORK-LEVEL FEATURES

3.1 Introduction

Spam filtering systems use two mechanisms to filter spam: content filters, which classify messages based on the contents of a message; and sender reputation, which maintains information about the IP address of a sender as an input to filtering. Content filters (e.g., [39, 51]) can block certain types of unwanted email messages, but they can be brittle and evadable, and they require analyzing the contents of email messages, which can be expensive. Hence, spam filters also rely on *sender reputation* to filter messages; the idea is that a mail server may be able to reject a message purely based on the reputation of the sender, rather than the message contents. DNS-based blacklists such as Spamhaus [110] maintain lists of IP addresses that are known to send spam. Unfortunately, these blacklists can be both incomplete and slow-to-respond to new spammers [93]. This unresponsiveness becomes more serious as both botnets and BGP route hijacking make it easier for spammers to dynamically obtain new, unlisted IP addresses [93, 94]. Network administrators are still searching for spam-filtering mechanisms that are both *lightweight* (i.e., they do not require detailed message or content analysis) and *automated* (i.e., they do not require manual update, inspection, or verification).

Towards this goal, this chapter presents *SNARE* (Spatio-temporal Network-level Automatic Reputation Engine), a sender reputation engine that can accurately and automatically classify email senders based on lightweight, network-level features that can be determined early in a sender's history—sometimes even upon seeing only a single packet. *SNARE* relies on the intuition that about $75 - 95\%$ of spam email can be attributed to botnets, which often exhibit unusual sending patterns that differ from those of legitimate email senders. *SNARE* classifies senders based on *how* they are sending messages (i.e., traffic patterns), rather than *who* the senders are (i.e., their IP addresses). In other words, *SNARE* rests

on the assumption that there are lightweight network-level features that can differentiate spammers from legitimate senders; this chapter finds such features and uses them to build a system for automatically determining an email sender's reputation.

SNARE bears some similarity to other approaches that classify senders based on network-level behavior [8,38,58,69,94], but these approaches rely on inspecting the message contents, gathering information across a large number of recipients, or both. In contrast, *SNARE* is based on *lightweight* network-level features, which could allow it to scale better and also to operate on higher traffic rates. In addition, *SNARE* is *more accurate* than previous reputation systems that use network-level behavioral features to classify senders: for example, *SNARE*'s false positive rate is an order of magnitude less than that in previous work [94] for a similar detection rate. It is the first reputation system that is both as accurate as existing static IP blacklists and automated to keep up with changing sender behavior.

Despite the advantages of automatically inferring sender reputation based on "network-level" features, a major hurdle remains: We must identify *which features* effectively and efficiently distinguish spammers from legitimate senders. Given the massive space of possible features, finding a collection of features that classifies senders with both low false positive and low false negative rates is challenging. This chapter identifies thirteen such network-level features that require varying levels of information about senders' history.

Different features impose different levels of overhead. Thus, we begin by evaluating features that can be computed purely locally at the receiver, with no information from other receivers, no previous sending history, and no inspection of the message itself. We found several features that fall into this category are surprisingly effective for classifying senders, including: The AS of the sender, the geographic distance between the IP address of the sender and that of the receiver, the density of email senders in the surrounding IP address space, and the time of day the message was sent. We also looked at various aggregate statistics across messages and receivers (e.g., the mean and standard deviations of messages sent from a single IP address) and found that, while these features require slightly more computation and message overhead, they do help distinguish spammers from legitimate senders as well. After identifying these features, we analyze the relative importance of

these features and incorporate them into an automated reputation engine, based on the *RuleFit* [32] ensemble learning algorithm.

In addition to presenting the first automated classifier based on network-level features, this chapter presents several additional contributions. First, we presented a detailed study of various network-level characteristics of both spammers and legitimate senders, a detailed study of how well each feature distinguishes spammers from legitimate senders, and explanations of why these features are likely to exhibit differences between spammers and legitimate senders. Second, we use state-of-the-art ensemble learning techniques to build a classifier using these features. Our results show that *SNARE*'s performance is at least as good as static DNS-based blacklists, achieving a 70% detection rate for about a 0.2% false positive rate. Using features extracted from a single message and aggregates of these features provides slight improvements, and adding an AS "whitelist" of the ASes that host the most commonly misclassified senders reduces the false positive rate to 0.14%. This accuracy is roughly equivalent to that of existing static IP blacklists like SpamHaus [110]; the advantage, however, is that *SNARE* is *automated*, and it characterizes a sender based on its sending *behavior*, rather than its IP address, which may change due to dynamic addressing, newly compromised hosts, or route hijacks. Although *SNARE*'s performance is still not perfect, we believe that the benefits are clear: Unlike other email sender reputation systems, *SNARE* is both automated and lightweight enough to operate solely on network-level information. Third, we provide a deployment scenario for *SNARE*. Even if others do not deploy *SNARE*'s algorithms exactly as we have described, we believe that the collection of network-level features themselves may provide useful inputs to other commercial and open-source spam filtering appliances.

The rest of this chapter is organized as follows. Section 3.2 presents background on existing sender reputation systems and a possible deployment scenario for *SNARE* and introduces the ensemble learning algorithm. Section 3.3 describes the network-level behavioral properties of email senders and measures first-order statistics related to these features concerning both spammers and legitimate senders. Section 3.4 evaluates *SNARE*'s performance using different feature subsets, ranging from those that can be determined from a

single packet to those that require some amount of history. We investigate the potential to incorporate the classifier into a spam-filtering system in Section 3.5. Section 3.6 discusses evasion and other limitations, and Section 3.7 concludes.

3.2 Background

In this section, we provide background on existing sender reputation mechanisms, present motivation for improved sender reputation mechanisms , and describe a classification algorithm called *RuleFit* to build the reputation engine. We also describe McAfee's TrustedSource system, which is both the source of the data used for our analysis and a possible deployment scenario for *SNARE*.

3.2.1 Email Sender Reputation Systems

Today's spam filters look up IP addresses in DNS-based blacklists to determine whether an IP address is a known source of spam at the time of lookup. One commonly used public blacklist is Spamhaus [110]; other blacklist operators include SpamCop [109] and SORBS [104]. Current blacklists have three main shortcomings. First, they only provide reputation at the granularity of IP addresses. Unfortunately, as earlier work observed [94], IP addresses of senders are dynamic: roughly 10% of spam senders on any given day have not been previously observed. This study also observed that many spamming IP addresses will go inactive for several weeks, presumably until they are removed from IP blacklists. This dynamism makes maintaining responsive IP blacklists a manual, tedious, and inaccurate process; they are also often coarse-grained, blacklisting entire prefixes—sometimes too aggressively—rather than individual senders. Second, IP blacklists are typically incomplete: A previous study has noted that as much as 20% of spam received at spam traps is not listed in any blacklists [93]. Finally, they are sometimes inaccurate: Anecdotal evidence is rife with stories of IP addresses of legitimate mail servers being incorrectly blacklisted (e.g., because they were reflecting spam to mailing lists). To account for these shortcomings, commercial reputation systems typically incorporate additional data such as SMTP metadata or message fingerprints to mitigate these shortcomings [1]. Previous work introduced "behavioral blacklisting" and developed a spam classifier based on a single behavioral feature:

Table 1: Description of data used from the McAfee dataset.

Field	*Description*
timestamp	UNIX timestamp
ts_server_name	Name of server that handles the query
score	Score for the message based on a combination of anti-spam filters
source_ip	Source IP in the packet (DNS server relaying the query to us)
query_ip	The IP being queried
body_length	Length of message body
count_taddr	Number of To-addresses

the number of messages that a particular IP address sends to each recipient domain [94]. This chapter builds on the main theme of behavioral blacklisting by finding better features that can classify senders earlier and are more resistant to evasion.

3.2.2 Data and Deployment Scenario

This section describes McAfee's TrustedSource email sender reputation system. We describe how we use the data from this system to study the network-level features of email senders and to evaluate *SNARE*'s classification. We also describe how *SNARE*'s features and classification algorithms could be incorporated into a real-time sender reputation system such as TrustedSource.

Data source. TrustedSource is a commercial reputation system that allows lookups on various Internet identifiers such as IP addresses, URLs, domains, or message fingerprints. It receives query feedback from various different device types such as mail gateways, Web gateways, and firewalls. We evaluated *SNARE* using the query logs from McAfee's TrustedSource system over a fourteen-day period from October 22–November 4, 2007. Each received email generates a lookup to the TrustedSource database, so each entry in the query log represents a single email that was sent from some sender to one of McAfee's TrustedSource appliances. Due to the volume of the full set of logs, we focused on logs from a single TrustedSource server, which reflects about 25 million email messages as received from over 1.3 million IP addresses each day. These messages were reported from approximately 2,500 distinct TrustedSource appliances geographically distributed around the world. While there

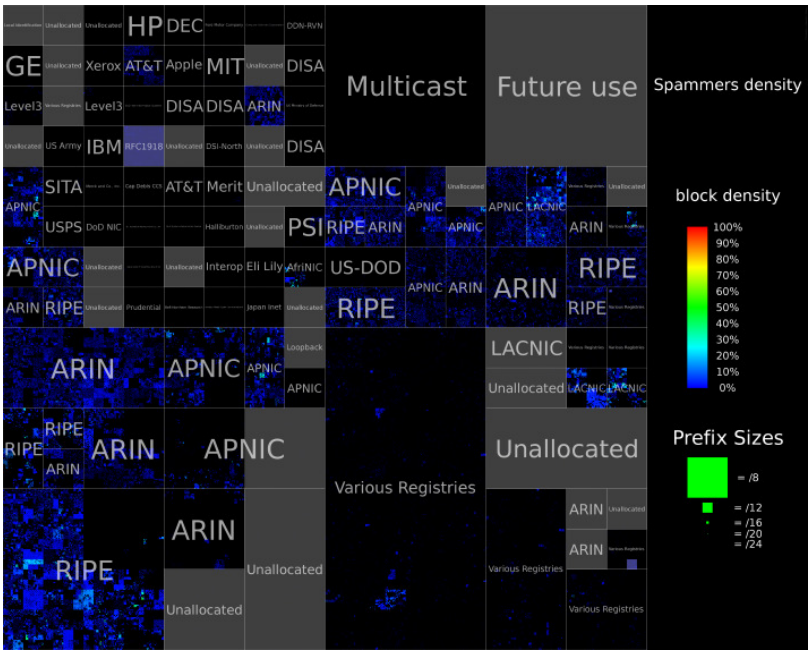


Figure 3: Distribution of senders' IP addresses in Hilbert space for the one-week period (October 22–28, 2007) of our feature study. (The grey blocks are unused IP space.)

is not a precise one-to-one mapping between domains and appliances, and we do not have a precise count for the number of unique domains, the number of domains is roughly of the same order of magnitude.

The logs contain many fields with *metadata for each email message*; Figure 1 shows a subset of the fields that we ultimately use to develop and evaluate *SNARE*'s classification algorithms. The `timestamp` field reflects the time at which the message was received at a TrustedSource appliance in some domain; the `source_ip` field reflects the source IP of the machine that issued the DNS query (i.e., the recipient of the email). The `query_ip` field is the IP address being queried (i.e., the IP address of the email sender). The IP addresses of the senders are shown in the Hilbert space, as in Figure 3[1], where each pixel represents a /24 network prefix and the intensity indicates the observed IP density in each block. The distribution of the senders' IP addresses shows that the TrustedSource database collocated a representative set of email across the Internet. We use many of the other features in Figure 1 as input to *SNARE*'s classification algorithms.

[1] A larger figure is available at `http://www.gtnoise.net/snare/hilbert-ip.png`.

To help us label senders as either spammers or legitimate senders for both our feature analysis (Section 3.3) and training (Sections 3.2.3 and 3.4), the logs also contain *scores* for each email message that indicate how McAfee scored the email sender based on its current system. The `score` field indicates McAfee's sender reputation score, which we stratify into five labels: certain ham, likely ham, certain spam, likely ham, and uncertain. Although these scores are not perfect ground truth, they do represent the output of both manual classification and continually tuned algorithms that also operate on more heavy-weight features (e.g., packet payloads). Our goal is to develop a fully automated classifier that is as accurate as TrustedSource but (1) classifies senders *automatically* and (2) relies only on lightweight, evasion-resistant network-level features.

Deployment and data aggregation scenario. Because it operates only on network-level features of email messages, *SNARE* could be deployed either as part of TrustedSource or as a standalone DNSBL. Some of the features that *SNARE* uses rely on aggregating sender behavior across a wide variety of senders. To aggregate these features, a monitor could collect information about the global behavior of a sender across a wide variety of recipient domains. Aggregating this information is a reasonably lightweight operation: Since the features that *SNARE* uses are based on simple features (i.e., the IP address, plus auxiliary information), they can be piggybacked in small control messages or in DNS messages (as with McAfee's TrustedSource deployment).

3.2.3 Supervised Learning: RuleFit

Ensemble learning: *RuleFit*. Learning ensembles have been among the popular predictive learning methods over the last decade. Their structural model takes the form

$$F(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m f_m(\mathbf{x}) \tag{1}$$

Where $\mathbf{x}$ are input variables derived form the training data (spatio-temporal features); $f_m(\mathbf{x})$ are different functions called ensemble members ("base learner") and M is the size of the ensemble; and $F(\mathbf{x})$ is the predictive output (labels for "spam" or "ham"), which takes a linear combination of ensemble members. Given the base learners, the technique

determines the parameters for the learners by regularized linear regression with a "lasso" penalty (to penalize large coefficients a_m).

Friedman and Popescu proposed *RuleFit* [32] to construct regression and classification problems as linear combinations of simple rules. Because the number of base learners in this case can be large, the authors propose using the rules in a decision tree as the base learners. Further, to improve the accuracy, the variables themselves are also included as basis functions. Moreover, fast algorithms for minimizing the loss function [31] and the strategy to control the tree size can greatly reduce the computational complexity.

Variable importance. Another advantage of *RuleFit* is the interpretation. Because of its simple form, each rule is easy to understand. The relative importance of the respective variables can be assessed after the predictive model is built. Input variables that frequently appear in important rules or basic functions are deemed more relevant. The importance of a variable x_i is given as importance of the basis functions that correspond directly to the variable, plus the average importance of all the other rules that involve x_i. The *RuleFit* paper has more details [32]. In Section 3.4.3, we show the relative importance of these features.

Comparison to other algorithms. There exist two other classic classifier candidates, both of which we tested on our dataset and both of which yielded poorer performance (i.e., higher false positive and lower detection rates) than *RuleFit*. Support Vector Machine (SVM) [12] has been shown empirically to give good generalization performance on a wide variety of problems such as handwriting recognition, face detection, text categorization, etc. On the other hand, they do require significant parameter tuning before the best performance can be obtained. If the training set is large, the classifier itself can take up a lot of storage space and classifying new data points will be correspondingly slower since the classification cost is $O(S)$ for each test point, where S is the number of support vectors. The computational complexity of SVM conflicts with *SNARE*'s goal to make decision quickly (at line rate). Decision trees [90] are another type of popular classification method. The resulting classifier is simple to understand and faster, with the prediction on a new test point taking $O(\log(N))$, where N is the number of nodes in the trained tree. Unfortunately,

decision trees compromise accuracy: its high false positive rates make it less than ideal for our purpose.

3.3 Network-level Features

In this section, we explore various spatio-temporal features of email senders and discuss why these properties are relevant and useful for differentiating spammers from legitimate senders. We categorize the features we analyze by increasing level of overhead:

- *Single-packet features* are those that can be determined with no previous history from the IP address that *SNARE* is trying to classify, and given only a *single packet* from the IP address in question (Section 3.3.1).
- *Single-header and single-message features* can be gleaned from a single SMTP message header or email message (Section 3.3.2).
- *Aggregate features* can be computed with varying amounts of history (i.e., aggregates of other features) (Section 3.3.3).

Each class of features contains those that may be either purely local to a single receiver or aggregated across multiple receivers; the latter implies that the reputation system must have some mechanism for aggregating features in the network. In the following sections, we describe features in each of these classes, explain the intuition behind selecting that feature, and compare the feature in terms of spammers vs. legitimate senders.

No single feature needs to be perfectly discriminative between ham and spam. The analysis below shows that it is unrealistic to have a single perfect feature to make optimal resolution. As we describe in Section 3.2.3, *SNARE*'s classification algorithm uses a *combination* of these features to build the best classifier. We do, however, evaluate *SNARE*'s classifier using these three different classes of features to see how well it can perform using these different classes. Specifically, we evaluate how well *SNARE*'s classification works using only single-packet features to determine how well such a lightweight classifier would perform; we then see whether using additional features improves classification.

3.3.1 Single-Packet Features

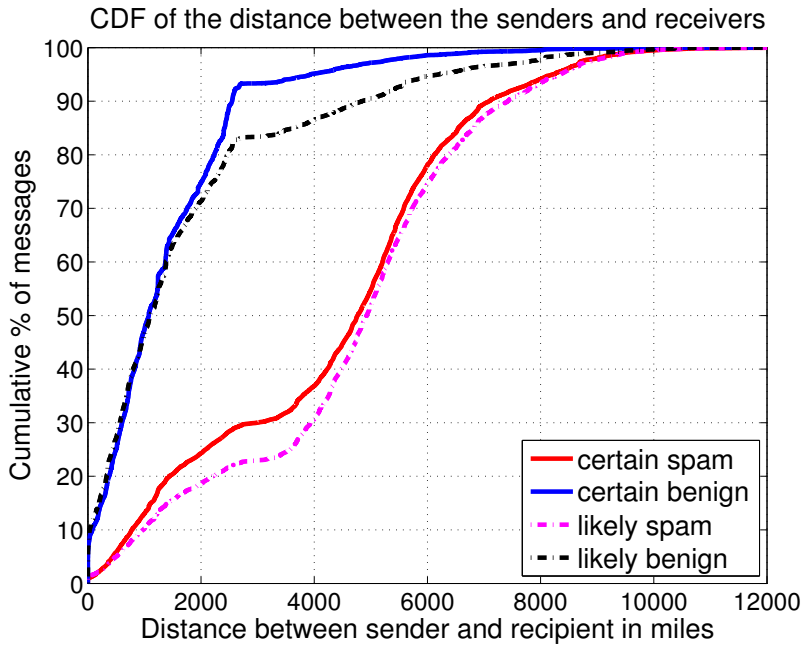
In this section, we discuss some properties for identifying a spammer that rely only on a single packet from the sender IP address. In some cases, we also rely on auxiliary information, such as routing table information, sending history from neighboring IP addresses, etc., not solely information in the packet itself. We first discuss the features that can be extracted from just a single IP packet: the geodesic distance between the sender and receiver, sender neighborhood density, probability ratio of spam to ham at the time-of-day the IP packet arrives, AS number of the sender and the status of open ports on the machine that sent the email. The analysis is based on the McAfee's data from October 22–28, 2007 inclusive (7 days).[2]

3.3.1.1 Sender-receiver geodesic distance: Spam travels further

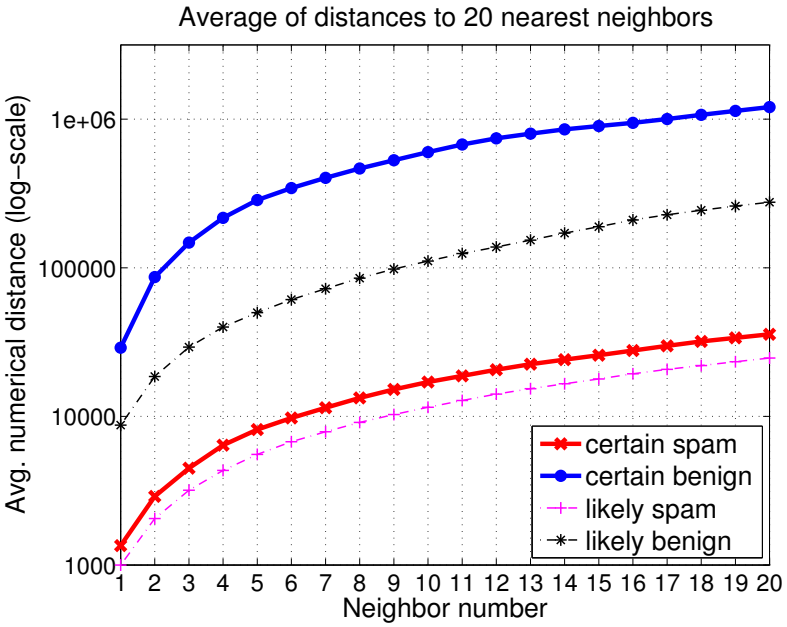
Recent studies suggest that social structure between communicating parties could be used to effectively isolate spammers [11,37]. Based on the findings in these studies, we hypothesized that legitimate emails tend to travel shorter geographic distances, whereas the distance traveled by spam will be closer to random. In other words, a spam message may be just as likely to travel a short distance as across the world.

Figure 4(a) shows that our intuition is roughly correct: the distribution of the distance between the sender and the target IP addresses for each of the four categories of messages. The distance used in these plots is the geodesic distance, that is, the distance along the surface of the earth. It is computed by first finding the physical latitude and longitude of the source and target IP using the MaxMind's GeoIP database [34] and then computing the distance between these two points. These distance calculations assume that the earth is a perfect sphere. For *certain ham*, 90% of the messages travel about 2,500 miles or less. On the other hand, for *certain spam*, only 28% of messages stay within this range. In fact, about 10% of spam travels more than 7,000 miles, which is a quarter of the earth's circumference at the equator. These results indicate that geodesic distance is a promising metric for

[2]The evaluation in Section 3.4 uses the data from October 22–November 4, 2007 (14 days), some of which are not included in the data trace used for measurement study.



(a) Geodesic distance between the sender and recipient's geographic location.



(b) Average of numerical distances to the 20 nearest neighbors in the IP space.

Figure 4: Spatial differences between spammers and legitimate senders.

distinguishing spam from ham, which is also encouraging, since it can be computed quickly using just a single IP packet.

3.3.1.2 Sender IP neighborhood density: Spammers are surrounded by other spammers

Most spam messages today are generated by botnets [93,126]. For messages originating from the same botnet, the infected IP addresses may all lie close to one another in numerical space, often even within the same subnet. One way to detect whether an IP address belongs to a botnet is to look at the past history and determine if messages have been received from other IPs in the same subnet as the current sender, where the subnet size can be determined experimentally. If many different IPs from the same subnet are sending email, the likelihood that the whole subnet is infested with bots is high.

The problem with simply using subnet density is that the frame of reference does not transcend the subnet boundaries. A more flexible measure of *email sender density* in an IP's neighborhood is the distances to its k nearest neighbors. The distance to the k nearest neighbors can be computed by treating the IPs as set of numbers from 0 to $2^{32}-1$ (for IPv4) and finding the nearest neighbors in this single dimensional space. We can expect these distances to exhibit different patterns for spam and ham. If the neighborhood is *crowded*, these neighbor distances will be small, indicating the possible presence of a botnet. In normal circumstances, it would be unusual to see a large number of IP addresses sending email in a small IP address space range (one exception might be a cluster of outbound mail servers, so choosing a proper threshold is important, and an operator may need to evaluate which threshold works best on the specific network where *SNARE* is running).

The average distances to the 20 nearest neighbors of the senders are shown in Figure 4(b). The x-axis indicates how many nearest neighbors we consider in IP space, and the y-axis shows the average distance in the sample to that many neighbors. The figure reflects the fact that a large majority of spam originates from hosts have high email sender density in a given IP region. The distance to the k^{th} nearest neighbor for spam tends to be much shorter on average than it is for legitimate senders, indicating that spammers generally reside in areas with higher densities of email senders (in terms of IP address space).

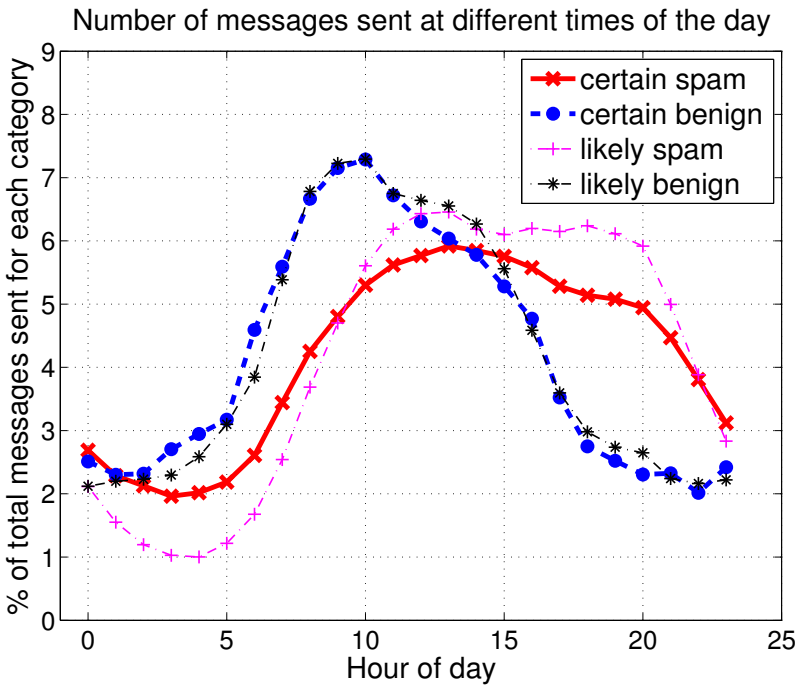


Figure 5: Differences in diurnal sending patterns of spammers and legitimate senders.

3.3.1.3 Time-of-day: Spammers send messages according to machine off/on patterns

Another feature that can be extracted using information from a single packet is the time of day when the message was sent. We use the *local* time of day at the sender's physical location, as opposed to Coordinated Universal Time (UTC). The intuition behind this feature is that local legitimate email sending patterns may more closely track "conventional" diurnal patterns, as opposed to spam sending patterns.

Figure 5 shows the relative percentage of messages of each type at different times of the day. The legitimate senders and the spam senders show different diurnal patterns. Two times of day are particularly striking: the relative amount of ham tends to ramp up quickly at the start of the workday and peaks in the early morning. Volumes decrease relatively quickly as well at the end of the workday. On the other hand spam increases at a slower, steadier pace, probably as machines are switched on in the morning. The spam volume stays steady throughout the day and starts dropping around 9:00 p.m., probably when machines are switched off again. In summary, legitimate senders tend to follow workday cycles, and

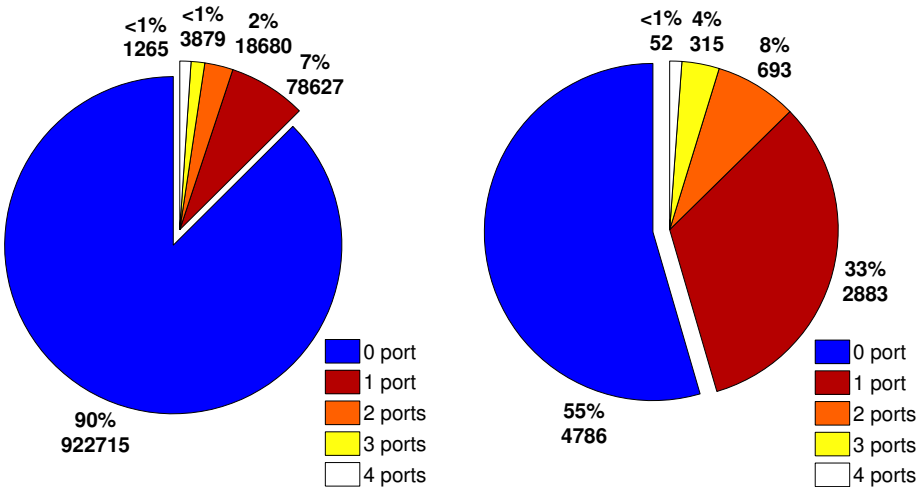
spammers tend to follow machine power cycles.

To use the timestamp as a feature, we compute the probability ratio of spam to ham at the time of the day when the message is received. First, we compute the *a priori* spam probability $p_{s,t}$ during some hour of the day t, as $p_{s,t} = n_{s,t}/n_s$, where $n_{s,t}$ is the number of spam messages received in hour t, and n_s is the number of spam messages received over the entire day. We can compute the *a priori* ham probability for some hour t, $p_{h,t}$ in a similar fashion. The probability ratio, r_t is then simply $p_{s,t}/p_{h,t}$. When a new message is received, the precomputed spam to ham probability ratio for the corresponding hour of the day at the senders timezone, r_t can be used as a feature; this ratio can be recomputed on a daily basis.

3.3.1.4 AS number of sender: A small number of ASes send a large fraction of spam

As previously mentioned, using IP addresses to identify spammers has become less effective for several reasons. First, IP addresses of senders are often transient. The compromised machines could be from dial-up users, which depend on dynamic IP assignment. If spam comes form mobile devices (like laptops), the IP addresses will be changed once the people carry the devices to a different place. In addition, spammers have been known to adopt stealthy spamming strategies where each bot only sends several spam to a single target domain, but overall the botnets can launch a huge amount of spam to many domains [93]. The low emission-rate and distributed attack requires to share information across domains for detection.

On the other hand, previous study revealed that a significant portion of spammers come from a relatively small collection of ASes [93]. More importantly, the ASes responsible for spam differ from those that send legitimate email. As a result, the AS numbers of email senders could be a promising feature for evaluating the senders' reputation. Over the course of the seven days in our trace, more than 10% of unique spamming IPs (those sending certain spam) originated from only 3 ASes; the top 20 ASes host 42% of spamming IPs. Although our previous work noticed that a small number of ASes originated a large fraction of spam [93], we believe that this is the first work to suggest using the AS number



(a) spam (b) benign mail

Figure 6: Distribution of number of open ports on hosts sending spam and legitimate mail.

of the email sender as input to an automated classifier for sender reputation.

3.3.1.5 Status of service ports: Legitimate mail tends to originate from machines with open ports

We hypothesized that legitimate mail senders may also listen on other ports besides the SMTP port, while bots might not; our intuition is that the bots usually send spam directly to the victim domain's mail servers, while the legitimate email is handed over from other domains' MSA (Mail Submission Agent). The techniques of reverse DNS (rDNS) and Forward Confirmed Reverse DNS (FCrDNS) have been widely used to check whether the email is from dial-up users or dynamically assigned addresses, and mail servers will refuse email from such sources [29].

We propose an additional feature that is orthogonal to DNSBL or rDNS checking. Outgoing mail servers open specific ports to accept users' connections, while the bots are compromised hosts, where the well-known service ports are closed (require root privilege to open). When packets reach the mail server, the server issues an active probe sent to the source host to scan the following four ports that are commonly used for outgoing mail service: 25 (SMTP), 465 (SSL SMTP), 80 (HTTP) and 443 (HTTPS), which are associated with outgoing mail services. Because neither the current mail servers nor the McAfee's data

offer email senders' port information, we need to probe back sender's IP to check out what service ports might be open. The probe process was performed during both October 2008 and January 2009, well after the time when the email was received. Despite this delay, the status of open ports still exposes a striking difference between legitimate senders and spammers. Figure 6 shows the percentages and the numbers of opening ports for spam and ham categories respectively. The statistics are calculated on the senders' IPs from the evaluation dataset we used in Section 3.4 (October 22–28, 2007). In the spam case, 90% of spamming IP addresses have *none* of the standard mail service ports open; in contrast, half of the legitimate email comes from machines listening on at least one mail service port. Although firewalls might block the probing attempts (which causes the legitimate mail servers show no port listening), the status of the email-related ports still appears highly correlated with the distinction of the senders. When providing this feature as input to a classifier, we represent it as a bitmap (4 bits), where each bit indicates whether the sender IP is listening on a particular port.

3.3.2 Single-Header and Single-Message Features

In this section, we discuss other features that can be extracted from a single SMTP header or message: the number of recipients in the message, and the length of the message. We distinguish these features from those in the previous section, since extracting these features actually requires opening an SMTP connection, accepting the message, or both. Once a connection is accepted, and the SMTP header and subsequently, the compete message are received. At this point, a spam filter could extract additional non-content features.

3.3.2.1 Number of recipients: Spam tends to have more recipients

The features discussed so far can be extracted from a single IP packet from any given specific IP address combined with some historical knowledge of messages from other IPs. Another feature available without looking into the content is the number of address in "To" field of the header. This feature can be extracted after receiving the entire SMTP header but before accepting the message body. However, the majority of messages only have one address listed. Over 94% of spam and 96% of legitimate email is sent to a single recipient.

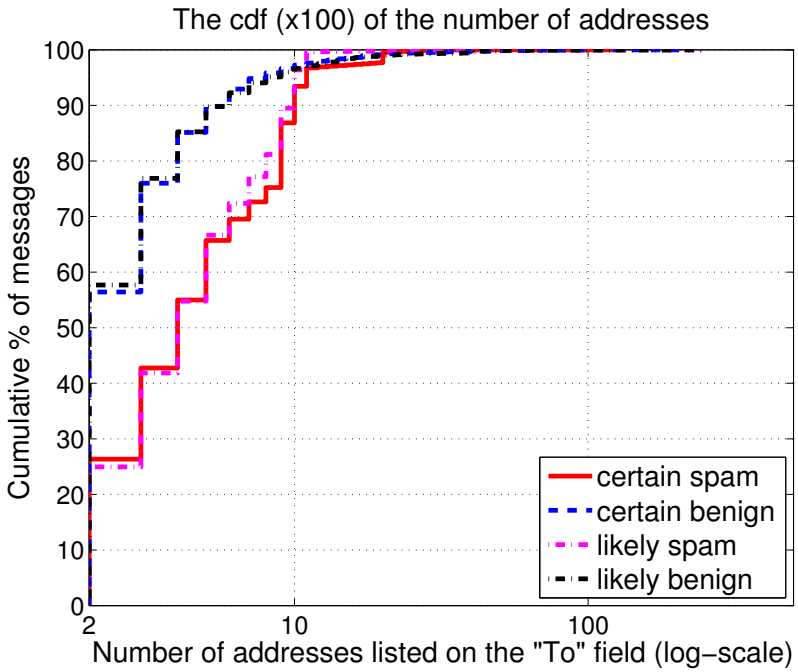


Figure 7: Distribution of number of addresses listed on the "To" field for each category (ignoring single-recipient messages).

Figure 7 shows the distribution of number of addresses in the "To" field for each category of messages for all emails that are sent to more than one recipient. The x-axis is on a log-scale to focus the plot on the smaller values. Based on this plot and looking at the actual values, it appears that if there are very large number of recipients on the "To" field (100 or more), there does not seem to be a significant difference between the different types of senders for this measure. The noticeable differences around 2 to 10 addresses show that, generally, ham has fewer recipients (close to 2) while spam is sent to multiple addresses (close to 10). (We acknowledge that this feature is probably evadable and discuss this in more detail in Section 3.6.1).

3.3.2.2 Message size: Legitimate mail has variable message size; spam tends to be small

Once an entire message has been received, the email body size in bytes is also known. Because a given spam sender will mostly send the same or similar content in all the messages, it can be expected that the variance in the size of messages sent by a spammer will be lower

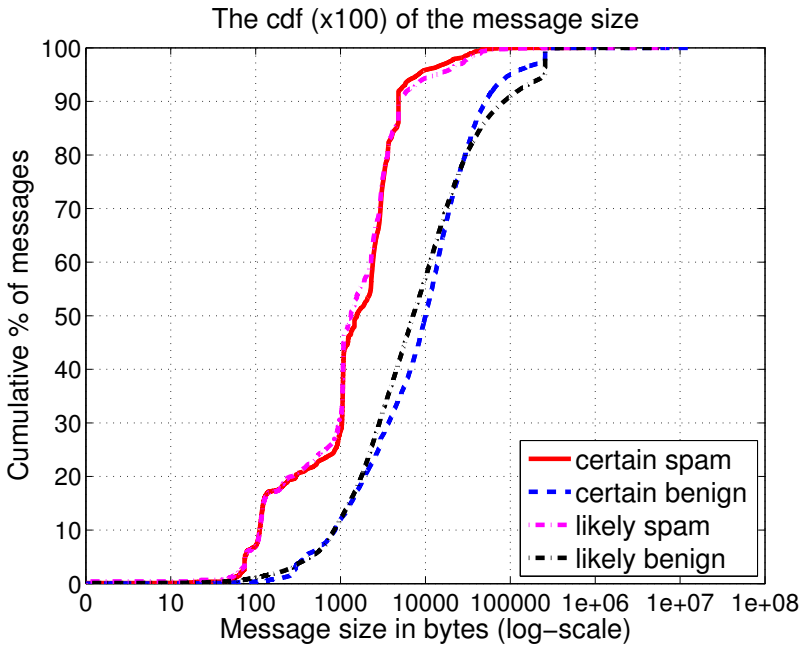


Figure 8: Distribution of message size (in bytes) for the different categories of messages.

than among the messages sent by a legitimate sender. To stay effective, the spam bots also need to keep the message size small so that they can maximize the number of messages they can send out. As such the spam messages can be expected to be biased towards the smaller size. Figure 8 shows the distribution of messages for each category. The spam messages are all clustered in the 1–10KB range, whereas the distribution of message size for legitimate senders is more evenly distributed. Thus, the message body size is another property of messages that may help differentiate spammers from legitimate senders.

3.3.3 Aggregate Features

The behavioral properties discussed so far can all be constructed using a single message (with auxiliary or neighborhood information). If some history from an IP is available, some *aggregate IP-level features* can also be constructed. Given information about multiple messages from a single IP address, the overall *distribution* of the following measures can be captured by using a combination of *mean and variance of*: (1) geodesic distance between the sender and recipient, (2) number of recipients in the "To" field of the SMTP header, and (3) message body length in bytes. By summarizing behavior over multiple messages

and over time, these aggregate features may yield a more reliable prediction. On the flip side, computing these features comes at the cost of increased latency as we need to collect a number of messages before we compute these. Sometimes gathering aggregate information even requires cross-domain collaboration. By averaging over multiple messages, these features may also smooth the structure of the feature space, making marginal cases more difficult to classify.

3.4 Evaluating the Reputation Engine

In this section, we evaluate the performance of *SNARE*'s *RuleFit* classification algorithm using different sets of features: those just from a single packet, those from a single header or message, and aggregate features.

3.4.1 Setup

For this evaluation, we used fourteen days of data from the traces, from October 22, 2007 to November 4, 2007, part of which are different from the analysis data in Section 3.3. In other words, the entire data trace is divided into two parts: the first half is used for measurement study, and the latter half is used to evaluate *SNARE*'s performance. The purpose of this setup is both to verify the hypothesis that the feature statistics we discovered would stick to the same distribution over time and to ensure that feature extraction would not interfere with our evaluation of prediction.

Training. We first collected the features for each message for a subset of the trace. We then randomly sampled 1 million messages from each day on average, where the volume ratio of spam to ham is the same as the original data (i.e., 5% ham and 95% spam; for now, we consider only messages in the "certain ham" and "certain spam" categories to obtain more accurate ground truth). Only our evaluation is based on this sampled dataset, *not* the feature analysis from Section 3.3, so the selection of those features should not have been affected by sampling. We then intentionally sampled equal amounts of spam as the ham data (30,000 messages in each categories for each day) to train the classifier because training requires that each class have an equal number of samples. In practice, spam volume is huge, and much spam might be discarded before entering the *SNARE* engine, so sampling

Table 2: *SNARE* performance using *RuleFit* on different sets of features using covariant shift. Detection and false positive rates are shown in bold. (The detection is fixed at 70% for comparison, in accordance with today's DNSBLs [23]).

(a) Single Packet

	Classified as	
	Spam	Ham
Spam	**70%**	30%
Ham	**0.44%**	99.56%

(b) Single Header/Message

	Classified as	
	Spam	Ham
Spam	**70%**	30%
Ham	**0.29%**	99.71%

(c) 24+ Hour History

	Classified as	
	Spam	Ham
Spam	**70%**	30%
Ham	**0.20%**	99.80%

on spam for training is reasonable.

Validation. We evaluated the classifier using temporal cross-validation, which is done by splitting the dataset into subsets along the time sequence, training on the subset of the data in a time window, testing using the next subset, and moving the time window forward. This process is repeated ten times (testing on October 26, 2007 to November 4, 2007), with each subset accounting for one-day data and the time window set as 3 days (which indicates that long-period history is not required). For each round, we compute the detection rate and false positive rate respectively, where the detection rate (the "true positive" rate) is the ratio of spotted spam to the whole spam corpus, and false positive rate reflects the proportion of misclassified ham to all ham instances. The final evaluation reflects the average computed over all trials.

Summary. Due to the high sampling rate that we used for this experiment, we repeated the above experiment for several trials to ensure that the results were consistent across trials. As the results in this section show, detection rates are approximately 70% and false positive rates are approximately 0.4%, even when the classifier is based only on single-packet features. The false positive drops to less 0.2% with the same 70% detection as the classifier incorporates additional features. Although this false positive rate is likely still too high for *SNARE* to subsume all other spam filtering techniques, we believe that the performance may be good enough to be used in conjunction with other methods, perhaps as an early-stage classifier, or as a substitute for conventional IP reputation systems (e.g., SpamHaus).

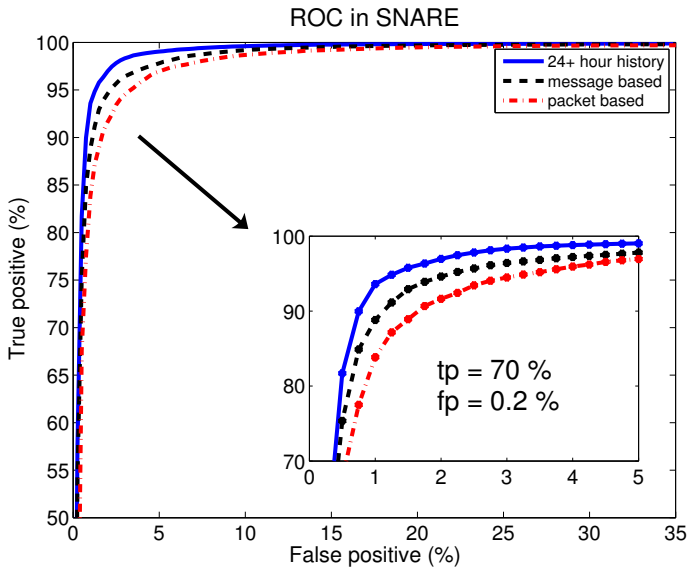


Figure 9: ROC in *SNARE*.

3.4.2 Accuracy of Reputation Engine

In this section, we evaluate *SNARE*'s accuracy on three different groups of features. Surprisingly, we find that, even relying on only single-packet features, *SNARE* can automatically distinguish spammers from legitimate senders. Adding additional features based on single-header or single-message, or aggregates of these features based on 24 hours of history, improves the accuracy further.

3.4.2.1 Single-Packet Features

When a mail server receives a new connection request, the server can provide *SNARE* with the IP addresses of the sender and the recipient and the time-stamp based on the TCP SYN packet alone. Recall from Section 3.3 even if *SNARE* has never seen this IP address before, it can still combine this information with recent history of behavior of other email servers and construct the following features: (1) geodesic distance between the sender and the recipient, (2) average distance to the 20 nearest neighbors of the sender in the log, (3) probability ratio of spam to ham at the time the connection is requested (4) AS number of the sender's IP, and (5) status of the email-service ports on the sender.

To evaluate the effectiveness of these features, we trained *RuleFit* on these features. The

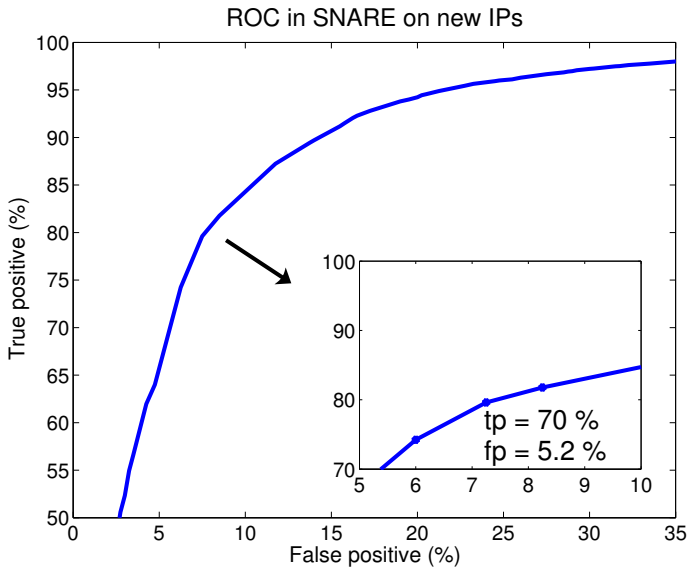


Figure 10: ROC on fresh IPs in *SNARE*.

dash-dot curve in Figure 9 demonstrate the ROC curve of *SNARE*'s reputation engine. The fp = 0.2% and tp = 70% statistics refer to the curve with 24-hour history (solid line), which will be addresses later. We check the false positive given a fixed true positive, 70%. The confusion matrix is shown in Table 2(a). Just over 0.44% of legitimate email gets labelled as spam. This result is significant because it relies on features constructed from a limited amount of data and just a single IP packet from the candidate IP. Sender reputation system will be deployed in conjunction with a combination of other techniques including content based filtering. As such, as a first line of defense, this system will be very effective in eliminating a lot of undesired senders. In fact, once a sender is identified as a spammer, the mail server does not even need to accept the connection request, saving network bandwidth and computational resources. The features we describe below improve accuracy further.

3.4.2.2 Single-Header and Single-Message Features

Single-packet features allow *SNARE* to rapidly identify and drop connections from spammers even before looking at the message header. Once a mail server has accepted the connection and examined the entire message, *SNARE* can determine sender reputation with increased confidence by looking at an additional set of features. As described in

Section 3.3.2, these features include the number of recipients and message body length. Table 2(b) shows the prediction accuracy when we combine the single-packet features (i.e., those from the previous section) with these additional features. As the results from Section 3.3 suggest, adding the *message body length* and *number of recipients* to the set of features further improves *SNARE*'s detection rate and false positive rate.

It is worth mentioning that the number of recipients listed on the "To" field is perhaps somewhat evadable: a sender could list the target email addresses on "Cc" and "Bcc" fields. Besides, if the spammers always place a single recipient address in the "To" field, this value will be the same as the large majority of legitimate messages. Because we did not have logs of additional fields in the SMTP header beyond the count of email addresses on the "To" field, we could not evaluate whether considering number of recipients listed under "Cc" and "Bcc" headers is worthwhile.

3.4.2.3 Aggregate Features

If multiple messages from a sender are available, the following features can be computed: the mean and variance of geodesic distances, message body lengths and number of recipients. We evaluate a classifier that is trained on *aggregate statistics* from the past 24 hours together with the features from previous sections.

Table 2(c) shows the performance of *RuleFit* with these aggregate features, and the ROC curve is plotted as the solid one in Figure 9. Applying the aggregate features decreases the error rate further: 70% of spam is identified correctly, while the false positive rate is merely 0.20%. The content-based filtering is very efficient to identify spam, but can not satisfy the requirement of processing a huge amount of messages for big mail servers. The prediction phase of *RuleFit* is faster, where the query is traversed from the root of the decision tree to a bottom label. Given the low false positive rate, *SNARE* would be a perfect first line of defense, where suspicious messages are dropped or re-routed to a farm for further analysis.

3.4.3 Other Considerations

Detection of "fresh" spammers. We examined data trace, extracted the IP addresses not showing up in the previous training window, and further investigated the detection

Table 3: Ranking of feature importance in *SNARE*.

rank	*Feature Description*
1	AS number of the sender's IP
2	average of message length in previous 24 hours
3	average distance to the 20 nearest IP neighbors of the sender in the log
4	standard deviation of message length in previous 24 hours
5	status of email-service ports on the sender
6	geodesic distance between the sender and the recipient
7	number of recipient
8	average geodesic distance in previous 24 hours
9	average recipient number in previous 24 hours
10	probability ratio of spam to ham when getting the message
11	standard deviation of recipient number in previous 24 hours
12	length of message body
13	standard deviation of geodesic distance in previous 24 hours

accuracy for those 'fresh' spammers with all *SNARE*'s features. If fixing the true positive as 70%, the false positive will increase to 5.2%, as shown in Figure 10. Compared with Figure 9, the decision on the new legitimate users becomes worse, but most of the new spammers can still be identified, which validates that *SNARE* is capable of *automatically* classifying "fresh" spammers.

Relative importance of individual features. We use the fact that *RuleFit* can evaluate the *relative importance* of the features we have examined in Sections 3.3. Table 3 ranks all spatio-temporal features (with the most important feature at top). The top three features—*AS num*, *avg length* and *neig density*—play an important role in separating out spammers from good senders. This result is quite promising, since most of these features are lightweight: Better yet, two of these three can be computed having received only a single packet from the sender. As we will discuss in Section 3.6, they are also relatively resistant to evasion.

Correlation analysis among features. We use mutual information to investigate how tightly the features are coupled, and to what extent they might contain redundant information. Given two random variables, mutual information measures how much uncertainty of one variable is reduced after knowing the other (i.e., the information they share). For

Table 4: Mutual information among features in *SNARE*; packet-based features are shown with numbers in dark circles. (The indices are the feature ranking in Table 3.)

	❶ (8.68)	2 (7.29)	❸ (2.42)	4 (6.92)	❺ (1.20)	❻ (10.5)	7 (0.46)	8 (9.29)	9 (2.98)	❿ (4.45)	11 (3.00)	12 (6.20)
2 (7.29)	**4.04**											
❸ (2.42)	**1.64**	1.18										
4 (6.92)	**3.87**	4.79	1.23									
❺ (1.20)	**0.65**	0.40	0.11	0.43								
❻ (10.5)	**5.20**	3.42	0.88	3.20	0.35							
7 (0.46)	**0.11**	0.08	0.02	0.08	0.004	0.15						
8 (9.29)	**5.27**	5.06	1.20	4.79	0.46	5.16	0.13					
9 (2.98)	**1.54**	1.95	0.53	2.03	0.09	1.17	0.10	2.08				
❿ (4.45)	**0.66**	0.46	0.07	0.49	0.02	0.87	0.006	0.85	0.13			
11 (3.00)	**1.87**	1.87	0.75	2.04	0.16	1.55	0.09	2.06	1.87	0.20		
12 (6.20)	**2.34**	2.53	0.49	2.12	0.20	2.34	0.07	2.30	0.52	0.31	0.73	
13 (8.89)	**4.84**	4.78	1.15	4.69	0.41	4.77	0.11	6.47	1.98	0.69	2.04	2.13

discrete variables, the mutual information of X and Y is calculated as following.

$$I(X,Y) = \sum_{x,y} p(x,y) \log(\frac{p(x,y)}{p(x)p(y)}) \tag{2}$$

When logarithm base-two is used, the quantity reflects how many bits can be removed to encode one variable given the other one. Table 4 shows the mutual information between pairs of features for one day of training data (October 23, 2007). We do not show statistics from other days, but features on those days reflect similar quantities for mutual information. The features with continuous values (e.g., geodesic distance between the sender and the recipient) are transformed into discrete variables by dividing the value range into 4,000 bins (which yields good discrete approximation); we calculate mutual information over the discrete probabilities. The indexes of the features in the table are the same as the ranks in Table 3; the packet-based features are marked with black circles. We also calculate the entropy of every feature and show them next to the indices in Table 4.

The interpretation of mutual information is consistent only within a single column or row, since comparison of mutual information without any common variable is meaningless. The table, of course, begs additional analysis but shows some interesting observations. The top-ranked feature, AS number, shares high mutual information (shown in bold) with several other features, especially with feature 6, geodesic distance between sender and recipient. The aggregate features of first-order statistics (e.g., feature 2, 4, 8) also have high values with each other. Because spammers may exhibit one or more of these features across

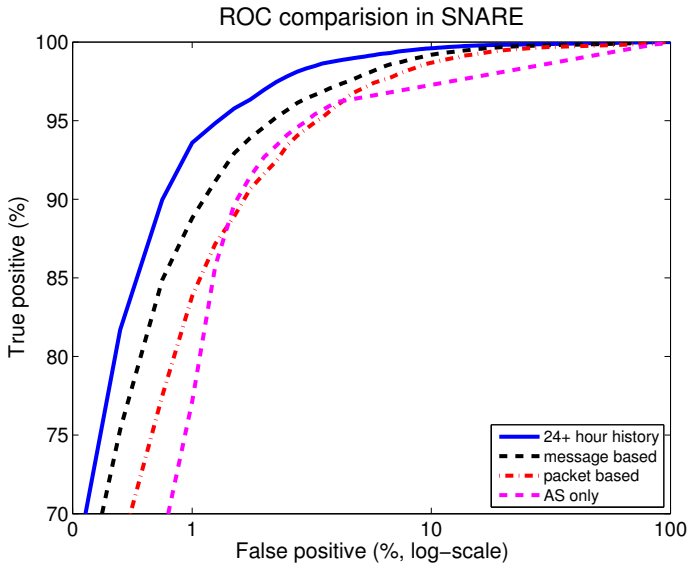


Figure 11: ROC comparison with AS-only case.

each message, aggregating the features across multiple message over time indicates that, observing a spammer over time will reveal many of these features, though not necessarily on any message or single group of message. For this reason, aggregate features are likely to share high mutual information with other features that are common to spammers.

One possible reason that aggregate features have high mutual information with each other is that aggregating the features across multiple messages over time incorporates history of an IP address that may exhibit many of these characteristics over time.

Performance based on AS number only. Since AS number is the most influential feature according to *RuleFit* and shares high mutual information with many other features, we investigated how well this feature alone can distinguish spammers from legitimate senders. We feed the AS feature into the predictive model and plot the ROC as the lower dashed curve in Figure 11. To make a close comparison, the "packet-based", "message-based", and "history-based" ROCs (the same as those in Figure 9) are shown as well, and the false positive is displayed on a log scale. The classifier gets false positive 0.76% under a 70% detection rate. Recall from Table 2 the false positive rate with "packet-based" features is almost a half, 0.44%, and that with "history-based" features will further reduce to 0.20%, which demonstrates that other features help to improve the performance. We also note that

using the AS number alone as a distinguishing feature may cause large amounts of legitimate email to be misclassified, and could be evaded if an spammer decides to announce routes with a forged origin AS (which is an easy attack to mount and a somewhat common occurrence) [56, 64, 129].

3.5 A Spam-Filtering System

This section describes how *SNARE*'s reputation engine could be integrated into an overall spam-filtering system that includes a whitelist and an opportunity to continually retrain the classifier on labeled data (e.g., from spam traps, user inboxes, etc.). Because *SNARE*'s reputation engine still has a non-zero false positive rate, we show how it might be incorporated with mechanisms that could help further improve its accuracy, and also prevent discarding legitimate mail even in the case of some false positives. We propose an overview of the system and evaluate the benefits of these two functions on overall system accuracy.

3.5.1 System Overview

Figure 12 shows the overall system framework. The system needs not reside on a single server. Large public email providers might run their own instance of *SNARE*, since they have plenty of email data and processing resources. Smaller mail servers might query a remote *SNARE* server. We envision that *SNARE* might be integrated into the workflow in the following way:

1. **Email arrival.** After getting the first packet, the mail server submits a query to the *SNARE* server (only the source and destination IP). Mail servers can choose to send more information to *SNARE* after getting the SMTP header or the whole message. Sending queries on a single packet or on a message is a tradeoff between detection accuracy and processing time for the email (i.e., sending the request early will make mail server get the response early). The statistics of messages in the received queries will be used to build up the *SNARE* classifier.

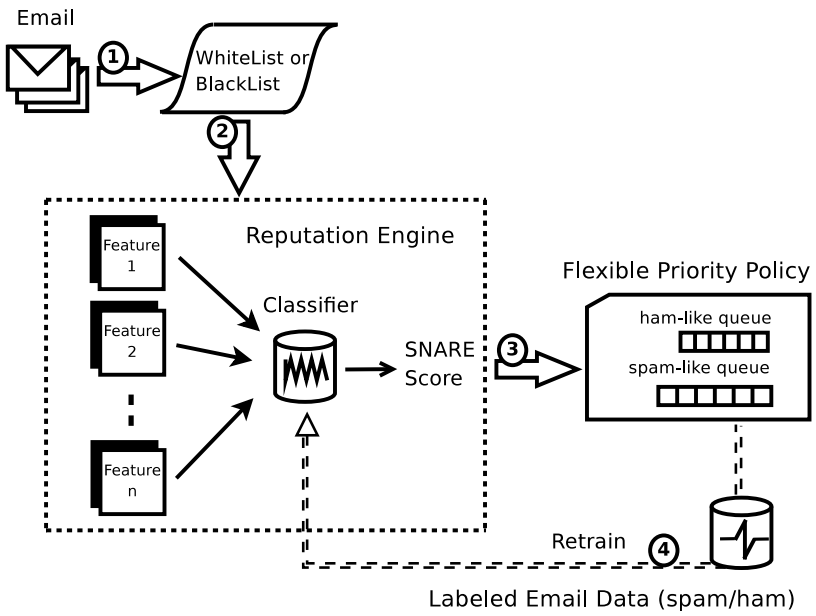


Figure 12: *SNARE* framework.

2. **Whitelisting.** The queries not listed in the whitelist will be passed to *SNARE*'s reputation engine (presented in Section 3.2.3) *before* any spam-filtering checks or content-based analysis. The output is a score, where, by default, positive value means likely spam and negative value means likely ham; and the absolute values represent the confidence of the classification. Administrators can set a different score threshold to make tradeoff between the false positive and the detection rate. We evaluate the benefits of whitelisting in Section 3.5.2.1.

3. **Greylisting and content-based detection.** Once the reputation engine calculates a score, the email will be delivered into different queues. More resource-sensitive and time-consuming detection methods (e.g., content-based detection) can be applied at this point. When the mail server has the capability to receive email, the messages in ham-like queue have higher priority to be processed, whereas the messages in spam-like queue will be offered less resources. This policy allows the server to speed up processing the messages that *SNARE* classifies as spam. The advantage of this hierarchical detecting scheme is that the legitimate email will be delivered to users' inbox sooner. Messages in the spam-like queue could be shunted to more resource-intensive

spam filters before they are ultimately dropped.[3]

4. **Retraining** Whether the IP address sends spam or legitimate mail in that connection is not known at the time of the request, but is known after mail is processed by the spam filter. *SNARE* depends on accurately labelled training data. The email will eventually receive more careful checks (shown as "Retrain" in Figure 12). The results from those filters are considered as ground truth and can be used as feedback to dynamically adjust the *SNARE* threshold. For example, when the mail server has spare resource or much email in the spam-like queue is considered as legitimate later, *SNARE* system will be asked to act more generous to score email as likely ham; on the other hand, if the mail server is overwhelmed or the ham-like queue has too many incorrect labels, *SNARE* will be less likely to put email into ham-like queue. Section 3.5.2.2 evaluates the benefits of retraining for different intervals.

3.5.2 Evaluation

In this section, we evaluate how the two additional functions (whitelisting and retraining) improve *SNARE*'s overall accuracy.

3.5.2.1 Benefits of Whitelisting

We believe that a whitelist can help reduce *SNARE*'s overall false positive rate. To evaluate the effects of such a whitelist, we examined the features associated with the false positives, and determine that, 43% of all of *SNARE*'s false positives for a single day originate from just 10 ASes. We examined this characteristic for different days and found that 30% to 40% of false positives from any given day originate from the top 10 ASes. Unfortunately, however, these top 10 ASes do not remain the same from day-to-day, so the whitelist may need to be retrained periodically. It may also be the case that other features besides AS number of the source provide an even better opportunity for whitelisting. We leave the details of refining the whitelist for future work.

[3] Although *SNARE*'s false positive rates are quite low, some operators may feel that any non-zero chance that legitimate mail or sender might be misclassified warrants at least a second-pass through a more rigorous filter.

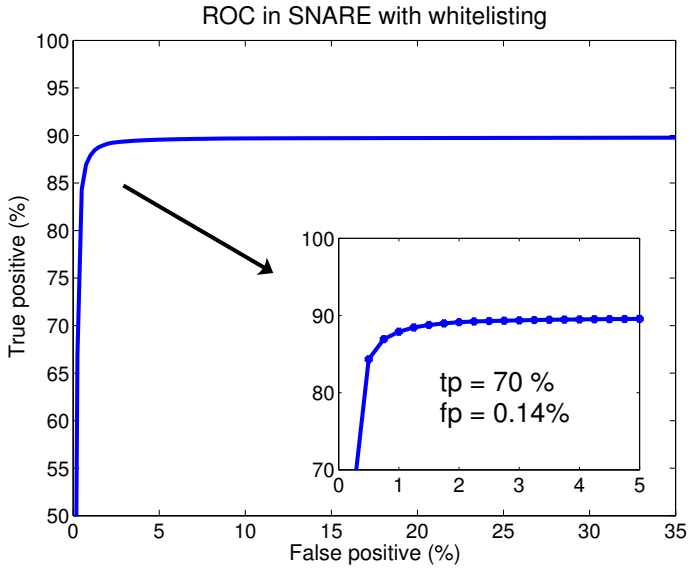


Figure 13: ROC in *SNARE* with whitelisting on ASes.

Figure 13 shows the average ROC curve when we whitelist the top 50 ASes responsible for most misclassified ham in each day. This whitelisting reduces the best possible detection rate considerably (effectively because about 11% of spam originates from those ASes). However, this whitelisting also reduces the false positive rate to about 0.14% for a 70% detection rate. More aggressive whitelisting, or whitelisting of other features, could result in even lower false positives.

3.5.2.2 Benefits of Retraining

Setup. Because email sender behavior is dynamic, training *SNARE* on data from an earlier time period may eventually grow stale. To examine the requirements for periodically retraining the classifier, we train *SNARE* based on the first 3 days' data (through October 23–25, 2007) and test on the following 10 days. As before, we use 1 million randomly sampled spam and ham messages to test the classifier for each day.

Results. Figure 14 shows the false positive and true positive on 3 future days, October 26, October 31, and November 4, 2007, respectively. The prediction on future days will become more inaccurate with time passage. For example, on November 4 (ten days after training), the false positive rate has dropped given the same true positive on the ROC

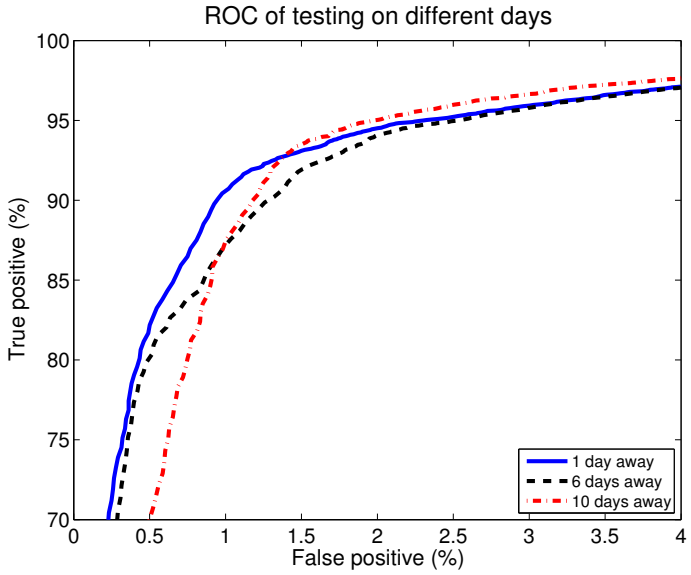


Figure 14: ROC using previous training rules to classify future messages.

curve. This result suggests that, for the spammer behavior in this trace, retraining *SNARE*'s classification algorithms daily should be sufficient to maintain accuracy. (We expect that the need to retrain may vary across different datasets.)

3.6 Discussion and Limitations

In this section, we address various aspects of *SNARE* that may present practical concerns. We first discuss the extent to which an attacker might be able to evade various features, as well as the extent to which these features might vary across time and datasets. We then discuss scalability concerns that a production deployment of *SNARE* may present, as well as various possible workarounds.

3.6.1 Evasion-Resistance and Robustness

In this section, we discuss the evasion resistance of the various network-level features that form the inputs to *SNARE*'s classification algorithm. Each of these features is, to some degree, evadable. Nevertheless, *SNARE* raises the bar by making it more difficult for spammers to evade detection without altering the techniques that they use to send spam. Although spammers might adapt to evade some of the features below, we believe that it will be difficult for a spammer to adjust all features to pass through *SNARE*, particularly

without somewhat reducing the effectiveness of the spamming botnet. We survey each of the features from Table 3 in turn.

AS number. AS numbers are more persistently associated with a sender's identity than the IP address, for two reasons: (1) The spamming mail server might be set up within specific ASes without the network administrator shutting it down. (2) Bots tend to aggregate within ASes, since the machines in the same ASes are likely to have the same vulnerability. It is not easy for spammers to move mail servers or the bot armies to a different AS; therefore, AS numbers are robust to indicate malicious hosts.

Message length. In our analysis, we discovered that the size of legitimate email messages tends to be much more variable than that of spam (perhaps because spammers often use templates to sent out large quantities of mail [15]). With knowledge of this feature, a spammer might start to randomize the lengths of their email messages; this attack would not be difficult to mount, but it might restrict the types of messages that a spammer could send or make it slightly more difficult to coordinate a massive spam campaign with similar messages.

Nearest neighbor distances. Nearest neighbor distance is another feature that will be hard to modify. Distances to k nearest neighbors effectively isolate existence of unusually large number of email servers within a small sequence of IP addresses. If the spammers try to alter their neighborhood density, they will not be able to use too many machines within a compromised subnet to send spam to the same set of destinations. Although it is possible for a botnet controller to direct bots on the same subnet to target different sets of destinations, such evasion does require more coordination and, in some cases, may restrict the agility that each spamming bot has in selecting its target destinations.

Status of email service ports. Some limitation might fail the active probes, e.g., the outgoing mail servers use own protocol to mitigate messages (such as Google mail) or a firewall blocks the connections from out of the domain. But the bots do not open such ports with high probability, and the attackers need to get root privilege to enable those ports (which requires more sophisticated methods and resources). The basic idea is to find

out whether the sender is a legitimate mail server. Although we used active probes in *SNARE*, other methods could facilitate the test, such as domain name checking or mail server authentication.

Sender-receiver geodesic distance. The distribution of geodesic distances between the spammers' physical location and their target IP's location is a result of the spammers' requirement to reach as many target mail boxes as possible and in the shortest possible time. Even in a large, geographically distributed botnet, requiring each bot to bias recipient domains to evade this feature may limit the flexibility of how the botnet is used to send spam. Although this feature can also be evaded by tuning the recipient domains for each bot, if bots only sent spam to nearby recipients, the flexibility of the botnet is also somewhat restricted: it would be impossible, for example, to mount a coordinate spam campaign against a particular region from a fully distributed spamming botnet.

Number of recipients. We found that spam messages tend to have more recipients than legitimate messages; a spammer could likely evade this feature by reducing the number of recipients on each message, but this might make sending the messages less efficient, and it might alter the sender behavior in other ways that might make a spammer more conspicuous (e.g., forcing the spammer to open up more connections).

Time of day. This feature may be less resistant to evasion than others. Having said that, spamming botnets' diurnal pattern results from when the infected machines are switched on. For botnets to modify their diurnal message volumes over the day to match the legitimate message patterns, they will have to lower their spam volume in the evenings, especially between 3:00 p.m. and 9:00 p.m. and also reduce email volumes in the afternoon. This will again reduce the ability of botnets to send large amounts of email.

3.6.2 Other Limitations

We briefly discuss other current limitations of *SNARE*, including its ability to scale to a large number of recipients and its ability to classify IP addresses that send both spam and legitimate mail.

Scale. *SNARE* must ultimately scale to thousands of domains and process hundreds of

millions of email addresses per day. Unfortunately, even state-of-the-art machine learning algorithms are not well equipped to process datasets this large; additionally, sending data to a central coordinator for training could potentially consume considerably bandwidth. Although our evaluation suggests that *SNARE*'s classification is relatively robust to sampling of training data, we intend to study further the best ways to sample the training data, or perhaps even perform in-network classification.

Dual-purpose IP addresses. Our conversations with large mail providers suggest that one of the biggest emerging threats are "web bots" that send spam from Web-based email accounts.As these types of attacks develop, an increasing fraction of spam may be sent from IP addresses that also send significant amounts of legitimate mail. These cases, where an IP address is neither good nor bad, will need more sophisticated classifiers and features, perhaps involving timeseries-based features.

3.7 Summary

Although there has been much progress in content-based spam filtering, state-of-the-art systems for *sender reputation* (e.g., DNSBLs) are relatively unresponsive, incomplete, and coarse-grained. Towards improving this state of affairs, this chapter has presented *SNARE*, a sender reputation system that can accurately and automatically classify email senders based on features that can be determined early in a sender's history—sometimes after seeing only a single IP packet. The *SNARE* system represents a substantial progress towards building a practical and effective sender reputation system using limited network-level metadata about the messages. Some of the network-level features that we identified have since been incorporated into spam filtering products at Yahoo! and McAfee.

CHAPTER IV

MONITORING THE INITIAL DNS LOOKUPS AND HOSTING OF SPAMMER DOMAINS

4.1 Introduction

Miscreants usually depend on victims to click on embedded URLs in spam messages and direct them to Web sites that host scams, malware, and other malicious content. To mitigate these threats, network operators try to derive a reputation for each domain that reflects the likelihood that the domain is associated with a particular type of attack (e.g., scam, phishing, malware hosting). The rate at which new domains appear makes quickly developing a reputation for these domains particularly challenging: in our analysis, we find that over tens of thousands of new domains are registered every day. Existing DNS reputation systems use the characteristics of DNS lookups from resolvers that look up a domain to distinguish legitimate from malicious domains [3,4]. Unfortunately, these systems must observe a significant volume of DNS lookups before determining the reputation for a domain, which only occurs *after* compromise has taken place.

Towards facilitating pre-attack detection of malicious domains, we study the initial DNS activity for each domain and characterize how the observable behavior for a malicious domain differs from that of legitimate domains. We study two aspects of initial DNS behavior associated with domains: (1) the *DNS infrastructure* used to resolve the domains to IP addresses; and (2) the *DNS lookup patterns* from the networks that perform initial lookups to the domain. Certain characteristics of the DNS infrastructure may be unique to malicious domains, such as the IP address ranges and ASes that host either the authoritative name servers for the sites, or the sites themselves. Identifying infrastructure that is common across malicious domains may provide hints for identifying malicious domains before the attacks themselves are mounted. Characteristics of early DNS lookups can help network operators discover valuable information about the nature of the domains that are being

looked up. Notably, we find that domains that are registered for malicious purposes are initially queried from a much more diverse set of subnets than legitimate domains.

Our study of DNS behavior early in a domain's life cycle is motivated by our ultimate desire to perform early detection of malicious domains. We use domains collected at several large spam traps as a source of domains associated with spam campaigns. To characterize the resource record behavior of each domain, we perform periodic iterative queries of newly registered domains in March 2011. To characterize DNS lookup patterns across networks, we use information about DNS lookups collected from the Verisign top-level domain servers, coupled with registration information about these domains.

We focus exclusively on the early DNS behavior of a domain, which is enabled by two important pieces of information. First, registration records alert us when a domain is registered, and allow us to begin querying it immediately, before attacks. Second, we study a global view of early DNS lookup patterns across the entire Internet for `.com` and `.net` domains. Our study reveals the following findings:

- *Domain registration and resource record establishment happens before attacks take place.* As many as 55% of spam campaigns may occur at least one day after the domain referenced in the spam messages were registered, offering the potential for early discovery of malicious domains based on initial DNS behavior.
- *DNS infrastructure for malicious domains is located in different address space regions and autonomous systems than the infrastructure for legitimate domains.* A few autonomous systems and IP address regions host infrastructure only for domains that are associated with malicious activity. Identifying these at domain registration time can potentially enable early detection.
- *Early lookup patterns for a newly registered malicious domains differ significantly from the patterns for a legitimate domain.* Domains associated with spam campaigns are initially looked up by a more diverse set of network address regions than legitimate domains. Especially, the newly registered spam domains become "popular" more quickly.

These features may ultimately be used to develop unique fingerprints for distinguishing legitimate domains from those that are associated with Internet attacks.

The rest of this chapter is organized as follows. In Section 4.2 we present the problem context and the data sets used for our analysis. Section 4.3 studies the characteristics of resource records for newly registered legitimate and malicious domains. Section 4.4 studies the lookup characteristics for different types of domains. We conclude our work in Section 4.5.

4.2 Context and Data Collection

We provide a brief overview of DNS records and lookups, as well as a description of the data used in our study.

4.2.1 DNS Resource Records and Lookups

When an entity registers a DNS domain, domain name registries insert several basic entries into the zone files to refer to the services for the domain. NS records point to the authoritative name servers for the zone, MX records point to the domain's mail servers, and A records point to the hosts. The NS and MX records can be further resolved to IP addresses. A single domain is typically assigned multiple server records for redundancy, but the number of IP addresses associated with the records is typically much less than the number of the domains being registered.

In March 2011, three million second-level domains under `.com` and `.net` were newly registered with NS records, but the number of distinct IPs mapped from NS records was just 150 thousand; A records and MX records have similar statistics. This observation indicates that the same server has been repeatedly used by many different domains to host DNS infrastructure.

Recursive DNS servers relay the users' queries to the zone's authoritative servers to acquire resource records, which reduces DNS traffic in the wide area. Recursive servers commonly respond to the hosts' requests within their respective networks, so the set of recursive servers querying for a domain can be a reasonable approximation for the networks that have attempted to reach the domain. The top-level domain (TLD) name servers thus

Table 5: DNZA format examples.

type	*example*
DNZA entry	add-new example.com NS ns1.example.com
Query record	example.com 111.111.111.0 , 22.22.22.0

provide a natural vantage point for monitoring the lookups directed to the second-level domains (the direct sub-domains below a TLD). Although DNS caching prevents us from determining the volume of lookups to a domain, the distribution of the recursive servers contains rich information about which networks have issued lookups for a domain; this statistic is particularly useful during the early part of a domain's life cycle, when it is initially registered and no caching has yet taken place.

4.2.2 Data Collection

We describe our data and the process of probing for resource records and correlating with spam messages.

DNS data. The top-level domain servers are responsible for maintaining the zone information (more specific, second-level domains) and answer the queries for the registered domains. Verisign, Inc. operates the generic top-level domains (gTLDs) for `.com` and `.net`, which account for over 45% of registered domain names on the Internet [120].The servers maintain two kinds of dynamics about the second-level domains. The first type of information is the *Domain Name Zone Alert (DNZA)*. This information includes changes about the zone, such as whether a domain name was newly registered or a name server's IP address was modified. The DNZA files keep track of these changes.

The second type of information concerns the *DNS queries* issued by the recursive servers. After the recursive servers sent queries to the TLD name servers for resolving the second-level domains names, Verisign's systems aggregated the source IP addresses into /24 subnets for logging and the TLD name servers recorded the querying subnets each day. The query records show the relationship between the domain names and the queriers. Verisign deploys multiple TLD name servers to resolve second-level domain names, and we collected the logs of querying /24s from 33 of those servers for analysis. Table 5 shows the example format

of each type of data. The DNZA entry indicates that an "add-new" command created a new domain `example.com` and the NS record was `ns1.example.com`; The query record means that there were queries from /24s of "111.111.111.0" and "22.22.22.0" for the domain. The DNZA files and the query data were collected at Verisign's `.com` and `.net` TLD name servers during the period of March 2011. On average, about 80 million domains were queried each day.

Resource records. The DNZA entries with "add-new" commands show what domains are newly registered. To get the new zone's resource records, we must perform active queries to their authority servers, since the second-level domains' records are not available within the TLD name servers. After a second-level domain under `.com` or `.net` is registered, we probe the domain once a day to discover the resource records and the resolved IPs. As mentioned in Section 4.2.1, we collect NS, MX and A records. We performed the probing procedure during March 2011. For example, 190 thousand domains were created on March 1, 2011; we repeatedly queried those domains once a day over the next 30 days. At the end of March 2011, we accumulated 4 million domains for monitoring. We use the PlanetLab platform [89] to make it feasible to query a large set of domains. Each PlanetLab node is responsible to query a subset of domains, and deliver the collected information back to the central monitor. Eventually, we deployed around 150 PlanetLab nodes to perform the probing procedure throughout the month. Though the daily querying does not capture all the changes in the resource records, it continually tracks the "snapshots" of DNS infrastructure and implies the change trends.

Spamming. Scam domains appearing in spam messages are the major targets in our study, since timestamps in each email help explicitly identify when the spamming activities occurred, and spam is related to many different attacks, such as phishing. We used a spam trap to capture emails sent from spammers during March 2011. Because the domains for the spam trap have no legitimate email addresses, emails received at the mail server were all spam. The second-level domains appearing in the messages' URLs were extracted as being involved in spamming activities (overall, 40% of unique second-level domains were found under `.com` and `.net`). In the context of this chapter, we use "scam domains"

and "malicious domains" interchangeably to refer to the second-level domains identified being associated with spam. From spam traps, we identified 2,045 scam domains as newly registered during March 2011. We also checked the domains with Spamhaus [110], and identified 4,587 blacklisted second-level domains. The union of these two sets yielded a total of 5,988 `.com` and `.net` second-level domains that we considered spamming-related.

To obtain a representative set of legitimate domains for comparison, we sampled 6,000 domains registered during March 2011 that have not yet appeared in any blacklist.

4.3 Registration & Resource Records

We first check the time between the registration of a domain and the subsequent attack to investigate the potential for early detection. Then, we explore how DNS behavior associated with infrastructure—where a domain's resolvers initially reside—can be an early signal for malicious domains.

4.3.1 Time Between Registration and Attack

We hypothesize that there may be some time between when spammers register new domains and when they send spam. We examine the extent of the delay between the time when a domain is initially registered and when it is ultimately used in an attack. If such a delay exists, it might allow blacklist operators to list the malicious domains, possibly before the spam campaign occurs.

How much time occurs between the domain registration and attack? Figure 15 shows the distribution between the time when we start to observe records about the malicious domains registered in March 2011, and the earliest time when the domains appeared in the spam messages. We take the timestamps in our spam traps, as well as emails received at the Yahoo! mail servers. Yahoo! Inc. provides the received time of all email messages and the URLs contained in the messages. The Yahoo! email gives a broader coverage of monitoring around the world. We take the earliest time points when seeing a "bad" domain in email messages (either in Yahoo! data or in spam trap) as the estimated start of the spamming attack about that domain. The x-axis represents the delay between when a domain was registered and when we first witnessed the domain associated with a spam

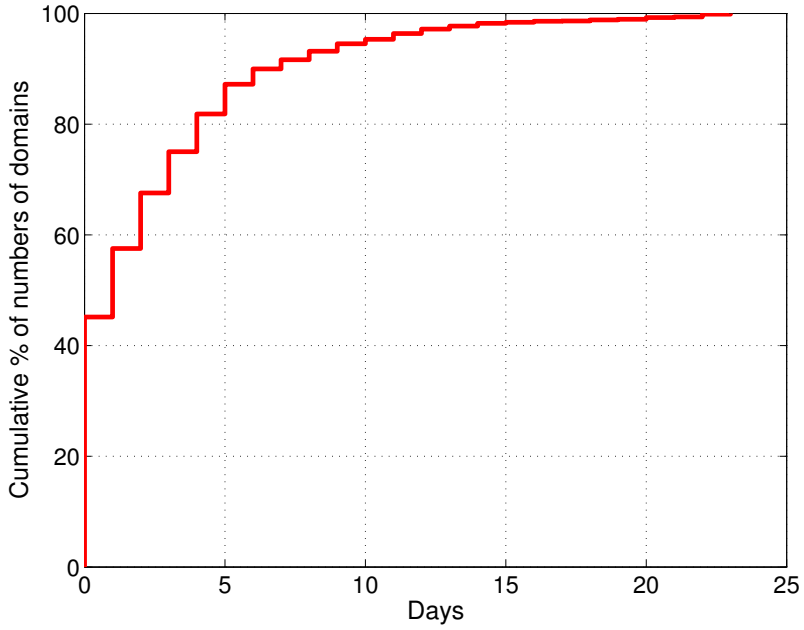


Figure 15: Days between a malicious domain's registration (in March 2011) and the time when the domain showed up in spam.

campaign, and the y-axis is the percentage of the malicious domains registered in March 2011.

Finding 4.3.1 (Delay until attack) *More than 55% of the malicious domains appeared in spam campaigns more than one day after they were registered.*

We define the first five days after domain registration as *pre-attack period*. About 20% of domains might not be used in attacks during this period, and the time windows for other domains being explored in spamming are also limited. In the rest of the chapter, we will analyze the characteristics of DNS infrastructure for malicious domains both throughout their lifetime (i.e., after the domains' registration) and within the pre-attack period. In Section 4.4, we further investigate the lookup behavior during the early stage.

4.3.2 Location of DNS Infrastructure

When determining the IP address that maps to each DNS record, we find that the assigned records for spam domains across IP space are far from uniform.

How is the DNS infrastructure that hosts a domain initially distributed across

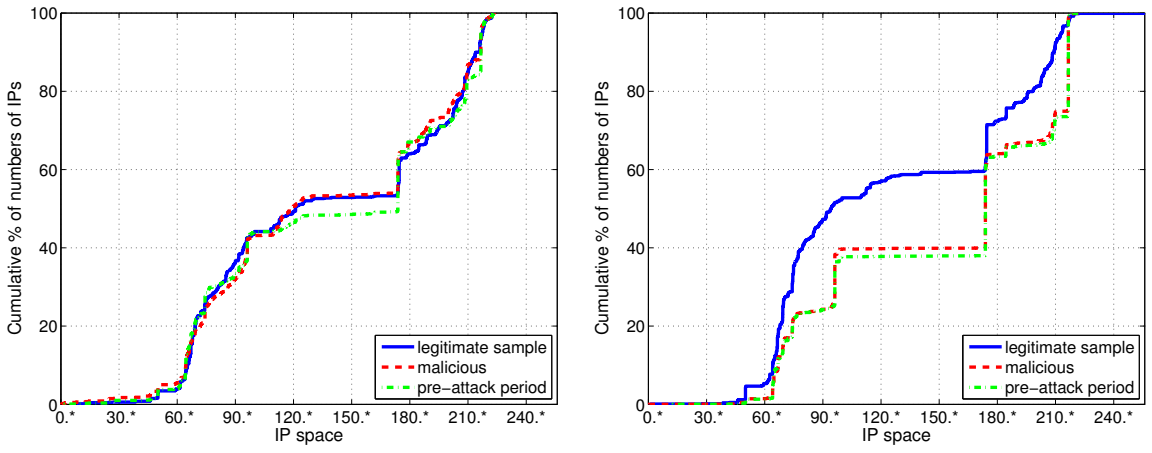
IP address space? The initial distribution of domain records across IP address space may provide clues as to a domain's reputation. Figure 16 shows how the IPs associated with NS, MX, and A records from malicious and legitimate domains are distributed across IP address space. The x-axis represents IPv4 space. If an IP maps to multiple records from different domains, we count it only once in the figure. The y-axis indicates the percentage of addresses less than or equal to the IP value on the x-axis. The solid blue curves plot the distribution of legitimate sample domains, the red dashed curves show the outcome of malicious domains, and the green dash-dot curves represent observed records for the malicious domains during pre-attack period. Interestingly, we observe that the DNS records associated with malicious domains are distributed differently than the records associated with legitimate ones.

Finding 4.3.2 (Distribution across IP address space) *The IP addresses used by malicious domains in the NS, MX and A records are distributed densely in a small fraction of IP address space.*

The IP addresses associated with DNS resource records are not distributed evenly across the IP address space. Some network range has more IPs pointed from NS, MX or A records; while the record IPs in other fraction of address space are distributed sparsely. Particularly two network blocks carried records from malicious domains, 96.45.0.0/16 and 216.162.0.0/16. The prefix 173.213.0.0/16 has many IPs in spamming domains, but the same range hosts legitimate domains, too. This observation indicates that if IPs corresponding to different domains' records reside close to each other in a network block, those domains may appear in spam in the future.

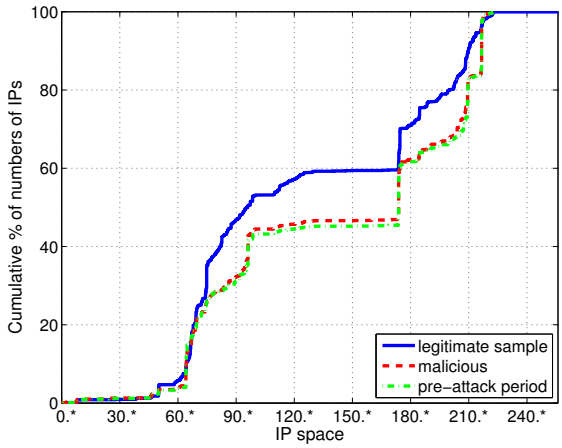
Is the DNS infrastructure for malicious domains located in particular ASes? Of course, the IP addresses of the records are not sufficient to confirm that a domain is scam-related. We examined the distribution of the resource records across ASes and compared the distribution of legitimate and malicious domains. Table 6 shows the top three ASes ranked by the percentage of domains ever having records being resolved into ASes.

Finding 4.3.3 (Distribution across ASes) *More than* 30% *of the malicious domains have at least one record resolving to one or two particular ASes, which are different from*



(a) NS records

(b) MX records



(c) A records

Figure 16: Fraction of IP addresses associated with malicious domains and comparison with legitimate domains.

those ASes mostly used by legitimate domains.

We observe that many of the new legitimate domains have larger registrars like `GoDaddy` operate their DNS, and host their service infrastructure with well-known provider, like `Google`. On the other hand, spamming domains' records are scattered across multiple ASes and countries. Spammers appear to prefer certain specific ASes to host their DNS infrastructure.

Are there "bad" ASes that host DNS infrastructure exclusively for malicious domains? We define an AS as "tainted" once the number of malicious domains whose

Table 6: Top three ASes containing domains' records.

(a) Legitimate domains

Type	*AS*	*domain ratio*	*AS Name*	*Country*
	8560	15.9%	1&1 Internet AG	Germany
	26496	10.9%	GoDaddy.com, Inc.	U.S.
NS	4134	10.1%	Chinanet Backbone	China
	26496	30.5%	GoDaddy.com, Inc.	U.S.
	15169	7.3%	Google Inc.	U.S.
MX	21844	7.0%	ThePlanet.com	U.S.
	26496	31.8%	GoDaddy.com, Inc.	U.S.
	8560	4.3%	1&1 Internet AG	Germany
A	21844	4.1%	ThePlanet.com	U.S.

(b) Malicious domains

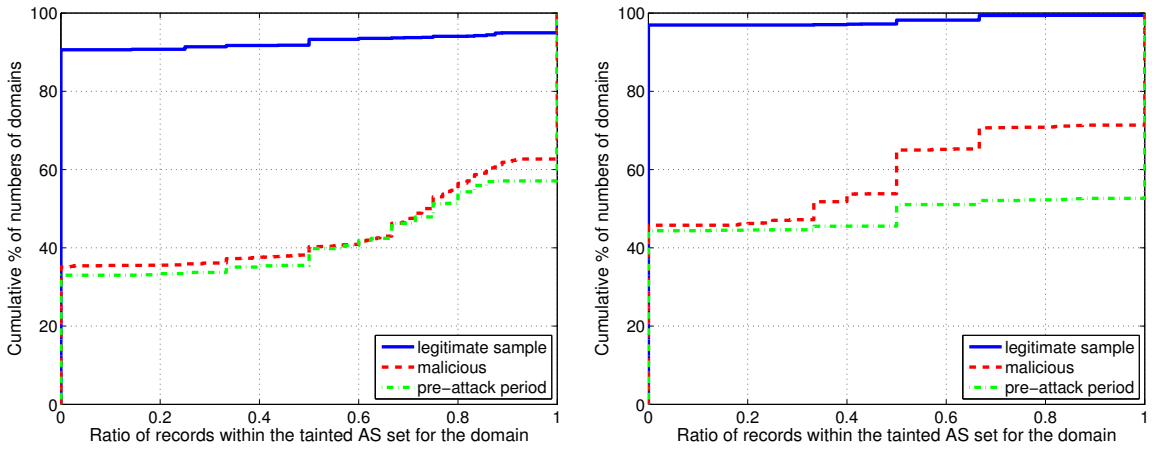
Type	*AS*	*domain ratio*	*AS Name*	*Country*
	4134	33.6%	Chinanet Backbone	China
	28753	17.0%	Leaseweb De	Germany
NS	31365	16.3%	SGSTelekom	Turkey
	197088	23.9%	Colohost LLC	Latvija
	3292	19.3%	TDC Data Networks	U.S.
MX	5632	12.3%	3dgwebhosting.com Inc	U.S.
	4134	19.3%	Chinanet Backbone	China
	197088	14.3%	Colohost LLC	Latvia
A	30890	13.8%	Evolva Telecom	Romania

(c) Malicious domains in pre-attack period

Type	*AS*	*domain ratio*	*AS Name*	*Country*
	4134	37.8%	Chinanet Backbone	China
	28753	20.4%	Leaseweb De	Germany
NS	27699	11.3%	Tel. De Sao Paulo S.A.	Brazil
	197088	14.3%	Colohost LLC	Latvija
	3292	21.8%	TDC Data Networks	U.S.
MX	5632	12.3%	3dgwebhosting.com Inc	U.S.
	4134	19.7%	Chinanet Backbone	China
	197088	15.0%	Colohost LLC	Latvia
A	28753	11.6%	Leaseweb De	Germany

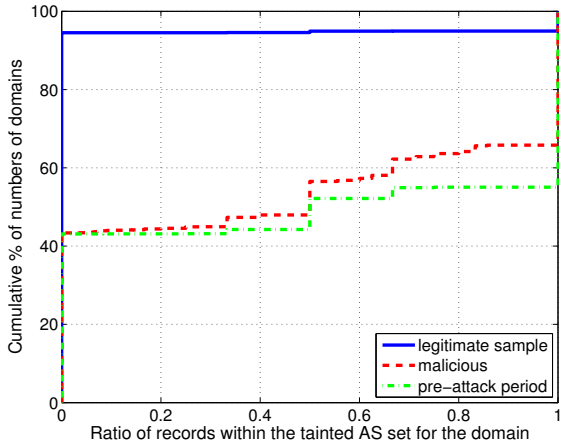
DNS records are resolved within the AS exceeds a threshold. The set of tainted ASes represent the networks that attackers most heavily use, as indicated by the malicious domains' registration. After a domain's registration, attackers create DNS entries for the domain, and the records resolve to different IP addresses. We then check whether the resulting IPs belong to the tainted ASes. If a domain accumulates many records that resolve to tainted ASes, we suspect that the domain is related to the observed attacks.

Finding 4.3.4 (Domains hosted by "bad" ASes) *Most legitimate domains have A, MX, and NS records that are hosted almost entirely in untainted ASes. On the other hand, the majority of spam domains have records hosted in tainted ASes, even during the pre-attack period.*



(a) NS records

(b) MX records



(c) A records

Figure 17: Distribution for the ratio of domains' records falling in the "tainted" AS set.

We derive the tainted AS set by including an AS that has hosted records for more than 100 spam domains. Figure 17 shows the ratio of the tainted record number to the number of all records for the domain. More than 90% of legitimate new domains have zero records belonging to the tainted AS set.

4.4 Early Lookup Behavior

The recursive DNS resolvers initially query the TLD name servers to get referrals to second-level domains. In this section, we explore the characteristics in the lookup networks to different types of domains. Queries to a malicious domain may signal the onset of attack, and the abnormal pattern in the global DNS traffic could help to detect the attack campaign

Table 7: Five largest clusters based on lookup networks.

total	*malicious*	*legitimate*	*% malicious*
1404	463	941	33.0%
157	156	1	99.4%
16	16	0	100.0%
10	10	0	100.0%
10	10	0	100.0%

in its infancy.

4.4.1 Network-Wide Patterns

We first investigate the querying patterns across different domains, to see whether similar sets of networks were looking up different domains. Our intuition is that domains that are used for malicious purposes may be looked up by similar groups of networks as well. For example, a user clicking on a URL in spam might click on other spam URLs. If two domains are queried by the same set of recursive DNS servers, they may be the same type of domain.

Are the networks querying different domains distributed similarly? We measure the similarity using an average pairwise similarity of querying /24 network blocks over n days. Suppose two domains A and B who have sequences of querying /24 set $\{a_1, a_2, \ldots, a_n\}$ and $\{b_1, b_2, \ldots, b_n\}$ over n days. The similarity between domains A and B is

$$S(A, B) = \frac{\sum_{i=1}^{n} J(a_i, b_i)}{n}, \text{ where } J(a_i, b_i) = \frac{|a_i \cap b_i|}{|a_i \cup b_i|}$$

where $J(a_i, b_i)$ is the Jaccard index of set a_i and b_i: the size of the set intersection divided by the size of union. Based on this pairwise similarity, we aggregate the domains into different groups using single-linkage clustering, a simple and efficient clustering method [130]. We considered a 5-day time period from March 1–5, 2011, during which there were 804 malicious domains and $1,104$ sampled legitimate domains registered. We terminate the clustering after $50,000$ comparisons, which places $1,631$ domains into 17 clusters that have more than a single domain. We expect domains to fall into distinct clusters. Tables 7 shows the statistics for the five largest clusters. The first three columns show the domain counts in each cluster. The last column means the percentage of the malicious domains in the cluster.

Finding 4.4.1 (Similarity in lookups) *Different malicious domains are looked up by similar group of network blocks, which may indicate that they are part of the same spamming campaign.*

The results show that clustering often works well: many of the clusters contain either only all good or all bad domains. The legitimate domains contained in the large cluster are only queried by a small number of networks, which a detection system could easily filter. These results suggest that domains of certain types do share similar network-wide spatial lookup patterns that may ultimately be used as input to a blacklist.

4.4.2 Evolution of Lookup Traffic

The numbers of distinct networks querying the TLD servers for the second-level domains approximate how widely around the world the users try to connect to these domains. Although there might be multiple connection attempts behind one recursive server, counting all querying recursive servers is a good indicator for the domain's initial "popularity". Our intuition is that once deployed, malicious domains may receive a lot of traffic in a short time, but visits to legitimate domains will increase relatively more slowly.

How quickly do the newly registered domains become popular? Figure 18 shows the average lookup volume from /24s for domains in different categories over time. The x-axis shows the number of days after a domains' registration. The y-axis shows the average number of querying /24s over the domains with error bars (i.e., standard error). The solid blue curve shows lookups for legitimate domains; the y-axis values are multiplied by 10 to make the figure more readable. The dashed red curve shows the patterns of malicious domains.

Finding 4.4.2 (Initial lookup trends) *Queries to the malicious domains increased quickly after the domains were registered, and usually reached the peak in the first 3–4 days.*

On the other hand, /24s querying for domains not reported as malicious increased slowly and stayed relatively low over the 30-day period. The markedly different lookup patterns of likely legitimate domains and those involved in spamming activities might ultimately help

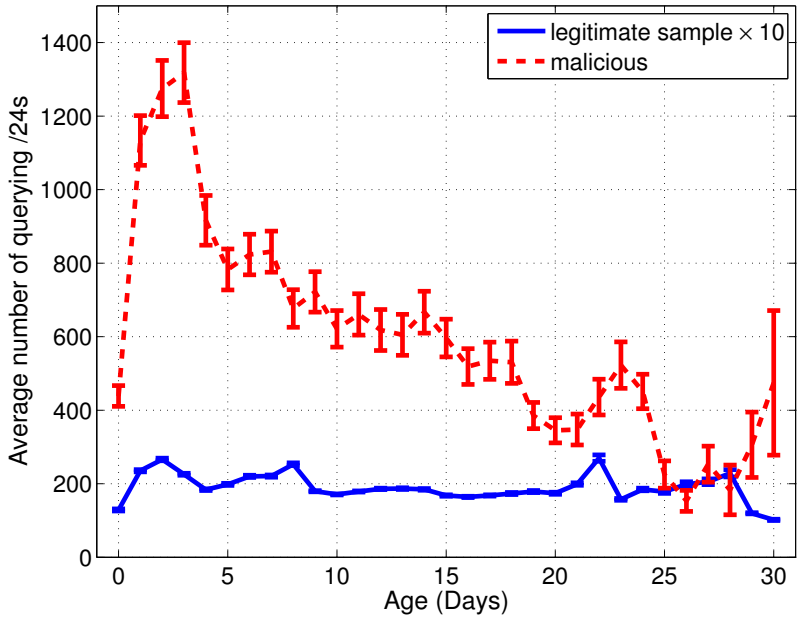


Figure 18: Number of querying /24s after domains' registration.

blacklist operators quickly detect bad domains, by watching for newly registered domains that suddenly become popular. The changes to malicious domains also indicate that the initial five-day period may contain valuable information, since these attack domains are heavily queried at the beginning, but lookups quickly trail off after that.

4.5 Summary

We have monitored DNS resource records for second-level domains newly registered in March 2011 and examined the lookup traffic to large authoritative top-level domain servers. We show the DNS characteristics observed at TLD name servers and extracted from zones' resource records for malicious domains are different than those for legitimate domains. Resource records of malicious domains tend to resolve to specific IP address range and ASes. Once we identify a set of "tainted" autonomous systems that host many scam domains, the legitimate domains rarely have resource records within the tainted AS set. We also discover that miscreant domains exhibit distinct clusters, in terms of the networks that look up these domains. Finally, we find that these domains become widely popular considerably more quickly after their initial registration time.

The distinct DNS characteristics and their tendency on different types of domains suggest that it may ultimately be possible to fingerprint domains based on their resource records and lookup traffic close to TLD name servers before an attack ever takes place. Although a single pattern in DNS might have limited power to identify malicious domains, the combination of our findings can guide the design of future "early warning" systems for DNS. This work has produced a patent with Verisign [85].

CHAPTER V

UNDERSTANDING THE DOMAIN REGISTRATION BEHAVIOR OF SPAMMERS

5.1 Introduction

Ideally, the decisions of blacklisting spammer domains could be made at *registration time* rather than at *usage time*, enabling "proactive blocking". However, developing registration-time decisions about which new domains will likely see subsequent malicious employment appears quite daunting given the very large rate at which new domains appear (tens of thousands per day for `.com`). Instead, existing DNS reputation systems use either evidence of malicious use (*e.g.*, appearance of names in a spam trap) or the characteristics of DNS lookup traffic [4, 9]. Such systems generally must observe a significant volume of DNS lookups before determining the reputation to associate with a domain.

In this work we seek to understand the nature of spammer domain registrations with an ultimate goal of hampering the ease with which attackers currently acquire large volumes of registered domains. We do so by analyzing a range of registration-time features as manifest in changes seen every 5 minutes to `.com` over a 5-month period. For partial ground truth in assessing which of these registrations reflected spammer activity, we draw upon domains identified by several large blacklist feeds associated with email spam campaigns.

Our study develops the following findings:

- We confirm the earlier finding that *only a handful of registrars account for the majority of spammer domains* [72]. 70% of spammer domains came from 10 registrars, though these registrars accounted for only about 20% of all new domains added to the zone. Thus, miscreants appear to prefer those specific registrars, and positive actions from these registrars could have significant impact in impeding the use of large volumes of newly registered domains for spam activity.

- *Groups of domains registered by a given registrar at a single time exhibit two statistically distinct patterns.* We show that groups of registrations very often follow a distribution well-described by a compound Poisson process, but many registrars also exhibit registration "spikes" that this process would produce only with exceedingly low probability.
- *Spammer domains occur in such "spikes" with much more prevalence than in general (non-spike) registration activity.* This finding suggests that spammers find economic and/or management benefit to registering domains in large batches, and thus detection procedures that leverage the presence of such spikes could force spammers to adopt less efficient approaches for their registrations.
- *Spammers frequently re-register expired domains that originally had a clean history.* Presumably using such names alleviates spammers of the burden of generating plausible-looking names (though we also observe algorithmically generated names), providing textual diversity as well as a benign past reputation that may aid in initially avoiding detection.

These findings will ultimately lead to the development of a detection procedure that can accurately identify names intended for malicious use at *time-of-registration* rather than only later at *time-of-use*, as we will show in Chapter 6.

The remainder of this chapter is organized as follows. Section 5.2 introduces the taxonomy in domain registration. Section 5.3 describes the datasets that we collected and used in our analysis. In Section 5.4 we use the datasets to illuminate the benefits of identifying spammer domains at registration time. We then proceed with investigating the prospects for doing so, starting in Section 5.5, which analyzes the distribution of spammer and non-spammer domains across registrars and DNS servers. Section 5.6 presents our findings regarding bulk registration and our approach to identify registration spikes. Section 5.7 associates the domain life cycle with registration to dissect spammers' strategies to acquire domains. Section 5.8 presents summary of this chapter.

5.2 Background: DNS Registration Process and Life Cycle

To set the context for our work, here we sketch the process by which malicious parties (and others) register domains and the subsequent life cycle regarding use of the domains.

Figure 19 shows the domain registration process. There are three roles in the figure: registrants (domain registration applicants), registrars (*e.g.*, `GoDaddy`), registries (*e.g.*, Verisign). Registries are responsible for managing the registration of domain names within the top-level domains (TLDs) and generating the zone files that list domain names and their authoritative nameservers. For example, Verisign serves as the registrar for `.com`, CNNIC for `.cn`, and DENIC for `.de` [99]. In this work we focus on `.com`, the largest TLD [120], which has long reflected a major target abused by miscreants for spamming activities [113].

ICANN accredits registrars, which contract with TLD registries to provide registration service to the public. Presently around 900 registrars exist across all TLDs, the bulk of which serve `.com` (and often other TLDs) [98]. A registrant selects a *designated registrar* to register a domain. The designated registrar in turn connects to the registry's SRS (Shared Registration System) via EPP (Extensible Provisioning Protocol, RFC5730 [48]) or RRP (Registry Registrar Protocol, RFC3632 [47]) to manage the zones. The registry updates the corresponding DNS zone information in the database and uses RZU (Rapid Zone Update) to add the DNS information in the top-level domain nameservers. Domain registration operates in a real-time fashion, resulting in only a short interval between registration requests and domains becoming active in the zone.

Figure 20 indicates the life cycle of a second-level domain in the `.com` zone. We show only a simplified cycle; see ICANN's registry agreements [52] for a full description of the possible states of a domain. In order to obtain a domain, registrants need to select available domain names that are not registered and in use. The registration term usually ranges from 1 year to 10 years. If the registrant chooses to renew a domain, the expiration date will be extended and the domain remains in the zone files. The renewal could occur at any time during the registration period, the *Auto-Renew Grace Period* (for domains the registrar has already marked for renewal) and the *Redemption Grace Period* (for domains marked for deletion). If the registrant chooses not to renew, the domain expires, gets removed from the

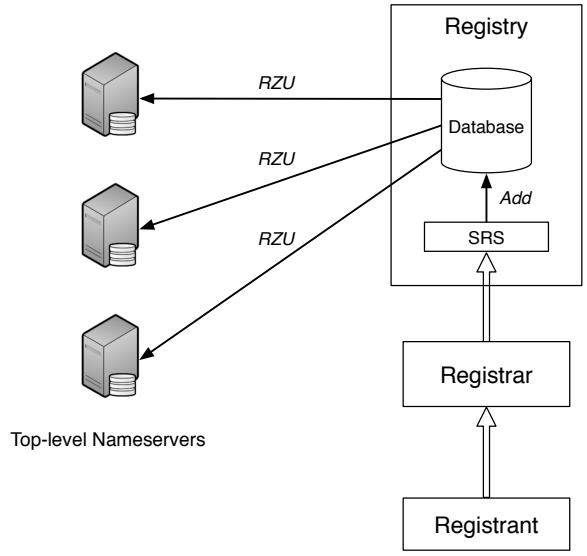


Figure 19: Process of second-level domain registration.

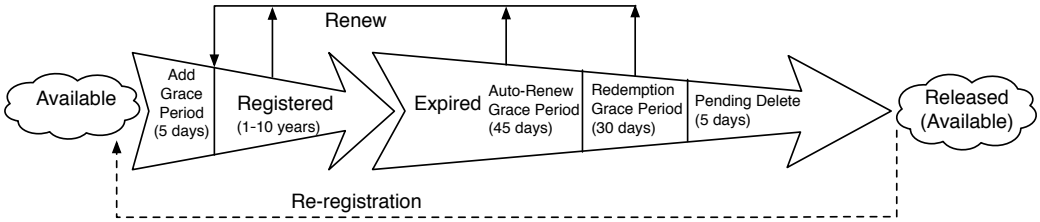


Figure 20: Life cycle of a second-level domain.

zone and becomes available for others to register. Two special periods mark the beginning and end of a domain's life cycle. The 5-day *Add Grace Period* begins immediately after domain registration, allowing the registrant to change their mind, undo the registration, and receive full credit for the registration fee [53]. To limit *domain tasting* abuse, *i.e.*, taking advantage of no-cost trial periods for domains, registrars limit the number of registrations a registrant may revert per month. The domain enters a 5-day *Pending Delete Period* that prevents further alterations to the domain's status before it gets unregistered and becomes available for re-registration [83,119]. We explore how this life cycle relates to the registration of spammer domains in Section 5.7.

Table 8: Summary of data feeds in the domain registration analysis.

Data	*Collection period*	*Update granularity*	*.com domains*
DNZA	March–July 2012	5 minutes	12,824,401
Spam trap	March–October 2012	real time	65,298
URIBL	March–October 2012	hourly	149,555
SURBL	August–October 2012	hourly	490,439

Table 9: Monthly data statistics of .com domain registrations.

	New domains	*Subset of new domains appearing in spam trap*	*Subset of new domains appearing in URIBL*	*Subset of new domains appearing in SURBL*	*Subset of new domains appearing in spam messages* ***or*** *blacklists*
March 2012	2,832,867	6,072	12,572	18,875	24,458
April 2012	2,596,192	3,970	12,111	21,824	27,300
May 2012	2,641,466	4,091	10,726	21,616	25,936
June 2012	2,383,010	2,861	8,651	21,872	24,763
July 2012	2,389,636	2,958	8,875	29,525	32,394
Registrations over 5 months	12,824,401	19,930	52,857	113,358	134,455

5.3 Data Collection

In this section we describe the datasets used in our analysis, which we summarize in Table 8. Our primary dataset consists of changes made to the .com zone every five minutes for a 5-month period, March–July 2012. In addition, we interpret the significance (*i.e.*, spammer or otherwise) of new registrations in .com based on any subsequent appearance of a given domain in either a "spam trap" we operate or a well-known blacklist. We term a newly registered domain that appears in either the spam trap or on one of the blacklists as a *spammer domain*, and a domain that does not as a *non-spammer domain*.

Domain Registration. Verisign operates the .com zone under contract to ICANN. Changes to the zone appear in *Domain Name Zone Alert (DNZA)* files, which indicate (1) the addition of new domains, (2) the removal of existing domains, and (3) changes to existing domains in terms of revisions to their associated nameservers. Our data includes captures of the DNZA files as recorded every five minutes, time periods we refer to as *epochs*.

Registrars and History. Domain registrations must be executed by an ICANN-accredited

registrar chosen by the user registering the domain (the "registrant"). The registrant pays the registrar a fee for this service. In general, we have no visibility into the registrants associated with particular domains (sometimes `WHOIS` information provides their identities, but numerous registrars provide a "private registration" service that masks this information). One can however obtain information about a given domain's registrar based on `WHOIS` information, or using third-party services such as DomainTools [25]. Thus, we can only attempt to tease out the registration behavior of individual users as inferable from the registration activities of individual registrars.

A given domain added to the zone might reflect either a first-time registration or a re-registration of a previously registered domain. We can distinguish these two based on historical `WHOIS` information; for re-registered domains, we can obtain when the domain was previously deleted from the zone [118].

Identifying Spammer Domains. In general we would like to associate with domains a label indicating whether an attacker registered the domain for spamming purposes. Since we lack comprehensive ground truth regarding the ultimate use of domains, to this end we use two proxies: subsequent appearance of a newly registered domain in: (1) an email spam campaign, or (2) a domain blacklist.

For the first of these, we operated a spam trap, *i.e.*, our own domain with an associated mail server that has no legitimate email addresses. We can confidently consider all emails sent to the spam trap as spam. Although the spam contains non-spam related domains (*e.g.*, `youtube.com`), by restricting our focus to domains recently registered (March–July 2012) we can filter down the domains appearing in the spam trap to those very likely used for spamming.

For the second, we subscribed to three major DNS blacklists, URIBL, SURBL, and Spamhaus DBL. During our subsequent analysis we found strong indications that the Spamhaus DBL very likely uses registration-time features to establish the reputation of a domain (see the discussion in Section 5.4.3). Given that, then since part of our focus is to assess to what degree registration-time features correlate with a domain's subsequent employment in an abusive context, for our purposes we cannot soundly use a domain's

presence on the Spamhaus DBL as such an indicator. Consequently, we omit this source from our analysis other than to demonstrate the indicators that it uses such features for blacklisting.

Summary of Data. Table 9 shows the number of second-level `.com` domains registered in each month, and the subset of those registrations that later appeared in either our spam trap or on one of the two blacklists. Of the 12,824,401 second-level domains registered in the `.com` zone over five months, 134,455 reflect spammer domains.

5.4 Longevity of Spammer Domains

In this section we look at several facets of the time periods over which spammers employ their domains: the age of domains (time since registration) when they appear on blacklists or in spam campaigns; the amount of time during which domains continue to see use once spammers begin to employ them; and the amount of time between a *recent* registration of a spammer domain and its subsequent appearance. Our examination shows that detecting spammer domains at the time of their registration can offer significant advantages.

5.4.1 Age of Domains Used for Spamming

We first consider the degree to which spammers employ relatively "fresh" (recently registered) domains in their spam campaigns. If spammers primarily rely upon long-lived domains, then we cannot hope to gain much benefit from disrupting the registration of new spammer domains.

Figure 21 shows the distribution of the amount of time elapsed between the registration of a given `.com` spammer domain and its appearance in either the URIBL blacklist or our spam trap. (We omit results for SURBL because we lack sufficiently long data from it to compute a comparable distribution.) In particular, for all `.com` domains that appeared in either URIBL or our spam trap from March–July 2012, we determine the domain's date of registration, and plot the difference between that time and the first such appearance. Overall, 35–40% of the domains were registered within the past 30 days, and 40–50% within 60 days. (In addition, since listings in URIBL will likely lag behind actual use, and the spam trap will include some long-lived benign domains such as `google.com`, the age of

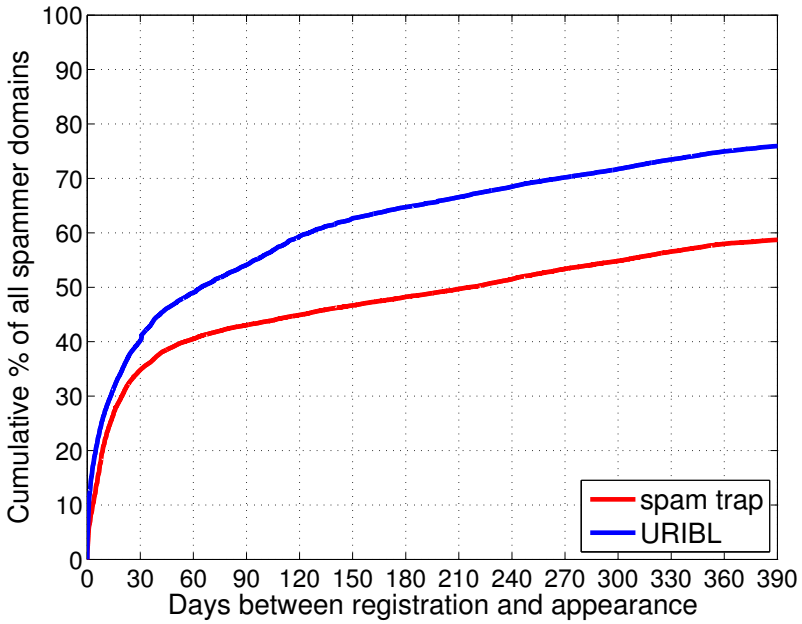


Figure 21: Distribution of days between domain registration and appearance of a *.com* domain in either our spam trap (red) or URIBL (blue).

actual spammer domains at time of first use will skew somewhat lower than these figures.)

Because domain registrations represent a direct cost for spammers, the fact that spammers frequently employ domains registered quite recently (within a few months) indicates that they have an ongoing need to acquire new domains. Thus, if we impair their registration activities, we can add friction to their general enterprise. In addition, given the quantity of domains that spammers use, we would expect that their need to continually acquire new domains will incline them towards registering new domains in batches, a feature that we analyze in Section 5.6.1.

5.4.2 Duration-of-Use in Spam Campaigns

Another facet of spammer domain longevity concerns for how long a spammer uses a given domain. If domains see only brief periods of use, then activity-based blacklisting will fail to effectively block the spammer's fruitful employment of a domain unless the blacklisting occurs very soon after the onset of use. If so, then the benefits of identifying spammer domains prior to use, such as at-time-of-registration, rise.

We can assess duration-of-use from our spam trap data, and indeed we find that more

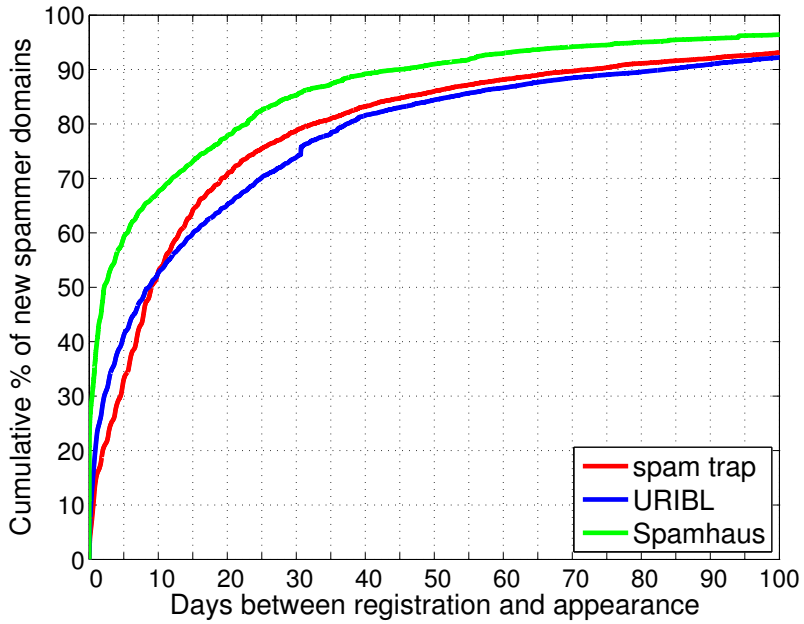


Figure 22: Distribution of days between domain registration and appearance of a newly registered *.com* domain in either our spam trap, URIBL, or the Spamhaus blacklist.

than 60% of the .com domains observed in the spam trap appear during only a single day (75% for $\leq$ 10 days, and only 5% for $\geq$ 60 days). This "single-shot" nature of most of the spammer domains complicates blacklisting efforts—though it also may reflect the efficacy of such efforts at narrowing the window during which spammers can profitably use their domains—and highlights the benefit of at-time-of-registration detection.

5.4.3 Lifetime of Recently Registered Domains

Finally, we examine the use by spammers of newly registered domains *given* that we flagged it as a spammer domain (and thus it necessarily appeared in our spam trap, or in one of our blacklists, during the coming months). Figure 22 shows the distribution of the time between the registration of such spammer domains and their appearance in our spam trap or in a blacklist.[1] We see that a number of days may pass after registration prior to the domain's appearance. This delay again indicates we might gain significant benefit from identifying

[1] It is important to keep in mind the distinction between Figure 21 and Figure 22. The former is conditioned on any domain appearing in either the spam trap or the blacklist during the five-month period; the latter is conditioned on the appearance of any domain *registered* during the five-month period.

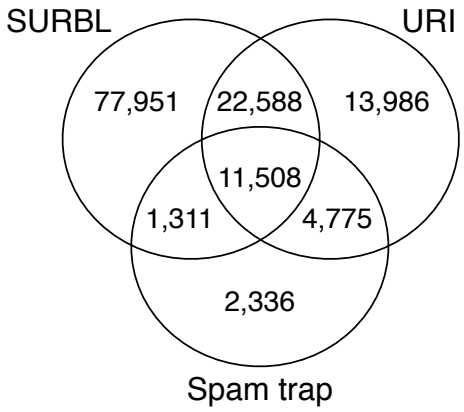


Figure 23: Venn diagram of spammer domains for different identification methods.

spammer domains at the time of registration, before any activity takes place.

The figure also shows the delay between registration and blacklisting for the Spamhaus DBL. We observed that Spamhaus blacklists domains much earlier than URIBL—or even than the appearance of the domain in our spam trap. We confirmed with Spamhaus that they base their DBL entries in part on information gathered at registration time to facilitate more proactive blacklisting. Because Spamhaus uses registration-time features to construct their blacklist, the presence of a Spamhaus domain in a blacklist does not provide *independent* evidence that features we consider for at-registration-time detection indeed will have power for identifying spammer domains. Given this lack of independence, we refrain from further analysis treating the appearance of a domain on the Spamhaus blacklist as separate confirmation of the domain as one employed by spammers.

Finally, we examine the extent to which any particular blacklist covers the full set of spammer domains, and the extent to which these blacklists overlap with one another. Figure 23 shows the intersection of spammer domains registered from March–July 2012, based on the information from our spam trap, and from URIBL and SURBL. We observe that each data source identified many spammer domains that did not appear in the other information sources. This lack of overlap presumably indicates that different blacklist sources use different criteria to determine whether to include a domain in its blacklist.

Table 10: The 10 registrars that registered the greatest number of spammer domains.

	Spammer domains			*All registered domains*	
Registrar	*Number*	*Percentage*	*Cumulative percentage*	*Percentage*	*Cumulative percentage*
eNom, Inc.	36,245	27.03	**27.03**	7.62	**7.62**
Moniker Online Services, Inc.	25,488	19.01	**46.05**	0.67	**8.30**
Tucows.com Co.	5,996	4.47	**50.52**	6.28	**14.57**
INTERNET.bs Corp.	5,786	4.32	**54.83**	0.70	**15.27**
Bizcn.com, Inc.	5,638	4.21	**59.04**	0.70	**15.97**
Trunkoz Technologies Pvt Ltd.	4,577	3.41	**62.45**	0.05	**16.02**
PDR Ltd. d/b/a PublicDomainRegistry.com	3,595	2.68	**65.13**	3.08	**19.10**
OnlineNIC, Inc.	2,857	2.13	**67.26**	0.70	**19.80**
Center of Ukrainian Internet Names	2,781	2.07	**69.33**	0.06	**19.86**
Register.com, Inc.	2,540	1.89	**71.22**	2.18	**22.04**
GoDaddy.com, LLC	5,532	4.13	75.35	30.75	53.79

5.5 Spam Domain Infrastructure

In this section we briefly look at the infrastructure supporting individual spammer domain registrations: the registrars used to register these domains, the DNS servers initially selected to resolve the domains, and how this infrastructure compares with that used for non-spammer domains. Our analysis supports the following:

- Nearly 70% of spammer domains originated from 10 registrars, while those registrars accounted for only 20% of all newly registered domains over the 5 months of our data.
- Spammer domains primarily use the regular authoritative DNS servers operated by the registrar, at least initially. This finding suggests that efforts to proactively blacklist spammer domains should focus on registrar-level analysis rather than DNS-server analysis.

We now develop these findings in more detail.

5.5.1 Registrars Used for Spammer Domains

We first examine the proportion of registrations at each registrar that correspond to spammer domains and how this proportion varies by registrar. Table 10 shows the registrars, ranked by the number of spammer domains that they registered over the five-month period of our study (shown in the second column); the third column shows the percentage of known spammer domains registered by that registrar. The fourth column indicates the cumulative percentage of spammer domains for the top registrars. Interestingly, 46% of the

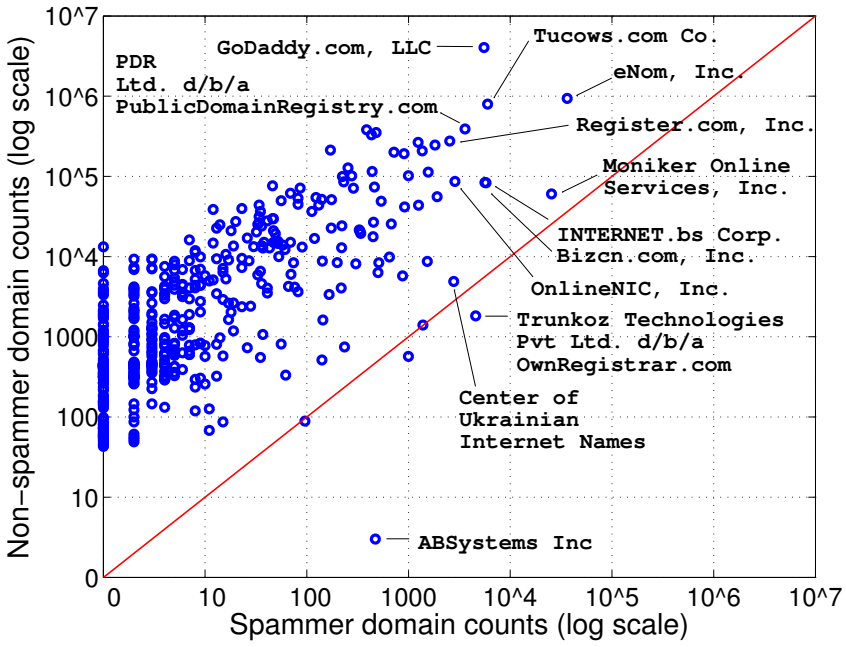


Figure 24: Counts of spammer versus non-spammer domains on the registrars.

spammer domains correspond to just two registrars. This statistic implies that the positive actions from a small set of registrars might significantly frustrate the use of newly registered spammer domains. We treated `GoDaddy` separately because it manages significantly more domains than other registrars; hence, even though it registers a significant number of spammer domains, the number of spammer domains that it registers remains a small fraction of the total number of domains that it registers.

Our findings agree with a similar study by Levchenko *et al.* [72], and also with an independent study that ranks the registrars serving rogue Internet pharmacies [97]. Indeed, three registrars—`Moniker`, `Tucows` and `Bizcn.com, Inc.`—appeared in both studies as top-ten registrars for spammer domains.

Next, we explore how the registrars compare, in terms of the number of spammer and non-spammer domains that they register. Figure 24 shows the number of spammer and non-spammer domains that each registrar registered over the course of our study; each dot represents a registrar. Dots above the diagonal show registrars that registered more non-spammer domains than spammer ones, and dots below the diagonal line reflect a ratio

towards higher spammer domain registrations than non-spammer. The figure labels the top registrars for spammer domains, and shows that 19 of the 919 registrars in our study registered more than 1,000 spammer domains. We see that spammers often use popular registrars to register spammer domains, perhaps because doing so may make it more difficult to identify spammer domains solely based on the registrar. On the other hand, for some registrars that do not register many domains, their fraction of spammer domains can be strikingly high (`ABSystems`, in particular [2]).

We speculate that the decisions by spammers regarding domain registrations are driven by both economic concerns (price of registration) as well as the ease of managing multiple domain registrations. Regarding the first of these, we note that registrars charge a range of fees. For example, `eNom` sets special prices for resellers, `GoDaddy` offers cheaper prices for bulk registration, and `INTERNET.bs` provides free private `WHOIS` protection. The management features that each registrar provides determine how easily a customer can manage a domain; for example, `eNom` provides APIs that allow users to manipulate the domain zone entries, and `Moniker` allows up to 500 domain registrations at a time.

5.5.2 Authoritative Nameservers

The zone updates we use for our study include NS records associated with each new domain. During March–July 2012, we observed 12,824,401 newly registered `.com` domains, but only 242,790 authoritative DNS servers assigned to those domains. We thought that we might find that spammer domains are disproportionately hosted on certain sets of authoritative DNS servers, which we assessed using three metrics:

- *Toxicity.* The percentage of domains that a nameserver hosts that are spammer domains. This metric represents the extent to which an authoritative nameserver sees use mainly in support of spamming activity. A toxicity of 100% indicates that the nameservers appear to operate solely under miscreant control; the presence of such nameservers in a new domain registration could effectively identify new spammer

[2] The badness of `ABSystems` comes as no surprise. This registrar effectively acts as the DNS infrastructure division of a large spamming operation known as "Quick Cart Pro."

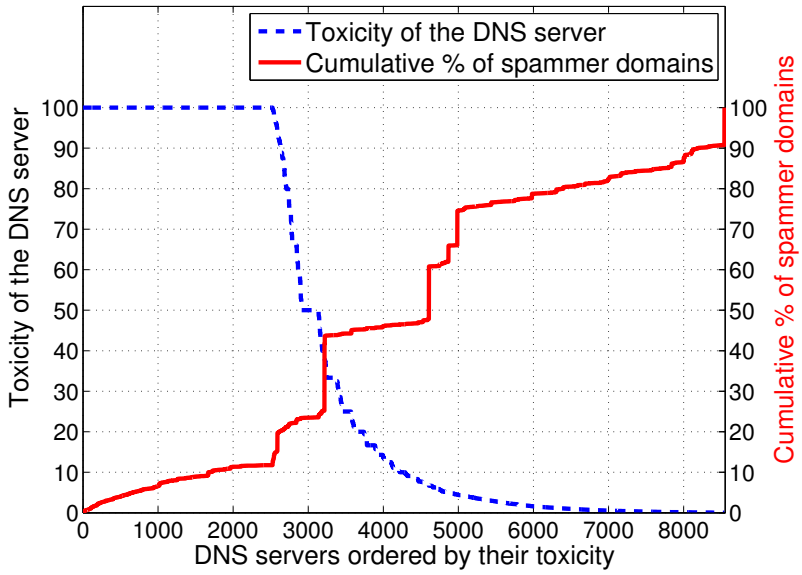


Figure 25: Cumulative distribution of spammer domains on DNS servers (ordered by toxicity).

domains.

- *Duplication.* Owners of a given domain typically use multiple DNS servers to host the same domain to achieve redundancy in case of failure [80]. Intuitively, a group of domains hosted by the same set of authoritative nameservers likely have some relationship. We compute the Jaccard index to measure the similarity of the authoritative nameservers in terms of the domains they host. Suppose there exist N nameservers, each of which hosts a set of domains D_i. We compute $|\cap_{i=1}^{N} D_i|/|\cup_{i=1}^{N} D_i|$. A higher Jaccard index for a pair of nameservers indicates a high overlap in terms of the set of domains that those nameservers resolve. This association may ultimately help with identifying groups of nameservers commonly used to host spammer domains.
- *Association.* The percentage of domains hosted on a nameserver that belong to a particular registrar. We define the registrar with the highest association score for a nameserver as the *primary registrar* for that nameserver.

Figure 25 shows the cumulative distribution of spammer domains over the DNS servers. The X-axis shows the indexes of DNS servers ordered by their toxicity; 8,543 of 242,790 DNS servers hosted spammer domains. The figure has two Y axes. The blue dashed curve shows

Table 11: Top nameservers hosting spammer domains. †*Note:* the domains registered on ns1.google.com and ns2.google.com migrated to other DNS servers immediately after registration.

DNS server	*Common spammer domains*	*Toxicity*	*Jaccard index*	*Primary registrar*	*Assoc. %*
ns[1,2].monikerdns.net	21,256	38.46	1.00	Moniker Online Services, Inc.	99.77
ns[3,4].monikerdns.net	17,012	33.74	1.00	Moniker Online Services, Inc.	99.75
dns[1-5].name-services.com	16,955	6.75	0.97	eNom, Inc.	99.88
dns[1-5].registrar-servers.com	11,016	4.58	0.99	eNom, Inc.	99.86
ns[1,2].google.com†	5,957	93.99	0.99	Trunkoz Technologies Pvt Ltd. d/b/a OwnRegistrar.com	19.90
ns[1,2].directionfindfree.com	5,302	5.55	1.00	Tucows.com Co.	82.00
ns[1,2].speee.jp	2,400	37.94	1.00	OnlineNIC, Inc.	98.70
ns[1-4].name.com	1,345	1.52	0.96	Name.com LLC	99.76
ns0[7,8].domaincontrol.com	1,089	0.17	1.00	GoDaddy.com, LLC	95.35
ns[3,4].cnmsn.com	1,047	24.89	0.99	Bizcn.com, Inc.	99.38

the toxicity of each DNS server and corresponds to the Y-axis on the left side. The red solid curve shows the cumulative percentage of spammer domains for the set of nameservers ranked by their toxicity, and maps to the Y-axis values on the right side of the plot. The nameservers hosting only spammer domains (*i.e.*, with toxicity 100%) only account for about 10% of all spammer domains.

Table 11 lists the top nameservers associated with spammer domains, in terms of the total number of spammer domains that they host. We group servers together if their Jaccard index exceeds 0.95, to ensure grouping similarity; we use a regular expression to represent groups of similar domains. For nameservers in common groups, we calculate the metrics based on common domains. The second column shows the number of common spammer domains for each DNS server group; we rank the groups in descending order of the number of common spammer domains for that group. The third and fourth columns indicate the server toxicity and the Jaccard index of duplication, respectively. When spammer domains are sheltered in large registrars (like `Moniker` or `eNom`), these registrars provide and operate their authoritative DNS servers, also hosting a large number of legitimate domains.

It becomes clear from these results that although spammers prefer certain nameservers in some cases, no clear-cut separation exists between nameservers used for spamming and those used for benign purposes. Hence, our earlier hope fails to pan out: we do not see how to fruitfully leverage the nameservers associated with domain registrations to identify

spammer domains. In the next section, we turn to exploring to what degree the patterns that spammer registrations exhibit can help distinguish spammer domain registrations from benign ones.

5.6 Detecting Registration Spikes

In this section we examine the extent to which spammers register domains in abnormally large batches ("spikes"). We first present evidence that suggests that domains associated with spamming are registered in groups. We then show that the number of domains that a given registrar registers in a given five-minute interval usually follows a distribution well-modeled by a compound Poisson process—but that many registrars also exhibit registration spikes that this process would produce only with exceedingly low probability. From this we conclude that these spikes represent a different underlying process than that corresponding to routine activity.

After deriving a model to explain both normal and anomalous registration activity, we show that spammers tend to register batches of domains in such spikes more often than do non-spammers. Our results suggest that methods to reliably identify registration spikes can serve as a useful feature for proactively detecting names that are subsequently used in spam campaigns.

5.6.1 Bulk Registrations by Spammers

Spammers acquire large volumes of domains to remain agile when conducting their operations [72]. We frequently observe spammer domains registered in spikes as large as hundreds of domains within a single five-minute `.com` update. We speculate that such registration behavior occurs due to: (1) convenience; (2) bulk pricing from registrars [36, 82]; or (3) the use of stolen credit cards to purchase a large number of domains in a short period of time, since the fraudulent purchases will trigger detection and result in voiding of the stolen credit card.

We use the term *bulk registration* to refer to the behavior of registering a batch of domains during a period of a few minutes. Since the registrants' information and behavior are not directly observable in the zone updates, we can only infer that a group of domains

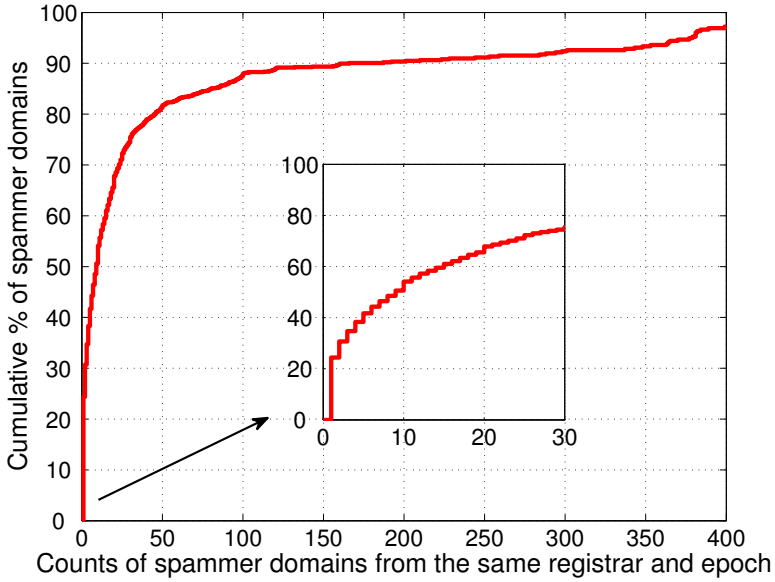


Figure 26: Distribution of bulk spammer domain registration from the same registrar and epoch.

may represent a bulk registration by observing updates with multiple domains within the same ⟨registrar, 5-minute-epoch⟩ tuple. The granularity of data that we have provides only an approximation of the behavior of each registrant because different registrants may register simultaneously from the same registrar, and the same registrant could spread their registrations across multiple registrars. Still, we observe the general tendency for spammers to perform registrations in batches, as we develop below.

Figure 26 shows the distribution of the number of spammer domains registered in the same epoch. The X-axis shows the number of spammer domains observed for a given registrar within a single epoch, and the Y-axis shows the cumulative percentage of spammer domains within such epochs. The inlay shows that 50% of the spammer domains were registered in groups of ten or more. We find that only 20% of the spammer domains got registered in isolation (with no other spammer domains), but more than 10% in batches exceeding 200 spammer domains.

We confirmed that the prevalence of multiple spammer domains registered together is not simply a reflection of general "overcrowding". In particular, when we examined the

registration patterns for the ten registrars (other than GoDaddy) responsible for about 70% of all spammer domains (per Table 10), we observe that only 8% of all registration epochs contained any activity involving spammer domains. Indeed, only Trunkoz has more than 20% of its epochs including such domains; for this particular registrar, about 60% of the epochs involve registrations of spammer domains, indicating that this registrar clearly represents an outlier in terms of reflecting consistently bad behavior.

Thus, we see a general trend reflecting behavior where many (but not all) spammer domains are registered in bulk. In the next section, we attempt to capture this notion in more principled terms by fitting the bulk of temporal registration patterns to a compound Poisson process and identifying registration spikes as epochs that deviate significantly from this distribution.

5.6.2 Detecting Abnormal Registration Batches

The evidence from the previous section suggests that spammers often register multiple domains at the same time. This phenomenon motivates us to identify a way to determine whether a registrar's set of registrations during a given epoch is "abnormally large". If so, then we posit that those registrations are more likely to reflect domain registration activity by spammers. We aim to identify registration activity behavior that *qualitatively* differs from routine (and thus, we presume, likely benign) activity. As noted above, we refer to such a set of registrations as a "spike".

Developing a model for registration batch size. Our challenge is to determine that a given set of registrations crosses the line into "abnormally large", thus constituting a spike. The difficulty we face is that simple approaches for spike detection can lack soundness. For example, simply setting a single threshold (registrations per epoch) may cause us to miss numerous spikes that appear in the registrations for some of the smaller registrars, since for those registrars, an abnormally large set of registrations might not be all that large in terms of absolute volume. If, on the other hand, we instead pick a fixed quantile, such as "treat as spikes all registrations larger than the 99th percentile of a given registrar's registration sizes", then we will necessarily define a subset of each registrar's activity as "abnormally

large"—failing to capture the notion of a *qualitative* difference.

Instead, we strive to develop a principled approach to identify qualitatively different (abnormally large) registration epochs, as follows. We hypothesize that a single model can capture the bulk of a registrar's registration activity (*i.e.*, the distribution of how many names the registrar registers during each of its epochs). We then look for epochs during which, according to that model, the volume of names registered was exceedingly unlikely (in a probabilistic sense). We deem such epochs as qualitatively different, and classify the corresponding set of registrations as a spike.

The first question we face concerns what sort of model to use to capture regular registrar activity. If registrars receive a steady stream of customers who act independently of one another, and each registers a single name, then a Poisson process should capture the corresponding activity well: during each epoch, the registrar registers a number of names corresponding to the number of customers who arrived since the last epoch. For this model, all we need to identify is the rate at which the customers arrive at the registrar with their requests. However, we would expect that diurnal patterns would cause the arrival rate at each registrar to vary over the course of each day, and indeed from inspection we find that this is the case. We adjust for this consideration by separately computing for each registrar the mean number of names they registered for each hour of the day, analogous to the nonhomogeneous Poisson processes used previously in characterizing network traffic [88]. For example, we determine a registrar's registration rate per epoch as the average over all epochs between 10 a.m. and 11 a.m., and then use that rate to parameterize a Poisson process to capture the number of registrations that we expect to occur in each such epoch.

We explored this simple Poisson model and found that while it works well for some registrars, for many registrars it often fails to produce convincing fits to the body of the distribution of names registered per epoch. This provides evidence that customers do not arrive at a registrar independently from one another, the rates at which they do vary significantly more rapidly than on a per-hour basis, and/or customers sometimes register more than a single name at a time in normal activity.

The first two of these possibilities appear somewhat at odds with how we expect users to

function under normal circumstances, which leads us to consider instead the third option. By employing a *compound Poisson* process, we can capture customers who arrive independently at a fixed rate, but each of whom makes a number of registrations drawn from a given distribution. The general compound Poisson formalism does not require a particular family for this second distribution. However, we achieved quite good results by using a second Poisson distribution; we later discovered that previous work has also modeled consumer purchase behaviors using a compound Poisson process that employs a second Poisson distribution [111].

In summary, for normal registrar activity we capture the number of domain registrations per epoch as $Y = \sum_{i=1}^{N} X_i$, where N represents the number of registrants during the epoch, and follows one Poisson distribution, and X_i $(1 \leq i \leq N)$ are *i.i.d.* Poisson distributions capturing the number of domains each registrant registers. For this model, we have:

$$\begin{aligned} E(Y) &= E(N)E(X_i) \\ Var(Y) &= E(N)[Var(X_i) + E(X_i)^2] \end{aligned}$$

Fitting the distribution. Given this model, we now turn to how to fit it to a given registrar's activity. (A reminder, we do this for each hour of the day separately, to accommodate diurnal patterns.) We need to estimate N's parameter, λ_N, and that for the X_i, λ_X. Given they are Poisson distributions, we have: $E(N) = \lambda_N$, $E(X_i) = \lambda_X$, and $Var(X_i) = \lambda_X$, and therefore:

$$\lambda_X = \frac{Var(Y)}{E(Y)} - 1, \quad \lambda_N = \frac{E(Y)}{\lambda_X}.$$

Of course, we cannot simply compute these estimates from each registrar's registration process because our goal is precisely to try to identify registration events that do *not* conform to the registrar's usual activity. For any given registrar, we do not know whether any of these qualitatively different events even exist, but we have strong confidence that they do exist for at least some registrars.

We thus refine the process of fitting the distribution based on the following intuition. Because the events we seek to detect reflect abnormally large registration batches, they will occur in the upper tail of the distribution of all of a registrar's registrations. Therefore, for

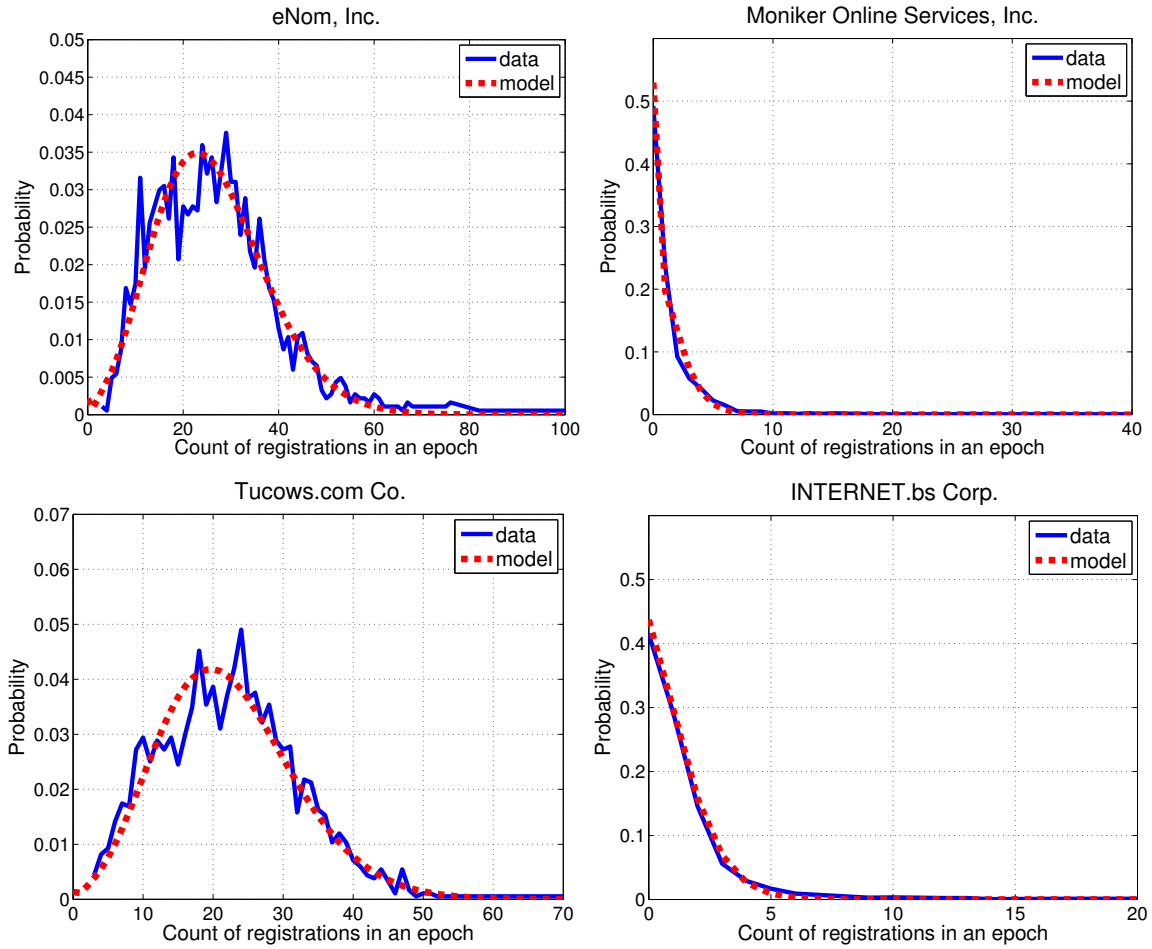


Figure 27: Compound Poisson processes fitted to the count of registrations per epoch for 4 registrars (hourly window, 10AM–11AM ET).

each registrar we progressively apply different *truncation thresholds* (proportion of the upper tail to discard) to see whether omitting extreme tail values provides us with a better fit of the remaining data to the compound Poisson process. Note that if all of the registration events indeed conform to the same compound Poisson process, then we would expect to do no better—and possibly a bit worse—as we discard upper tail events, since these in fact simply reflect the natural extremes of the process.

We use KL divergence to assess how well the model fits a truncated portion of a registrar's activity: given two probability distributions P and Q, the KL divergence of Q from P is $D_{KL}(P||Q) = \sum_i log_2(\frac{P(i)}{Q(i)})P(i)$, which captures the information lost when we use Q to approximate P. The smaller the KL divergence, the better a model fits the data. For

each registrar, we compute the KL divergence for all of its data versus that for a compound Poisson processes fitted to that data; the same but using the data with the upper 99.5% tail discarded; again, but discarding the upper 99.0% tail; etc., through the 90% tail (*i.e.*, we discard the top 10% of largest registration events). We then take as the best fit the tail truncation (if any) that provides the lowest KL divergence. To demonstrate the fitting results, Figure 27 shows the models and epoch distributions of the first 4 registrars listed in Table 10 regarding to the hourly window between 10 AM and 11 AM US Eastern Time.

If we find that this fit worked best with some of the upper tail truncated, then this provides evidence that the most extreme registration epochs behave qualitatively differently from the bulk of the epochs. We can in addition then compute the probability of observing those extreme events given the model fitted to the truncated data. If the extreme events are not in fact all that unlikely, then they may simply reflect stochastic fluctuations of a single underlying (compound Poisson) model. For example, if when truncating the upper 1% tail, we find that truncated model predicts probabilities for the points in the tail as 0.5%, then in fact those points are not so extreme, given the model, and we should not consider them as reflecting qualitatively different behavior. On the other hand, if those points have predicted probabilities of 0.005%, then they are quite unlikely, bolstering evidence that they represent a fundamentally different process.

In the next section, we refine the model by assessing each of these probabilities and explain how we make a final determination of whether a given registration epoch reflects a truly abnormal "spike".

5.6.3 Refining Threshold Probabilities

The compound Poisson model that we have derived enables us to assess the probability of observing a given number of domains registered by a registrar in a five-minute epoch. A low probability indicates a rare event; if the probability is sufficiently low, we can then conclude the presence of an "abnormal" spike. Figure 28 shows the number of domains that were registered in spikes depending on how low we set this probability; the X-axis is log-scaled. The blue dashed curve shows the proportion for all newly registered `.com` domains, and the

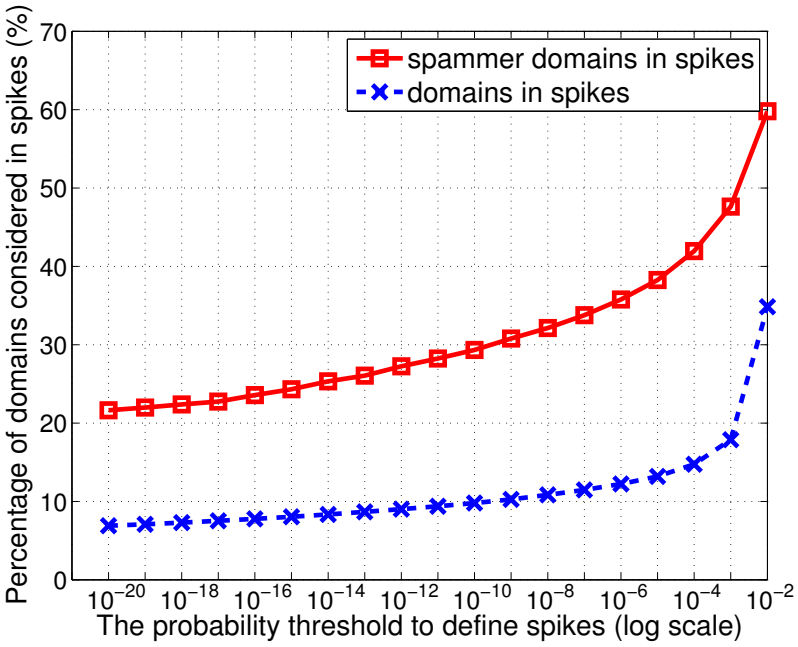


Figure 28: Percentages of domains deemed as registered in spikes according to different thresholds.

red solid curve shows the same statistic for spammer domains. Spammer domains appear in spikes with a much higher likelihood.

We also observe that the slope of the curves increases significantly at a probability of 10^{-3}, suggesting a modal change at that point. For this range of probabilities, the model incorporates spikes that arise simply due to stochastic fluctuations of the normal model, rather than reflecting qualitatively different registration behavior. To avoid mis-classifying registration events for this range of probabilities, we propose defining a spike as a registration size with probability $\leq 10^{-4}$. With that definition, we find about 15% of all domains were registered in spikes; in contrast, 42% of spammer domains were registered in such spikes.

5.7 Domain Registration Patterns

Spammers can potentially use different strategies to decide on which names to register for use in their campaigns. In this section we analyze the history of domains registered by spammers to assess the different approaches they use. We first define different types of registrations in terms of the domain life cycle discussed in Section 5.2. We then show that

some types are significantly more likely than others to correlate with spammer domains. We finish with a look at the nature of the names spammers choose when creating new domains.

5.7.1 Domain Categories

The most basic property of a domain registration concerns whether the domain is **brand-new**, *i.e.*, has never appeared in the zone before, and thus now gets registered for the first time. Such domains have no registration history.

On the other hand, a **re-registration** reflects a name that previously appeared in the zone that the registrant now registers once more after its expiry from the previous owners. For re-registration domains we possess registration history, such as previous registrar(s), registration time(s) and deletion time(s).

We further characterize re-registered domains as either *drop-catch* or *retread*. The former refers to a domain re-registered immediately after its expiry, a phenomenon that occurs quite frequently [24]. Conversely, if some time elapses between a domain's prior deletion and its re-registration, then we term it as a "retread". Thus, the "drop-catch" and "retread" categories are mutually exclusive, and together comprise all members of the "re-registration" category.

5.7.2 Prevalence of Registration Patterns

How common is each registration pattern? We define a domain registration as drop-catch if the domain was deleted and re-registered in the same 5-minute epoch. If more time elapses for a re-registered domain, then we consider it a retread. Among the spammer `.com` domains that were registered over the 5 months, 68% were brand-new, 30% were retread, and 2% were drop-catch.

Which registrations are more likely to reflect spammer domains? To better understand the role of each type of registration in spamming activity, we investigate the conditional probability that a registration reflects a spammer domain, given a specific category of registration. For example, to calculate the conditional probability of observing a spammer domain given that the registration is a retread, we divide the count of domains

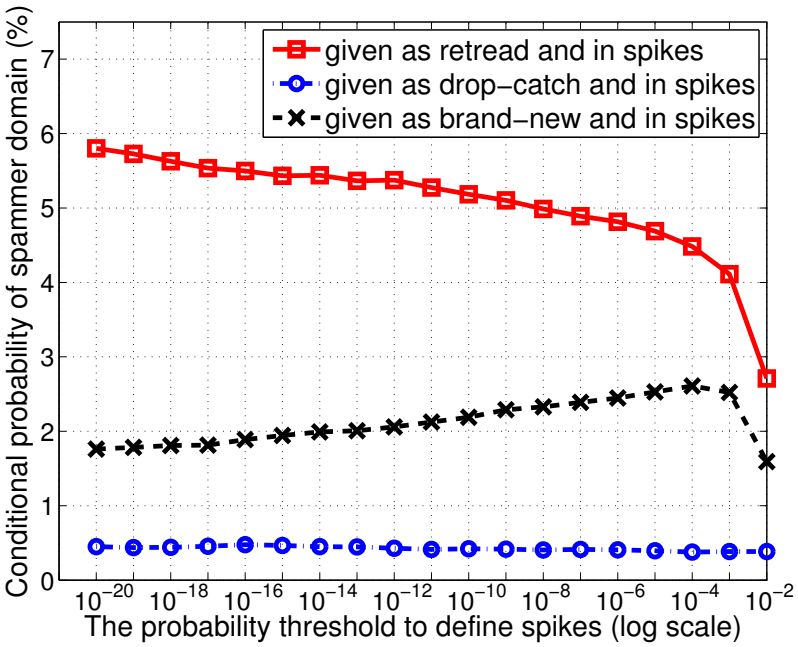


Figure 29: Conditional probabilities of registrations reflecting spammer domains given that the registration appeared in a spike, for retread, drop-catch, and brand-new registrations.

that are both retread and spammer by the count of retread domains.

This procedure then gives us the conditional probabilities of being a spammer domain given that the registration is retread, drop-catch, or brand-new as 1.34%, 0.33%, and 1.01%, respectively. Retread and brand-new have higher conditional probabilities of reflecting spammer domains compared to drop-catch registrations. One possible explanation for this could be that spammers simply use drop-catch domains less often in their spam campaigns. Usually registrars charge higher prices to purchase drop-catch domains; for example, three major drop catching services—Namejet.com, Pool.com, and Snapnames.com—charge $59, $60, and $69, respectively, for drop-catch registrations, significantly higher than typical domain registration rates of around $8–12 per registration [26]. Thus, drop-catch registrations appear significantly less economical for spammers.

Interestingly, if we *also* condition on the registration event occurring in a spike, certain types of registrations become much more likely to reflect spammer domains. Figure 29 shows the conditional probability of observing a spammer domain given a specific category *and* registration occurring in a spike (as defined in Section 5.6). The X-axis indicates the

probability thresholds in log scale. Each curve shows the conditional probability of a spammer domain for different spike detection threshold values. The red solid curve shows that the conditional probability of a spammer domain given that the registration is a retread and appears in a spike reaches as high as 6%, significantly higher than the conditional probability of a given spammer domain being a retread alone (1.34%). This observation indicates spammers are adept at finding previously used but expired domains and re-registering them in bulk. The black dashed curve shows the same statistic for brand-new domains; for this category, the conditional probability of spammer domains roughly doubles when observing a spike. The highest conditional probability for spammer domains occurs around a threshold of 10^{-4}, which indicates that when spammers register new domains in bulk, the spikes may not be as large as they are for registration spikes for other categories of domains. This difference may arise from the difficulty of finding large numbers of brand-new domain names that are suitable for use in spam campaigns. Finally, the conditional probabilities for spammer domains occurring in drop-catch registrations are small and do not vary significantly depending on the detection threshold.

5.7.3 Retread Registration Patterns

We have seen that spammers commonly re-register expired domains, especially when performing bulk registrations. Information about domain expiration is publicly released via various channels [83, 119]; spammers, of course, have access to this information and appear to exploit it when selecting the domains to register for subsequent spam campaigns. The majority of the retread registrations that reflect spammer domains were deleted from the zone within 90 days, which indicates that spammers tend to select domains that have expired recently (although not so recently as to qualify as drop-catch domains). We now examine the registration patterns of retread domain registrations in more detail.

Do spammers perform reconnaissance to determine whether a re-registered domain has been previously blacklisted? We study whether the retread registrations that reflect spammer domains have typically appeared in spammer activity in the time period before the spammer decided to re-register the domain. This analysis allows us to better

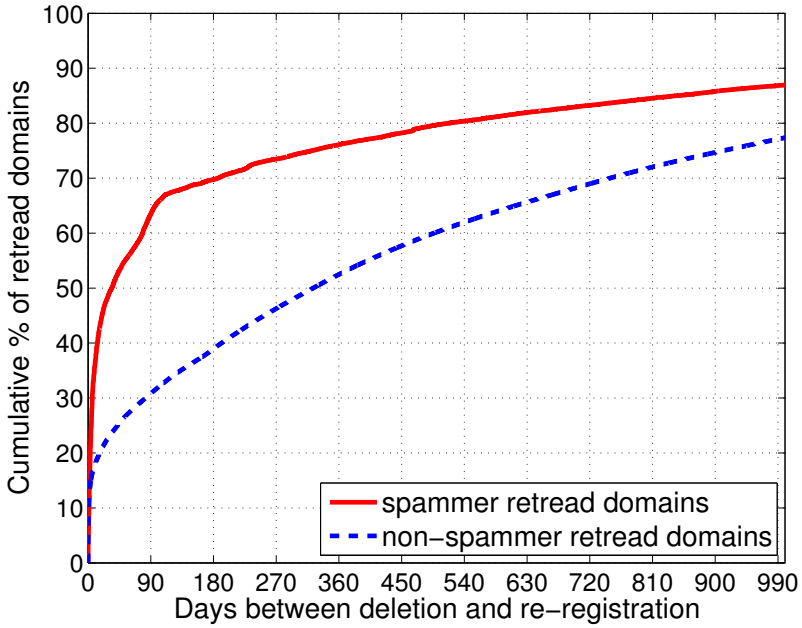


Figure 30: Distribution of days between domain deletion and re-registration.

understand whether spammers specifically aim to re-register expired domains with clean histories. We use both the blacklist reports and spam trap observations from the preceding five months—from October 2011 to February 2012—as our source of historical information about spammer domains. (The SURBL blacklist data was only available for October and November 2011, but we also use it for historical information about spammer domains.) Only 6.8% of the retread registrations during March–July 2012 that reflected spammer domains had ever previously appeared in a blacklist, which suggests that spammers indeed deliberately re-register expired domains with clean histories.

How long are retread domains dormant between periods of registration? Next, we investigate the amount of time that typically elapses between domain expiration and a retread registration. The distribution of the dormancy periods for retread registrations that do not reflect spammer domains is much more uniform than the distribution for spammer domains, which tend to be reused more quickly after they expire. Figure 30 shows a cumulative distribution of the dormancy period for retread registrations; more than 65% of spammer domains were dormant for less than 90 days. If we condition on retread domains

dormant for less than three months, and registered in moderately sized spikes (according to a threshold probability of 10^{-4}), the conditional probability of a retread domain being a spammer domain is 7.7%, again significantly higher than the conditional probability of being a spammer domain based on being a retread registration alone (1.34%).

5.7.4 Naming Patterns for Brand-New Domains

We now study the naming conventions that spammers use when registering brand-new domains, focusing in particular on such domains registered in spikes (once again using 10^{-4} as the threshold probability for defining a spike). We first compare the proportion of spammer and non-spammer domains that are registered in spikes that contain English words. To do so, we compare the domain names against a dictionary, looking for matches against words of at least four letters. We find that about 84% of the brand-new spammer domain registrations occurring in spikes contain an English word, versus about 82% of such non-spammer registrations. Thus, it appears that spammers create names that for the most part will appear plausible, as opposed to employing simple algorithms to crank out gobbledygook. Perhaps spammers seek to avoid detection by domain reputation algorithms that use entropy as a feature (*e.g.*, [6]), or aim to diminish user suspicions and increase the likelihood that users will visit the corresponding website.

We hypothesized that when spammers register new domains in bulk, that they may register domains that represent various combinations of English words that relate to the campaign itself, perhaps with slight variations (*e.g.*, one might expect a spam campaign involving watches to involve the registration of many domains containing the word "watch"). To test this hypothesis, we counted the number of brand-new domains in the same registration spike that share a common word, again considering only English words that are at least four characters long. Figure 31 shows the results of this analysis; many domains in the same spike share no common English words, yet the spammer domains show a slightly higher tendency to have common words in spikes. For example, about 40% of brand-new domains that appear in the same spike contain a common subword overall, yet slightly more than 50% of brand-new spammer domains contain a common subword when they appear

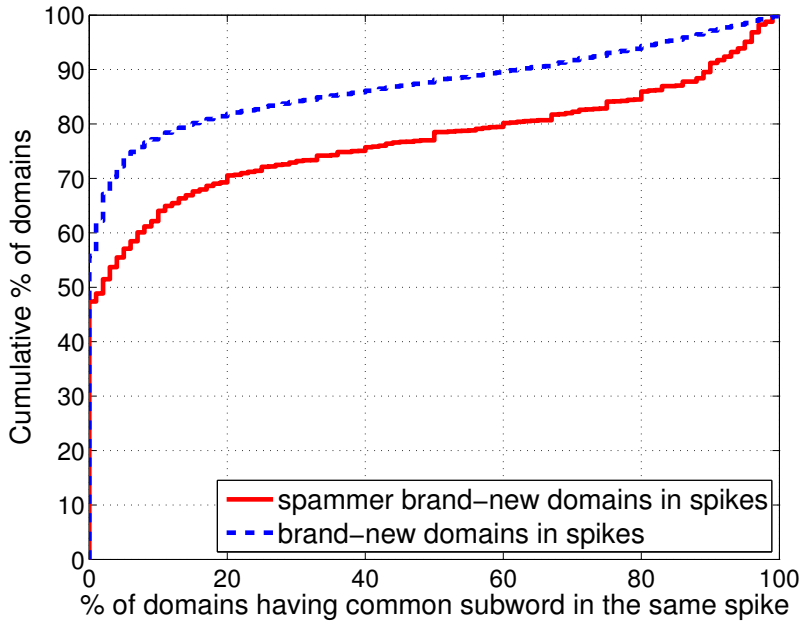


Figure 31: Cumulative percentage of (brand-new) domains that appear in the same registration spike and have at least one English subword in common with another domain in the spike.

in the same registration spike.

5.8 Summary

In this work we have analyzed the domain registration behavior of spammers, including both the infrastructure that they use to register their domains and the patterns that they exhibit when registering them. Our motivation in exploring these behaviors is ultimately to facilitate *time-of-registration* detection of such domains, enabling proactive blocking. We found that nearly half of spammer domains are less than 3 months old; spammer domains are often only used for short periods of time; and current blacklists (with the exception of Spamhaus DBL) identify spammer domains at time-of-use rather than time-of-registration.

We based our study on a large, fine-grained dataset that reflected all changes to the `.com` zone over a five-month period, as seen during five-minute intervals. We then analyzed this data in conjunction with several spam trap and blacklist feeds as *post facto* indicators of spammer domains. After confirming the previous finding that just a handful of registrars account for the bulk of spammer domain registrations, we examined the *registration process*

of each registrar, finding two distinct types of registration activity. In the first, predominant mode, the number of domains registrars register is well-described by a compound Poisson process. By fitting such a process to the bulk of a registrar's registration epochs, we can associate probabilities with "outlier" epochs that register large numbers of domains, allowing us to identify registration spikes that *qualitatively differ* from the registrar's usual registration practices. We then showed that spammers often register their domains in such spikes, whereas non-spammers do so much less frequently.

Spammers also often prefer to re-register domains that previously existed in the zone but subsequently expired. While spammers do not engage in "drop-catching" (immediately re-registering domains that have just expired), they prefer domains that have recently expired (within the past few months) and that in their previous life did not appear to be associated with spamming.

We also analyzed two other time-of-registration features: (1) the degree to which spammers tend to use distinctive nameservers to host their domains, and (2) whether newly registered spammer domains contain common English words. We did not find much discriminatory power for either of these features.

Other than a couple of particularly abuse-prone registrars—which by themselves do not account for a significant portion of spam domains—none of the time-of-registration features that we examined by themselves serve as a "smoking gun". Nevertheless, many features exhibit different behavior for spammer domains versus non-spammer domains. These findings motivate us to develop a time-of-registration detection system by incorporating a supervised learning technique in the next chapter.

CHAPTER VI

PREDATOR: PROACTIVE DETECTION OF SPAMMER DOMAINS AT TIME-OF-REGISTRATION

6.1 Introduction

Determining the reputation of DNS domains as early as possible is critical to protect users from malicious activity and reduce the profit for miscreants. However, existing domain reputation systems establish reputation by observing how the domains are used in practice (*e.g.*, how they are looked up, how they are used) [4, 5, 9, 70, 116], which is unfortunately often too late to prevent attacks from occurring. In this chapter, we explore whether it is possible to develop an algorithm to predict the reputation of a domain name *at registration time*. By predicting the reputation of the domain at registration time, in advance of its actual use, we may be able to prevent some attacks that depend on the DNS before they even take place. Although such a goal sounds appealing, it is also quite difficult. Unlike other DNS reputation systems that can observe how a domain is being used in practice (*e.g.*, by observing lookup patterns and actual use of the domain), a reputation system that operates at registration time has a much more limited set of features for deriving reputation. Such a system can look at features associated with how the domain is originally registered, including temporal characteristics of registration, the delegated registrars that register the domain, and even structural characteristics of the name itself, but all of these features are considerably weaker than anything that is actually associated with malicious activity.

Our goal of both being more accurate and earlier than state-of-the-art blacklists is a tall order, and evaluating the algorithm is also challenging in its own right. There is no perfect "ground truth" for domain reputation, so our best hope is to compare the reputations that our algorithm derives to those of existing blacklists. Indeed, many of the existing blacklists appear to have inaccurate reputation information, especially immediately following registration. Therefore, deriving accurate labels for training and testing our algorithm is not

strictly possible. When considering other blacklists as possible sources of ground-truth labels, we also incorporate additional heuristics, such as whether a blacklist quickly removes a newly registered domain in the time immediately after it was registered and initially listed.

Towards the goal of developing an algorithm for registration-time reputation, we design PREDATOR (*Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration*), a proactive reputation system that can accurately and automatically identify spammer domains at time-of-registration, rather than later at time-of-use. PREDATOR relies on the intuition that miscreants register large volume of domains to make illicit profit and ensure attack agility. The early warning results can mitigate spammer domains, before miscreants can profitably use them in attacks. In particular, we present three contributions:

- We extract a set of salient features to reflect how miscreants register domains, based on hosting infrastructure, registration history, naming patterns, and burst activities.
- We develop a proactive detection system, PREDATOR, by incorporating a supervised learning method to derive a reputation for each domain at registration time.
- We evaluate the detection accuracy using five months of records of second-level `.com` domain registrations, representing over 2 million domain registrations per month. Our evaluation shows that PREDATOR can accurately determine the reputation of new domains with a low false positive rate (0.35%) and a good detection rate (70%), and that such classification can occur at registration time, often weeks before the domains appear on other blacklists.

The rest of the chapter is organized as follows. Section 6.2 introduces the high-level architecture of PREDATOR. We describe statistical features of domain registrations in Section 6.3. Section 6.4 details how we build our classifier. We illustrate the experimental results in Section 6.5. Section 6.6 discusses actionable policies and evasion, and Section 6.7 concludes.

6.2 Architecture

We design PREDATOR to infer domains' reputation immediately after users register the domains. The decision process does not need to examine the Web content on the domains, or to wait until users are exposed in attacks. We intend the system to act as a first layer of defense to mitigate malicious URLs or domains that host spam-advertised sites. Based on the prediction results, network operators or users can either postpone to approve a domain registration, delay the distribution of a URL, or employ more resource-sensitive and time-consuming detection methods to ensure correct classification.

6.2.1 Design Goals

We follow five design goals to develop the domain reputation system.

1. *Early detection.* We aim to detect spammer domains at registration time, before miscreants can profit from the domains. The prediction relies on the domain registration information and patterns.

2. *High accuracy.* The system should have a low misclassification rate, so users can take practical actions based on the prediction results, *e.g.*, to block visits to the domains that have been flagged as malicious.

3. *Scalability.* We aim to process domain registrations under TLDs such as `.com` zone at a rate of around 80,000 new domains per day, with a peak rate of over 1,800 registrations in a single five-minute interval.

4. *Evasion resistant.* After miscreants know the existence of our detection mechanism, they will try to evade it. We aim to extract features that attackers can not easily change without considerably reducing the number or rate of their domain registrations.

5. *Ease of deployment.* We design our system to comply with existing services and protocols, and the prediction output can be easily incorporated with other detection mechanisms.

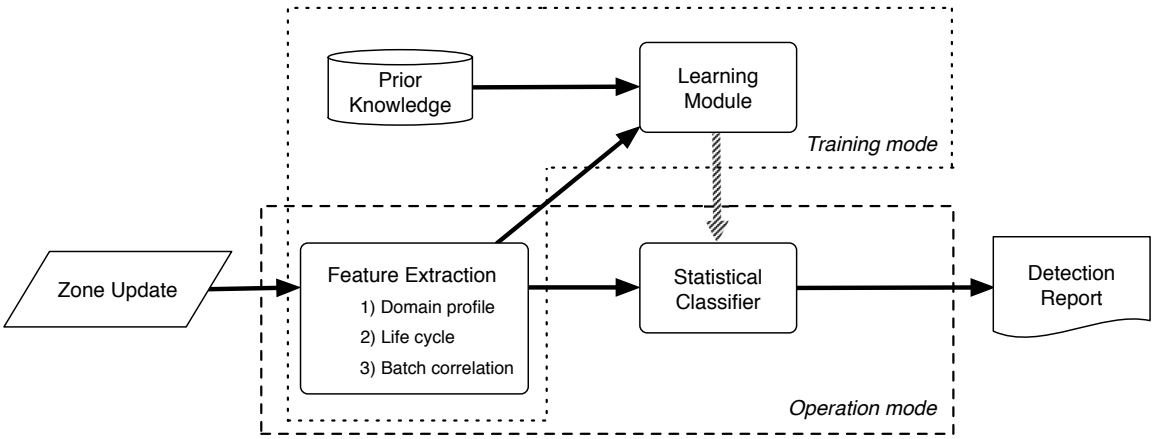


Figure 32: A high-level overview of PREDATOR.

6.2.2 Operation

Figure 32 shows the overview of operation in PREDATOR. We derive the domain registration information from zone update. The *Domain Name Zone Alert* (*DNZA*) files of `.com` zone contains changes to the zone, including (1) the addition of new domains, (2) the removal of existing domains, (3) changes of associated nameservers, and (4) changes of IP addresses of the nameservers. The DNZA files provide a real-time feed of domain registrations. We divide registration activities into five-minute intervals, which we define as *epochs*. The domains registered in the same epoch often share common properties. PREDATOR operates in two modes: an off-line training mode and an on-line operation mode.

Training mode. Based on the input registration information, we extract statistical features. Specifically, for each domain we get three groups of features.

- *Domain profile features.* The first group of statistical features is from the current registration (in Section 6.3.2). The features can be obtained from the public `WHOIS` information or derived from the domain names.

- *Life cycle features.* The second group of features we extract are those based on previous registration history (in Section 6.3.3). The features can be acquired from third-party services such as DomainTools [25], or they are available at registrars and registries.

- *Batch correlation features.* The last set of features examines the domains registered from the same registrar and within the same epoch (in Section 6.3.4). The information is available at registrars or registries.

We use prior knowledge to label a set of known spammer and non-spammer domains. With the labeled domains, the learning module takes the extracted features and uses a supervised learning technique to build a classifier. To achieve the goal of accurate and scalable detection, we design a classification algorithm in Section 6.4.

Operation mode. When a new domain is registered, we extract the corresponding features and input them into the classifier. The classifier assigns a reputation score by aggregating the weights learned in the training mode. If a domain is registered to launch malicious activities (*e.g.*, spam campaigns), we expect to assign a low reputation score. On the other hand, we want to assign a high reputation score if a domain is for legitimate Internet services. If the score is lower than a threshold, we generate a detection report to flag the domain as malicious. Network operators or users can take advantage of the early warning to limit the time that a malicious domain might be used for spam activities; in some cases, they may be able to prevent the attacks from occurring in the first place.

6.3 Feature Extraction

We first dissect an example of spammer domain registrations and further identify key statistical features and explain the motivation behind their selection.

6.3.1 Case Study of Spammer Domain Registrations

We have investigated some of the domain registration patterns in Chapter 6. We further dissect a concrete instance at registrar `Moniker`. Figure 33 shows the counts of `.com` domain registrations from registrar `Moniker` on a day of March, 2012. The x-axis shows the hours in a day (shown in Eastern time, since `Moniker` is a US-based registrar and the company is located on the East coast). The y-axis shows the count of `.com` domain registrations for every five-minute epoch. The red bars indicate the counts of domains appearing on blacklists (including Spamhaus, URI, and a spam trap), while the green

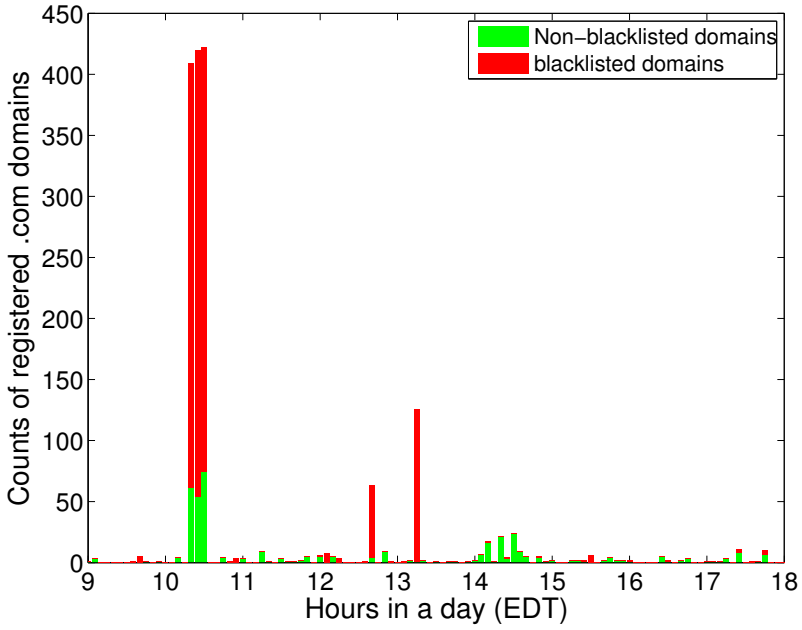


Figure 33: An example of domains registered by `Moniker` every five minutes in a day of March, 2012.

bars are the numbers of domains that are not blacklisted later. The summation of the two types composes the numbers of any `.com` domains registered from `Moniker` every five minutes. These instances help us to gain insight into the characteristics of spammer domain registrations. How many domains are registered together? Are the domains registered for the first time? What do the names look like? How long do the existing blacklists take to block those domains? We briefly explore these questions in the context of this case study.

Registration burst. The registrations of blacklisted domains were in spikes. Table 12 shows the detailed statistics of the five registration spikes with spammer domains in Figure 33. A five-minute epoch may have tens or even hundreds of domains registered for launching spam activities later.

Life-cycle inclination. We examine the life-cycle types of the spammer domains in our instances. Spammer domains in the same spike tend to belong to the same life-cycle type, brand-new (*i.e.*, first-time registration) or retread (*i.e.*, re-registration with some time after previous expiration) as shown in Table 12. The phenomena indicates that miscreants either make up names by themselves or take advantage of gathering expired names previously

Table 12: Example of five-minute epochs with spammer domain registrations from `Moniker`.

Five-minute epoch (EDT)	*Spammer domains*		*All domains*
	Brand-new	*Retread*	
10:15AM–10:20AM	0	**348**	409
10:20AM–10:25AM	0	**366**	420
10:25AM–10:30AM	0	**348**	422
12:35PM–12:40PM	**52**	7	63
1:10PM–1:15PM	**118**	7	126

owned by others.

Name similarity. When we took a close look at the spammer brand-new domains registered in the epoch of 1:10PM–1:15PM, many names appeared similar to each other. Table 13 displays examples of names sharing the same prefixes. The common prefixes are highlighted in bold.

Delay of existing detection. In Table 13 the second column shows how many days have passed until appearance on the blacklists for the sample domains in the epoch of 1:10PM–1:15PM. We check well-known public blacklists, such as Spamhaus and URI, and a spam trap that we operate. The detection delay may take days or even months after domain registrations. In the example shown, although the domains were registered from the same registrar, in the same epoch, and with similar names, the blacklists recorded them at considerably diverse time. The observation implies that those domains may be involved in spam activities for diverse services or targets (how to use in attacks) and at different time (when to use in attacks). On the other hand, the domain registrations exhibit common characteristics.

In the following sections, we expand upon these observations and explore more features. We categorize the features into three groups: domain profile features, life cycle features, and batch correlations features, according to from what sources people can collect the features. We then process the features in two manners, *categorical* features and *continuous* features. High-dimensional categorical features typically arise when representing features that are nominal in nature. For such a feature, there is no notion of order as the taken values are

Table 13: Example of spammer brand-new domains being registered in an epoch (1:10PM–1:15PM EDT) from Moniker.

Domains (highlight common strings)	*Blacklist delay*
askhomelender.com	12 days
askhomelenders.com	6 days
askhomelendersnow.com	51 days
askhomeslender.com	24 days
asklenderhome.com	92 days
asklendershome.com	32 days
asklendertoday.com	5 days
financilsart.com	122 days
financilspro.com	17 days
financilssart.com	9 days
financilss.com	71 days
financilssky.com	7 days
financilsspro.com	18 days
financilsssky.com	19 days
strokecarebeat.com	65 days
strokecaregreen.com	14 days
strokesoft.com	11 days

not related to each other. Low-dimensional continuous features naturally correspond to cardinal type of features. Most of the features ending-up being non-zero are of this type. In the rest of this section, we introduce the key statistical features, the motivation behind their selection, and the way that we process them.

6.3.2 Domain Profile Features

Domain profile features can be determined from the domain name or obtained from the public WHOIS information regarding the current registration.

- Registrar of the domain.
- Authoritative nameservers.
- Nameserver IP addresses and corresponding ASes.
- Registration time, such as daily hour and week day.

Table 14: Summary of PREDATOR features.

Category	*Feature*	*Feature type*
Domain profile features	Registrar	Categorical
	Authoritative nameservers	Categorical
	IP addresses of nameservers	Categorical
	ASes of nameserver IP addresses	Categorical
	Daily hour of registration	Categorical
	Week day of registration	Categorical
	Length of registration period	Continuous
	Trigrams in domain name	Categorical
	Containing "–"	Categorical
	Containing digits	Categorical
	Name length	Continuous
	Ratio of the longest English word	Continuous
	Edit distances to known-bad domains	Continuous
Life cycle features	Life cycle	Categorical
	Dormancy period for re-registration	Continuous
	Previous registrar	Categorical
	Re-registration from the same registrar	Categorical
Batch correlation features	Probability of registration spike	Continuous
	Brand-new proportion	Continuous
	Drop-catch proportion	Continuous
	Retread proportion	Continuous
	Name cohesiveness	Continuous

- Length of registration period, *i.e.*, from "Creation Date" to "Expiration Date" in `WHOIS`.
- Lexical patterns, such as trigrams, name length, and ratio of longest English word.

Registrar (categorical feature). End users need to select the delegated registrars to register domains. Miscreants often choose particular registrars due to the prices or policies from different ones. We find 70% of spammer domains were registered through 10 registrars. We map registrars to a group of binary features. A given dimension is set to 1 if and only if the corresponding registrar hosts the domain. There are 906 such categorical registrar features. Since a domain has a single designated registrar; only one such feature is set to 1 and the others are 0.

Authoritative nameservers (categorical feature). Without authoritative nameservers, people could not resolve the domains in the zone. Although many nameservers belong to major hosting companies that host many legitimate domains, they provide finer-granularity

indication of spammer domain registrations. The nameserver assignment usually happens close to domain registrations and might change with time. We collect authoritative nameservers for the domains within two hours of domain registrations from the zone update files and map them to a set of binary features, where 1 means the nameserver resolves the domain. Since a domain could have multiple nameservers or nameserver changes, more than one attribute could have value 1.

Nameserver IP addresses and ASes (categorical feature). Multiple nameservers can resolve to the same IP address, and different IP addresses can originate from the same Autonomous Systems (ASes). Both IP addresses and ASes indicate underlying hosting infrastructure, which provides a means for identifying portions of the Internet with a higher prevalence of hosting spammer domains. We extract the IP addresses of the nameservers, find the corresponding Autonomous System numbers, and convert them to a group of binary attributes. Similar to the nameserver feature, more than one attribute could have a value of 1. People can actively query to obtain the resolved IP addresses and the ASes. There are data sources, such as those from registrars or registries, that keep information about the historical IP addresses of nameservers. In our experiment, we derive the IP addresses of the nameservers from the zone update files. For a newly registered domain, we obtain the nameservers within two hours of the domain registration, and retrieve all IP addresses ever associated with the name servers within one year before the domain registration. We use the mapping dataset from iPlane to map the IP addresses to Autonomous System numbers [57].

Registration time (categorical feature). Miscreants need to repeatedly register domains for turnover in spam activities; this behavior exhibits certain temporal patterns. In `WHOIS` information, "Creation Date" indicates when a domain was registered. We have equivalent registration time from zone update data and extract the daily hour and week day according to Eastern time zone. The selection of Eastern time zone is not significant, since the purpose is to capture repeated temporal patterns of domain registrations. The time precision is five minutes, which is inherent to our zone update data. The hour of the day corresponds to 24 categorical features (24 hours in a day), and the day of the week maps to seven categorical

features (seven days in a week).

Registration period (continuous feature). A user can register a `.com` domain for 1–10 years. "Expiration Date" in `WHOIS` shows when the domain is about to expire. Longer registration term means a higher fee. We find that 80% of the spammer domains have a one-year initial registration term, presumably since spammers tend to abandon the domains due to blacklisting. We use the years between domain registration and its potential expiration as one feature.

Lexical patterns (categorical/continuous feature). The domain namers often exhibit lexical patterns that are indicative of spammer domains. We compute several categorical and continuous feature to capture the semantics of a domain name. When analyzing the naming patterns, we use only the name in the 2LD for the domain and ignore the trailing ".com".

1. *Trigram* (categorical feature). We use the standard trigram approach to represent a domain name and to examine the character sequence. A domain name is required to start with a letter, end with a letter or digit, and have as interior characters only letters, digits, and hyphen [80]. Since DNS systems treat domain names in a case-insensitive manner, we convert the names into lower case to process. We have a group of $37^3 = 50,653$ binary features that represent all the possible trigrams on the allowed alphabet of 26 letters, 10 digits and the hyphen character. A given feature is set to 1 if and only if the corresponding trigram appears in the domain name; It is otherwise set to 0.

2. *Containing "–"* (categorical feature). We include a binary feature to indicate whether the domain name contains any hyphen.

3. *Containing digits* (categorical feature). We include a binary feature to indicate whether the domain name contains any digits.

4. *Name length* (continuous feature). We compute a feature as the logarithm of the length (number of characters) of the domain name.

5. *Ratio of the longest English word* (continuous feature). Miscreants may either generate

pseudo-random names to avoid conflict with existing domains, or deliberately include readable words in the domain names to attract victim users to click and visit. We match the English words in a dictionary with a domain name to find the longest English word that the name contains. To normalize the feature, we compute the ratio of the length of the longest English word to the whole length of the name.

6. *Edit distances to known-bad domains* (continuous feature). We examine how similar a domain compares with a set of known-bad domains. We derive a set of previously known spammer domains based on the prior knowledge, compute the Levenshtein edit distances with the currently considered domain, and divide these edit distances by the length of the domain name for normalization. We take the five smallest normalized edit distances as features, which have values between 0 and 1 (we have experimented with more numbers of edit distances, and the detection performance remains similar).

6.3.3 Life Cycle Features

Life cycle features are based on previous registration history of a domain. If a domain has appeared in the zone before, we possess registration history, such as previous registrar, registration time and deletion time. Most of such features can be obtained from third-party services such as DomainTools [25], or they are available at registrars and registries. Due to the limitation of our data, we only consider the features regarding the most recent previous registration.

- Life cycle category, *i.e.*, brand-new, drop-catch, or retread.
- Time between previous deletion and re-registration (applicable to retread domains only).
- Previous registrar (applicable to drop-catch and retread domains only).
- Re-registration from the same registrar (applicable to drop-catch and retread domains only).

Life cycle (categorical feature). As mentioned in Section 5.7, we categorize domains as brand-new, drop-catch, and retread, depending on whether the domain is registered for the first time, or how long it passes between previous expiration and new registration. Although the life-cycle type itself may not be a strong indicator whether the domain is registered for spam-related activity, it often provides discriminative information when combining with other features, *e.g.*, the life-cycle types of the other domains registered from the same registrar and around the same time. In total, the life cycle categories map to three binary features and only one of them is set to 1.

Dormancy period for re-registration (continuous feature). The usual re-registration domains stretch to those that expired long time ago. On the other hand, spammers intentionally collect expired domains from publicly released sources [83, 119], which concentrate on recently expired domains. If a domain has appeared in the zone before, we take the logarithm of the seconds between its last expiry and current registration as a feature. Regarding brand-new domains, the feature of dormancy period is not applicable, and we assign a default value (0 in our expertiments).

Previous registrar (categorical feature). The previous registrar offer some insight from where and how spammers gather the expired domain information. We map previous registrars to a group of binary features. Only the feature corresponding to the previous registrar is set to 1, and the others have values 0. We handle brand-new domains whose previous registrar field is not applicable by simply adding a dummy registrar feature.

Re-registration from the same registrar (categorical feature). We add the features to explicitly indicate whether the registrar of a re-registration domain is the same registrar of the previous registration. To deal with brand-new domains which have no previous registrar, we use an additional dummy feature.

6.3.4 Batch Correlation Features

Batch correlation features are extracted from domains within the same ⟨registrar, five-minute epoch⟩ tuple, which we define as a batch. The batch information is initially known

by registrars or registries.

- Probability of registration spike.
- Proportions of different life-cycle domains in the same batch.
- Name cohesiveness (*i.e.*, similarity) in the same batch.

Probability of registration spike (continuous feature). Miscreants usually register domains in large batches, presumably due to cheaper price of bulk registration or management ease. We identify the qualitatively different registration behavior by using the model of compound Poisson process, as derived in Section 5.6. A low-probability batch size from the model indicates an abnormally large registration spike. We use the probability as a feature in our system.

Life cycle proportion (continuous feature). As mentioned before, the registration history can characterize a domain as brand-new, drop-catch, or retread. Miscreants were inclined to register particular life-cycle type of domains in a batch due to how they select the names. We generate three features, each measuring the proportion of different life-cycles for domains in the same batch. These three features sum to 1 by construction.

Name cohesiveness (continuous feature). Spammer domains registered in the same batch usually have the names close to each other, as miscreants use the same strategy or generation algorithm to produce a list of domains. To quantify the cohesiveness of the given domain name with respect to all other domain names in the same batch, we compute the edit distances of the domain to every other domain in the batch. We normalize these edit distances by dividing the length of the domain name to provide a similarity score. We then compute ten features as the logarithm of one plus the number of instances with similarity between $[0, 0.1], [0, 0.2], \ldots, [0, 1.0]$. The last feature is the logarithm of the size of the batch. We use the logarithmic scale to account for the large variability of the batch sizes.

Table 14 summarizes the features in PREDATOR. The first column shows the feature categories, the second column describes the detailed features, and the last column indicates whether the features have categorical or continuous values.

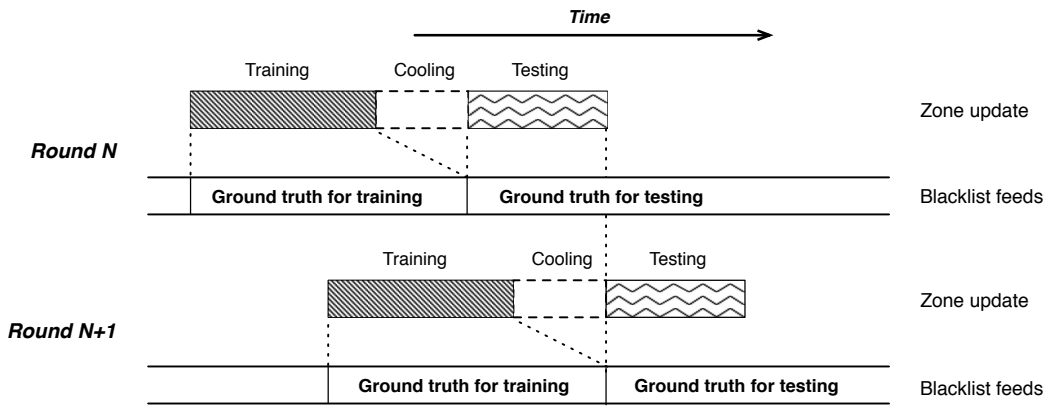


Figure 34: Sliding windows to train detection models.

6.4 Classifier Design

PREDATOR's classifier is responsible for taking the statistical features and deciding whether a domain name has characteristics that make it more likely to be either benign or malicious. This section introduces the supervised learning algorithm that we use, the process of building the detection models, and the derivation of feature importance based on the models.

6.4.1 Supervised Learning: CPM

We want a classifier that can be trained quickly over large sets of data. Our first design requirement is that the learning technique must be computationally inexpensive. Indeed, the number of domain registration in the `.com` zone can reach about two thousand within a five-minute epoch, and the dimension of the feature vectors that we derive is about 10^5. Hence, both training and operation phases must be fast enough to allow continuous operation and periodical re-training. Second, we want the algorithm to behave favorably under severe class imbalance, as we observe that less than 3% of the domains are labeled as involving in spam-related activity. The property of dealing with unbalanced data is particularly important in order to achieve acceptable detection rates at very low false positives.

While linear Support Vector Machines (SVM) [28,102] or comparable linear methods are most often used in such high performance settings, we find their classification accuracies too low for our purposes. Instead, we choose a new algorithm, the Convex Polytope Machine

(CPM) [63]. The CPM can be viewed as an extension of linear classifiers. The CPM maintains an ensemble of linear sub-classifiers and scores incoming instances under all of them. The final decision is based on the maximum of all scores. More formally, suppose $\mathbf{x} \in \mathbb{R}^d$ is an instance of d features, and $\mathbf{w}^1, \ldots, \mathbf{w}^K \in \mathbb{R}^d$ represent the weights of the K sub-classifiers. We derive the score of $\mathbf{x}$ as:

$$f(\mathbf{x}) = \max_{1 \leq k \leq K} \langle \mathbf{x}, \mathbf{w}^k \rangle$$

The prediction score of $f(\mathbf{x})$ reflects how likely a domain is registered for spam-related activity. Geometrically, a CPM defines a convex polytope as the decision boundary to separate the two instance classes. In our application, it appears that this richer, non-linear decision boundary gives us high classification accuracy. Training of a CPM can be efficiently achieved by using the gradient descent based technique, with running times up to five orders of magnitude faster than batch training methods [102]. To verify our selection, we tested SVM [28] and another classic classifier, Random Forest [46], on our dataset, both of which yielded poorer performance (*i.e.*, lower detection rate or longer training time) than CPM.

6.4.2 Building Detection Models

The first step of building the model is to normalize the continuous features. We transform real values into the $[0, 1]$ interval to ensure that continuous features do not overly dominate categorical features. We compute the ranges of each continuous feature and get the max/min values. The normalized feature of v is derived as $(v - v_{min})/(v_{max} - v_{min})$. On occasion, the dynamic range of values taken by v will cover several orders of magnitude. We logarithmically rescale those values prior to normalization.

The categorical features are represented in binary, which do not need additional normalization process.

We adapt a sliding window mechanism for re-training the models and evaluating the detection accuracy close to the real-deployment scenario. We define three windows: training, cooling, and testing. The training window $\triangle T_{train}$ indicates the period during which we collect domain registrations for training. We add a cooling window $\triangle T_{cool}$ to help accumulate the ground truth labeling of the training instances. The testing window $\triangle T_{test}$

makes predictions on the newly registered domains. As shown in Figure 34, suppose at round N the training window starts at time T_N. The model will be constructed at time $\widehat{T}_N = T_N + \triangle T_{train} + \triangle T_{cool}$, with the domains registered during $[T_N, T_N + \triangle T_{train}]$. Since it is usually time-consuming to know whether a domain is indeed involved with spam-related activity, especially based on the observation from blacklists, we use the ground truth collected during the period $[T_N, T_N + \triangle T_{train} + \triangle T_{cool}]$ to label the domains to build the model (in the training mode). In the testing period (corresponding to the operation mode), PREDATOR makes real-time prediction on those domains registered during $[\widehat{T}_N, \widehat{T}_N + \triangle T_{test}]$. The ground truth that we use in the testing period to evaluate the detection accuracy is composed of the domains showing on blacklists from time $\widehat{T}_N$ up to our last collection date of the blacklists.

In the next round, $N + 1$, we move the time windows forward by $\triangle T_{test}$, which makes the new model build at time $\widehat{T}_N + \triangle T_{test}$ (the end of the testing period of round N). The period $\triangle T_{test}$ indicates how frequently we re-train the model. We repeat this process to update the classification model in the training mode, and detect new spammer domains in the operation mode. Operators can customize the window lengths according to different requirements and settings.

6.4.3 Assessing Feature Importance

We can quantify the collective importance of a given subset $S \subset \{1, \ldots, d\}$ of features, for a given CPM model $\{\mathbf{w}^1, \ldots, \mathbf{w}^K\}$ and a dataset of points $\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$. The importance scoring requires a dataset along with the model. Indeed, some model weights have large magnitudes while at the same time the associated features have very low variance, *i.e.*, they are essentially constant. These dimensions are thus not particularly informative and should receive a low score. The scoring method is hence designed to measure the total amount of variation on the score $f(\mathbf{x})$ over the dataset induced by the features S. In the case of a single linear classifier ($K = 1$), this quantity can be measured as:

$$I_S^1 = \sqrt{\operatorname*{Var}_{\mathbf{x}} \left[\sum_{i \in S} \mathbf{w}_i^1 \mathbf{x}_i \right]}$$

Table 15: Summary of data feeds in PREDATOR.

Data	*Collection period*	*Update granularity*
DNZA	March–July 2012	5 minutes
Spam trap	March–October 2012	Real time
URIBL	March–October 2012	60 minutes
Spamhaus DBL	March–October 2012	30 minutes
McAfee SiteAdvisor	One time (June 2013)	N/A

To generalize this measure to the case $K \geq 2$, we note that in the CPM model, a given instance is implicitly assigned to one and only one sub-classifier, *i.e.*, an instance $\mathbf{x}$ is assigned to the sub-classifier with the maximum score. In case of a tie in the maximum, we can always choose the classifier with the smallest index. Hence, for each sub-classifier k, we compute the score I_S^k based on its subset of assigned instances A_k, and combine the scores. Formally, we have:

$$\begin{aligned} I_S &= \sqrt{\frac{|A_1|}{n} I_S^1 + \ldots + \frac{|A_K|}{n} I_S^K} \\ &= \sqrt{\frac{1}{n} \sum_{k=1}^{K} |A_k| \operatorname*{Var}_{\mathbf{x} \in A_k} \left[\sum_{i \in S} \mathbf{w}_i^k \mathbf{x}_i \right]} \end{aligned}$$

where $\mathbf{x} \in A_k$ if and only if $k = \arg\max_{k'} \langle \mathbf{w}^{k'}, \mathbf{x}^i \rangle$. Higher value of I_S indicates that the feature group S contributes more on the decision-making. We demonstrate the feature importance in Section 6.5.4.

6.5 Evaluation

In this section, we report our evaluation results of PREDATOR. First, we describe how we collected and labeled the data. We then present results regarding the detection accuracy by using the registration-based features. Next, we discuss how PREDATOR can be used to complement existing blacklists. Finally, we investigate the importance of the features that we have identified.

6.5.1 Data Set and Labels

Our primary dataset consists of changes made to the `.com` zone for a five-month period, March–July 2012. Verisign operates the `.com` zone under contract to ICANN and keeps

DNZA files to record changes to the zone. We obtain the DNZA files from Verisign (which have five-minute granularity), find the registrations of new domains, and extract the updates of authoritative nameservers and IP addresses. During March–July 2012, we have 12,824,401 newly registered second-level `.com` domains. To label the registered domains as legitimate or malicious, we collected public blacklisting information from March–October 2012 (8 months), including Spamhaus [110], URI [117], and a spam trap that we operate. If a domain appeared on blacklists after registration, we label the domain as being involved in spam-related activities and being registered by miscreants. To obtain benign labels, we queried McAfee SiteAdvisor [103] in June 2013 to find the domains that are reported as benign. Eventually we have about 2% of `.com` domains during our observation period with malicious labels and 4% with benign domain labels. We discuss the prediction results on the unlabeled domains in Section 6.5.2. Table 15 shows the data that we use in our experiments, the collection period, and the update granularity. Most of the data feeds are streamed in a real-time manner, including the DNZA files and blacklists, which allows us to practically evaluate the detection performance of PREDATOR close to the real deployment scenario.

6.5.2 Detection Accuracy

We demonstrate the accuracy of PREDATOR, regarding false positive and detection rate. When a new domain is registered, PREDATOR extracts statistical features, inputs them into the learned classifier, and outputs a reputation score. A lower score indicates a higher likelihood that the domain is registered for malicious activities. By setting different thresholds, we make tradeoffs between false positive rates and detection rates.

We use data from March 2012 to extract known-bad domain set and derive probability models for registration batches and April–July 2012 for our experiments. In Section 6.4.2, we have introduced the sliding window method to build detection models. We tested different window lengths, where better results were yielded from longer training window (*i.e.*, more domains for training), shorter cooling window, and shorter testing window (*i.e.*, more frequent for re-training). Finally, we demonstrate the performance results of PREDATOR by the setting of the training window to 35 days, the cooling window to one day, and the

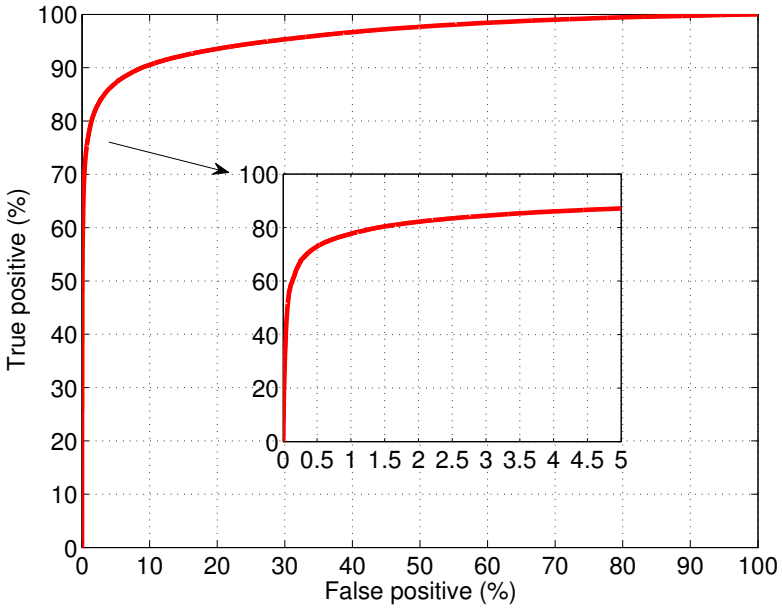


Figure 35: ROC of PREDATOR.

testing window to seven days, which produces good detection accuracy and allows realistic operation.

Figure 35 shows the ROC curve of PREDATOR. The x-axis shows the false positive, which is the percentage of misclassified benign domains to all benign instances. The y-axis shows the detection rate, which accounts for the ratio of correctly predicted spammer domains to all spammer domain samples. The inlay figure shows the ROC curve under the range of false positive between 0% and 5%. PREDATOR achieves good detection rates under low false positives. For example, with 70% detection rate, the false positive is 0.35%. This result is compelling, as it relies on features constructed only from limited information at registration time. As an early-warning mechanism, PREDATOR can effectively detect many domains registered for malicious activities.

The collection of our blacklists contains multiple sources, including Spamhaus, URI, and our spam trap as described above. To explore the sensitivity of our results, we examine PREDATOR's prediction accuracy using individual blacklists as the source of our labels for good and bad domains. For a particular blacklist, we extract malicious domain instances only based on the labels from that blacklist, while the benign labels remain the same as

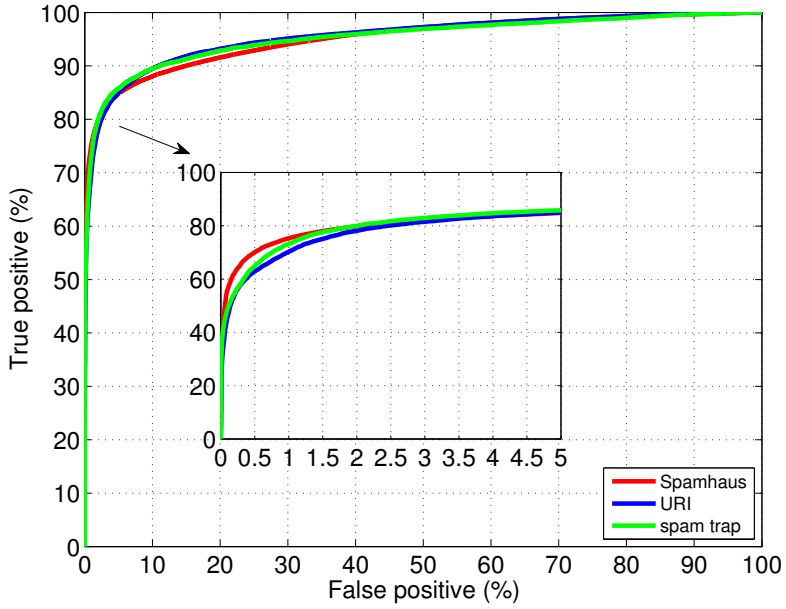


Figure 36: ROC of PREDATOR using different blacklists for labels of legitimate and malicious domains.

being obtained from McAfee SiteAdvisor. Figure 36 shows the accuracies on individual blacklists, including Spamhaus, URI, and spam trap. The trend is similar to that of the overall ROC result in Figure 35 and consistent across blacklists. The effect shows that PREDATOR has consistent performance and can successfully make prediction regarding different blacklists.

Next, we discuss the improvement of PREDATOR over blacklisting methods that are simply based on registrars or nameservers. Suppose we have an *oracle* at registration time to know the future blacklists, *i.e.*, whether a domain would be labeled as malicious later (until October 2012 in our evaluation). A registrar-based blacklisting method is to label a domain according to whether its registrar belongs to a set of *tainted* registrars. We determine the tainted registrar set by adding registrars from those that host malicious domains labeled by the oracle. To make optimal accuracy tradeoff, we add tainted registrars in descending order of the difference value between the true detection rate and the false positive under each registrar. Similarly, we develop a nameserver-granularity blacklisting method with an *oracle* to foresee spammer domains hosted on the namerservers. Figure 37 shows the ROC curves

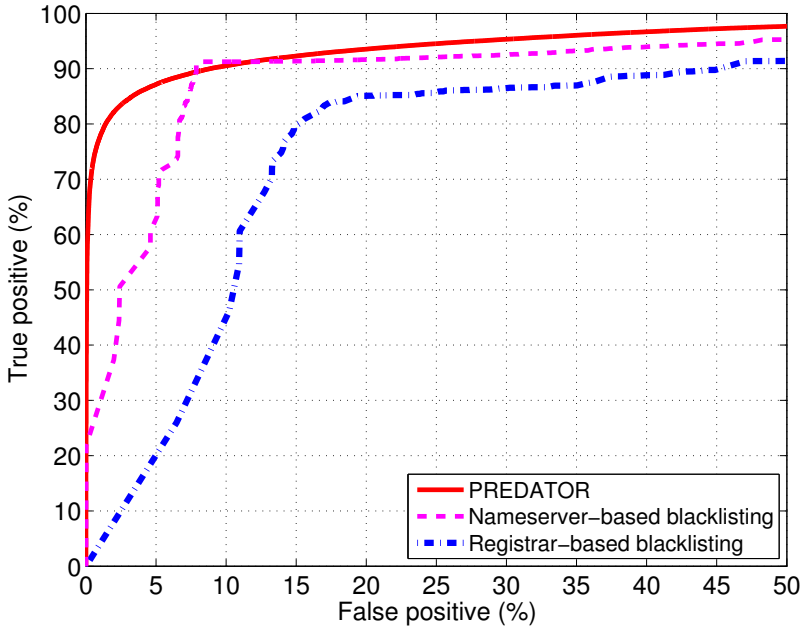


Figure 37: ROC of PREDATOR compared with those of nameserver-granularity and registrar-granularity blacklisting.

from PREDATOR, the nameserver-granularity blacklisting, and the registrar-granularity blacklisting method. The registrar-based blacklisting method has much lower accuracy. In particular, for 70% detection rate, PREDATOR achieves 0.35% false positive; While the nameserver-granularity blacklisting gets 5.20% false positive rate, and the registrar-granularity blacklisting gets 13.27% false positive rate. Under the 70% detection rate, PREDATOR has 15–40 times improvement compared to the straightforward namserver-granularity and registrar-granularity blacklisting methods even with the assistance of an *oracle*.

We project the 0.35% false positive to the entire `.com` zone. Since there are around 80,000 new domains everyday, the daily false positives are about 280 domains. Given that even the known spammer domains were over 1,700 every day, PREDATOR can greatly help to narrow down the set of suspect domains. Since our labeled data just accounts for 6% of all newly registered domains, we run tests to examine how many unlabeled domains are classified as spam-related by using the constructed detection model. With the threshold of 70% detection rate and 0.35% false positive rate, our system reports about 1,300 unlabeled

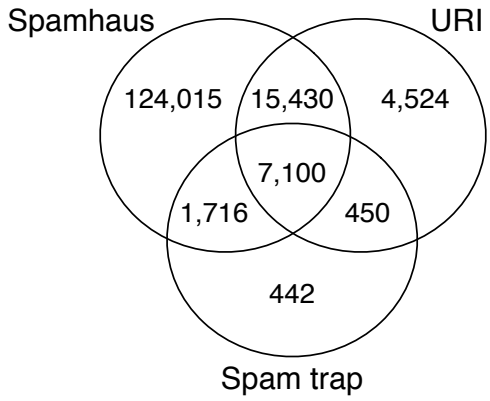


Figure 38: Venn diagram of blacklisted domains from Spamhaus, URIBL, and our spam trap.

domains per day as spam-related, which is on the same magnitude as the labeled spammer domains (1,700 per day). For those domains, we were not able to draw a definitive conclusion. However, we believe many of those domains are more or less involved in some malicious activities.

6.5.3 Comparison to Existing Blacklists

One of the difficulties in evaluating PREDATOR is that we do not know whether the existing blacklists already capture some of the features that PREDATOR uses for early detection, which makes it difficult both to establish any "ground truth" or to compare to existing blacklists. In this section, we investigate the labels from different blacklists in more detail. We also explore when these blacklists listed different domains that PREDATOR detects and find that in many cases PREDATOR can detect malicious domains earlier than existing blacklists.

The first property we examine is *completeness*, which explores how many spammer domains PREDATOR detects compared to other blacklists. Figure 38 shows the intersection of blacklisted domains registered from May-July 2012 (regarding testing period), based on the information from Spamhaus, URI, and our spam trap. Each blacklist has many domains not identified by other sources, which indicates the existing blacklists are not perfect to detect all malicious domains, or they probably even have a certain amount of mislabeled domains. The incompleteness of blacklists makes it challenging to get more

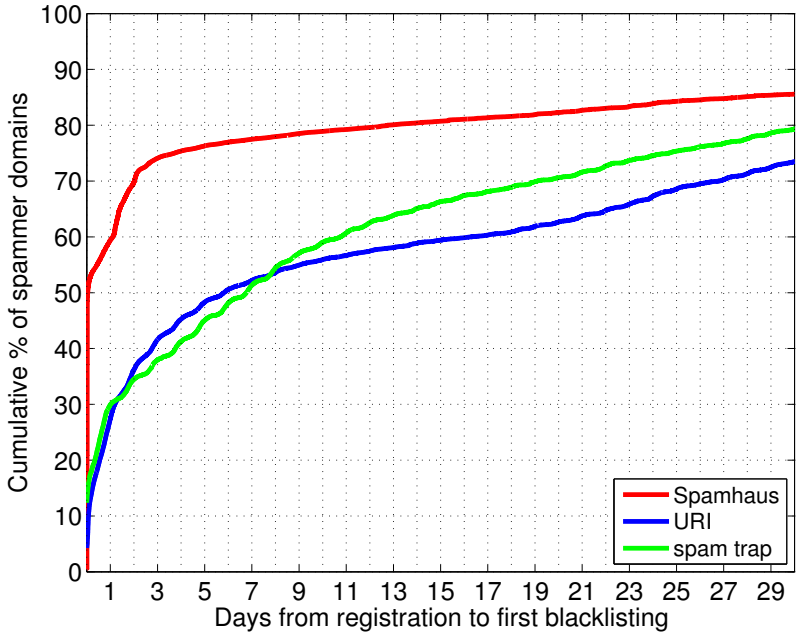


Figure 39: Distribution of days between domain registration and appearance in either our spam trap, URIBL, or the Spamhaus blacklist.

accurate registration-time detection, and also shows PREDATOR could complement current detection methods, given the central observation of domain registrations.

Another important blacklisting characteristics is *delay*, which shows how long after a spammer domain registration, blacklists could make it identified. The detection delay leaves gaps that could not successfully protect users, and attackers take more advantage of their domains for malicious activities. Figure 39 shows the days between domain registration and their first appearance on blacklist. The x-axis indicates how many days after domains' registration they appeared on blacklists. The y-axis shows what percentage of domains started to be blacklisted equal to or less than the days on the x-axis. We observe that both URI and our spam trap take time to identify spammer domains. *e.g.*, around 50% of blacklisted domains manifested after seven days. If the suspect domains are reported early, people could have more time to respond or prevent attacks, which is exactly what PREDATOR aims to achieve. On the other hand, Spmahaus clearly has a mode of time-of-registration blacklisting, where a certain amount of blacklisting happened shortly after domain registrations. We use a threshold of two hours to make a conservative estimate,

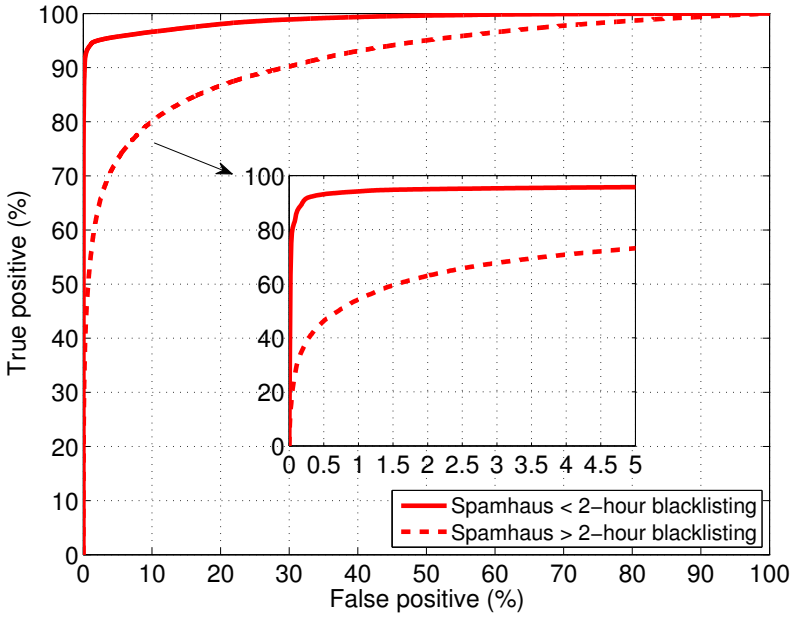


Figure 40: ROC of PREDATOR using domains that Spamhaus blacklisted within the first two hours of registration and after the first two hours of registration for labels.

since the Spamhaus feed that we use updates every half an hour. Regarding domains appearing on Spamhaus less than two hours after their registrations, we define it as time-of-registration blacklisting mode. On the other hand, if the blacklisting happens more than two hours after domain registrations, we define it as time-of-use blacklisting mode. Now we examine PREDATOR predictions on the two blacklisting modes of Spamhaus, and discuss the problems in Spamhaus time-of-registration blacklisting.

To try to infer whether Spamhaus may be using time-of-registration features to blacklist domains, and to explore how the features that Spamhaus uses compare to the features that we use in PREDATOR, we evaluate the accuracy of PREDATOR using the domains that Spamhaus blacklists in the first two hours of registration to label legitimate and malicious domains; we then repeat the analysis for domains that Spamhaus blacklists more than two hours of registration. Figure 40 shows the prediction accuracy of PREDATOR using these two sets of labels. PREDATOR achieves high prediction accuracy when using the domains that were blacklisted within two hours as labels: a 0.35% false positive and detection rate is above 93%. The high accuracy result suggests PREDATOR features already contain most

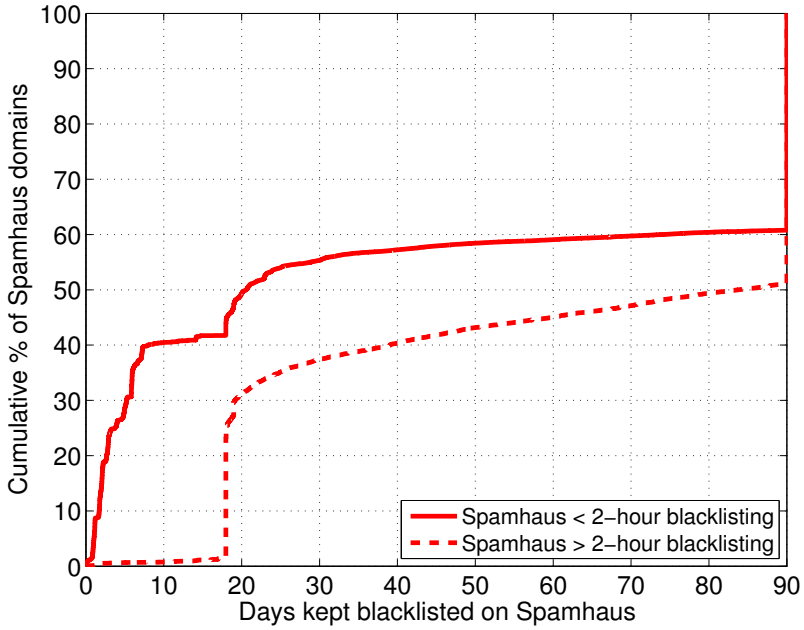


Figure 41: Distribution of days of domains keeping blacklisted on Spamhaus.

of those used by Spamhaus (we cannot confirm this finding because the details of Spamhaus algorithm are not public). PREDATOR also achieves high accuracy using the domains that Spamhaus blacklists more than two hours after registration as the basis for labeling: a 0.35% false positive for a detection rate of about 42%. This result shows PREDATOR can detect some malicious domains much faster than Spamhaus can.

We run two experiments to further examine the effect of Spamhaus time-of-registration blacklisting. First, we check the overlap with URI and the spam trap. Regarding Spamhaus time-of-registration blacklisting, only 9.34% of Spamhaus domains appeared in URI or the spam trap, while 23.77% domains from Spamhaus time-of-use blacklisting had overlap with URI or the spam trap. One possible reason is that part of Spamhaus time-of-registration blacklisting is misclassified instances, which were not reported from other blacklists. Second, we observe Spamhaus has delisting behavior to remove domains on the blacklist. We hypothesize if blacklisted domains get delisted sooner, it is more likely that they are false positives which Spamhaus recognizes to make correction. Given a domain registration, we consider 90 days after its first being blacklisted on Spamhaus B_f. During the period, we find when is the domain last blacklisted on Spamahaus B_l. We use $B_l - B_f$ to measure how

Table 16: Ranking of feature importance in PREDATOR (❶ domain profile category, ❷ life cycle category, and ❸ batch correlation category).

Rank	*Category*	*Feature*	*Score ratio*
1	❶	Authoritative nameservers	100.00%
2	❶	Trigrams in domain name	64.88%
3	❶	IP addresses of nameservers	62.98%
4	❶	Registrar	61.28%
5	❶	ASes of nameserver IP addresses	30.80%
6	❶	Daily hour of registration	30.30%
7	❸	Name cohesiveness	28.98%
8	❶	Week day of registration	22.58%
9	❷	Dormancy period for re-registration	20.58%
10	❷	Re-registration from the same registrar	19.50%
11	❷	Life cycle	18.55%
12	❶	Edit distances to known-bad domains	17.72%
13	❷	Previous registrar	16.50%
14	❸	Brand-new proportion	14.60%
15	❸	Retread proportion	13.71%
16	❸	Drop-catch proportion	12.90%
17	❶	Containing digits	11.25%
18	❶	Name length	10.71%
19	❶	Ratio of the longest English word	9.60%
20	❸	Probability of registration spike	8.66%
21	❶	Containing "–"	8.02%
22	❶	Length of registration period	3.34%

soon the domains is delisted, which has the maximum value of 90 days. Regarding domains registered during May–July 2012, Figure 41 shows the distributions of days being blacklisted on Spamhaus blacklisting less than two hours and more than two hours. We observe that 40% of domains being blacklisted by Spamhaus within two hours after their registrations (red solid curve) have blacklisting period less than ten days, which indicates Spamhaus time-of-registration blacklisting delists sooner and probably has more false positives. When conditioning on both blacklisting less than two hours after registrations and delisting within ten days, only 0.03% of such Spamhaus blacklisted domains showed in other blacklists (URI and the spam trap). The analysis results show the tendency that time-of-registration blacklisting from Spamhaus merely involves simple features and contains a certain amount of misclassified instances.

Table 17: Top 10 ranked features in PREDATOR when applying on Spamhaus < 2-hour blacklisting (❶ domain profile category, ❷ life cycle category, and ❸ batch correlation category).

Rank	*Category*	*Feature*	*Score ratio*
1	❶	Authoritative nameservers	100.0%
2	❶	Registrar	47.72%
3	❶	IP addresses of nameservers	44.26%
4	❶	Trigrams in domain name	37.91%
5	❶	ASes of nameserver IP addresses	24.98%
6	❶	Daily hour of registration	14.23%
7	❷	Re-registration from the same registrar	11.99%
8	❸	Retread proportion	11.48%
9	❷	Life cycle	10.93%
10	❸	Drop-catch proportion	10.70%

6.5.4 Feature Ranking

We have described the scoring method to quantify the feature importance in Section 6.4.3. We use the scores derived from the CPM model to rank the features that we have examined in Section 6.3. The scores represent how much the features can contribute to identify either malicious or benign labels. For easy interpretation, we calculate the score ratio by dividing the score values with the largest one. Table 16 ranks all registration-based features (with the most important feature at top). The marked numbers with black circles in the second column indicates the feature categories, ❶ domain profile category, ❷ life cycle category, and ❸ batch correlation category. Seven of the top 10 features belong to the domain profile category. This result is quite encouraging, since most of these features can be collected with less overhead and from public sources, such as `WHOIS` database.

The feature ranks can help us to "reverse engineer" what features Spamhaus relies on for the time-of-registration blacklisting mode. Table 17 lists the feature importance when we apply the detection algorithm on the domains being blacklisted by Spamhaus within two hours of the domain registrations. We only show the top ten features for simplicity. The ratio difference between the first-ranked feature and the second-ranked one appears larger, which indicates that Spamhaus was inclined to use the feature of authoritative nameservers for detection. To further examine the nameservers, we calculate what percentages of domains on each nameserver were on Spamhaus time-of-registration blacklisting.

When we consider nameservers which have more than 90% of their hosted domains showing on Spamhaus time-of-registration blacklisting, those nameservers account for 86% of all domains appearing on Spamhaus time-of-registration blacklisting. The observation suggests that Spamhaus heavily uses nameservers to make time-of-registration blacklisting decision. If we consider nameservers with 100% blacklisted domains, the percentage of all domains on Spamhaus time-of-registration blacklisting drops to 17%, probably due to the information disparity of the assigned nameservers between Spamhaus and our data.

6.6 Discussion

This section discusses how network operators or end users can take advantage of the prediction output of PREDATOR, and the evasion resistance of the various features.

6.6.1 Actionable Policies

PREDATOR provides early warnings about the potential malicious domains, which can be used from various perspectives or incorporated with other detection mechanisms.

Registries or registrars. Registries and registrars handle and manage all domain registrations under different TLDs, which provides an advantage point to cleanse malicious domains on the Internet. Recently, registrars and registrations have taken more attention and actions to prohibit unlawful use of domain names. For example, LegitScript works closely with registrars, like `eNom`, to seize rogue pharmacy domains [97]. PREDATOR could serve to help registries and registrars to monitor domain registrations and raise early warnings for domains registered with illicit intent.

URL crawling tools. A conventional method to detect malicious Web pages is to crawl the suspect URLs or domains. Since there are a large number of new domains registered everyday, it is expensive to examine all domains and Web sites, especially with continual monitoring. The prediction results from PREDATOR can allow URL crawling tools to give higher priority to crawl domains that are more likely to be malicious. People could reduce the number of domains for inspection, and focus more on suspicious domains.

End users. End users can directly use the prediction results to enable proactive filtering for spam emails or malicious URLs. PREDATOR can serve as the back-end of a query

service to return whether a domain is used for malicious activity, like conventional blacklists (Spamhaus and URI). Although the prediction decision from PREDATOR is not perfect, the increment of visitors to benign domains is orders of magnitude less than that to malicious domains, which makes the misclassification have lower impact on end users.

6.6.2 Evasion Resistance

Miscreants may change registration tactics to evade some of our features. However, we believe it is difficult for attackers to alter all features without considerably reducing the number or rate of their domain registrations. PREDATOR raises the bar for miscreants to profitably use the domains.

Domain profile features. If attackers hope to adapt registration behavior for individual domains, they need to make enough diversity in registrar and nameserver selection, registration time, length of registration period, and naming patterns. Some of the changes may directly increase the cost to purchase domains, such as to force the miscreants to select more expensive registrars or register with longer terms; Others can bring management difficulty, such as to generate lexically distinct names to evade detection.

Life cycle features. Miscreants often re-register expired domains that were previous owned by others. To avoid using domains with registration history, they need to rely more on generation algorithms to find available names, which in turn either causes a higher conflict rate with names already in use or leaves them with less meaningful names. The attempt to evade life cycle features will result in more obvious lexical patterns to detect spammer domains.

Batch correlation features. Attackers can slow down domain registrations and take more time to get a sizable number of domains to use. This accomplishes the purpose of PREDATOR to reduce the rate of miscreants to obtain domains for attacks, or even to prevent attacks from occurring. If miscreants adapt to register domains with lower rates, their attack agility of employing new domains will consequently decrease.

6.7 Summary

Because determining the reputation of DNS domains is critical to defending against many Internet attacks, establishing DNS domain name reputation as quickly as possible after registration time is tremendously important. Whereas existing DNS reputation systems establish domain reputation based on features that are only evident after the domain is in use, PREDATOR can accurately establish domain reputation at the time the domain is registered, which is the first to create a comprehensive reputation system before a DNS domain is ever used in any attack. We developed a detection system for establishing DNS domain reputation at the time a domain name is registered, relying on the much more limited set of features that are evident at registration time. We demonstrate that PREDATOR outperforms straightforward nameserver-granularity and registrar-granularity blacklisting methods, even with the presence of an *oracle* to foresee future malicious usage of the domains, thus making it both faster and more accurate.

CHAPTER VII

CONCLUDING REMARKS

The Internet has provided great convenience to people's daily lives, especially affecting work, education, economics, and entertainment. Email remains a major communication method, along with the bonanza of other messaging services, including Web forum, online chat, and social network. Unfortunately, miscreants target these popular services to make illicit profits. Spam exploits online communication platforms for unsolicited advertising: nowadays approximately 70–90% email on the Internet is spam. To stem the tide of spam, users and network administrators need proactive techniques to achieve early detection, which not only reduces the negative impact on users, but also raises the cost for spam campaigns.

7.1 Summary of Contributions

This thesis has performed a detailed *characterization* of spammer behavior and developed *early detection* systems. In summary, we made the following contributions:

1. *Detecting spammers with network-level features.* We develop a new spam-detection system, *SNARE*, based on lightweight network-level features, without looking at the contents of a message. *SNARE* relies on the intuition that most of spam email is attributed to botnets, which often exhibit unusual sending patterns that differ from those of legitimate email senders. Examples of network-level features include the autonomous system (AS) of the sender, the geographic distance between sender and receiver, the density of email senders in the surrounding IP address space, and the time of day the message is sent. We incorporate those features with a supervised learning technique and achieve comparable accuracy to existing static IP blacklists: about a 70% detection rate for less than a 0.3% false positive rate. *SNARE* improves on prior detection techniques because it automatically infers the reputations of spammers

from their sending behavior, which tends to be more invariant than the contents of the message or the IP address from which they are sending.

2. *Characterizing the initial DNS behavior and hosting of spammer domains.* We monitor the DNS behavior of malicious domains, as identified by appearance in a spam trap, shortly after the domains are registered. Our study includes two perspectives: the DNS infrastructure associated with the domain, as is observable from resource records; and DNS lookup patterns from networks that look up these domains initially. In particular, we analyze DNS information of `.com` and `.net` domains collected from the Verisign top-level domain servers, which allows a *global view* across the Internet, as opposed to a view from any single network. We find spammers prefer a few regions of IP addresses and ASes to host resource records and malicious domains exhibit distinct clusters, in terms of the querying networks. Those features are often evident before any attack even takes place.

3. *Understanding the domain registration behavior of spammers.* We examine the registration process of domains in `.com`, the largest top-level domain, in conjunction with several large blacklist feeds as indicators of spammer domains. Miscreants expose unusual registration behavior, due to economic concerns as well as the ease of management. We develop a model to describe the domain purchase behavior and study domain life cycles. Our findings suggest steps that registries or registrars could use to frustrate the efforts of miscreants to acquire domains in bulk.

4. *Proactive detection of spammer domains at time-of-registration.* We develop PREDATOR, a proactive detection system that can accurately and automatically identify spammer domains at time-of-registration, rather than later at time-of-use. Our work relies on the observation that miscreants register large volumes of domains to maintain revenue and ensure agility, which exhibits characteristic registration behavior. We derive temporal, lexical, and hosting features about domain registrations, and incorporate these features into a fast classification method. PREDATOR can achieve high accuracy with a low false positive rate, and raise early warnings in advance of

domains being actually used in spam activity.

7.2 Future Work

We conclude by suggesting various future research in detecting spam-related activity and protecting the Internet ecosystem.

Spam filtering in IPv6 space. The current spam filtering techniques are based on the IPv4 addresses. On the other hand, the sender reputation in IPv6 space is far less well understood. The lack of knowledge leaves an exploitable opportunity for spammers and causes potential risks to networks deploying IPv6-capable mail servers [10]. An analysis to quantify the abuse in IPv6 and evaluate what mitigation methods remain effective is desired with the increased deployment of IPv6. Although some DNSBLs have started to include IPv6 addresses [121], the large address space makes traditional blacklist hard to keep up. A future research is to adapt the network-level features that we introduce in Chapter 3 and other behavioral models to detect spam in IPv6 space.

Collaborative DNS monitoring. The DNS is a fundamental component of the Internet, and provides a valuable vantage point to detect malicious network activity. The DNS resolution operates in a hierarchical way. Observation at recursive nameservers can only collect information within local networks, such as Notos [4] and EXPOSURE [9]. On the other hand, monitoring DNS traffic above recursive nameservers, *e.g.*, at TLD servers or authoritative nameservers, gives global visibility, but misses query traffic from end users due to cache effects, like our work in Chapter 4 or Kopis [5]. A collaborative system with monitoring at both upper and lower DNS levels can contribute a more complete perspective and enable hierarchical detection.

Underground domain market. We show that around 900 registrars accredited by ICANN contract with registries to manage domain registrations, in our study in Chapter 5. However, there exists a hidden role in domain registration business—resellers. Resellers purchase large quantity of domains from registrars and sell them to individual customers. Many domain resellers are spammer-friendly and provide supply at underground market [72]. Since resellers neither are accredited by ICANN nor directly interact with registries, we lack a

solid understanding of their operations or effect. A further analysis about domain name market, particularly regarding domain resellers, can provide insight to mitigate spam and other malicious activity on the Internet.

REFERENCES

[1] Alperovitch, D., Judge, P., and Krasser, S., "Taxonomy of email reputation systems," in *Proc. of the First International Workshop on Trust and Reputation Management in Massively Distributed Computing Systems (TRAM)*, (Toronto, Canada), June 2007.

[2] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., and Spyropoulos, C. D., "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages," in *Proc. of 23rd ACM SIGIR Conference on Research and Development in Information Retrieval*, (Athens, Greece), July 2000.

[3] Antonakakis, M., Dagon, D., Luo, X., Perdisci, R., Lee, W., and Bellmor, J., "A centralized monitoring infrastructure for improving DNS security," in *13th International Symposium on Recent Advances in Intrusion Detection*, (Ottawa, Ontario, Canada), Sept. 2010.

[4] Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., and Feamster, N., "Building a dynamic reputation system for DNS," in *Proc. 19th USENIX Security Symposium*, (Washington, DC), Aug. 2010.

[5] Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou, N., and Dagon, D., "Detecting malware domains at the upper DNS hierarchy," in *Proc. 20th USENIX Security Symposium*, (San Francisco, CA), Aug. 2011.

[6] Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., and Dagon, D., "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in *Proc. 21st USENIX Security Symposium*, (Bellevue, WA), Aug. 2012.

[7] "APWG phishing activity trends report." `http://docs.apwg.org/reports/apwg_trends_report_q2_2014.pdf`, 2014.

[8] Beverly, R. and Sollins, K., "Exploiting the transport-level characteristics of spam," in *5th Conference on Email and Anti-Spam (CEAS)*, (Mountain View, CA), July 2008.

[9] Bilge, L., Kirda, E., Kruegel, C., and Balduzzi, M., "EXPOSURE: Finding malicious domains using passive DNS analysis," in *18th Annual Network & Distributed System Security Symposium*, (San Diego, CA), Feb. 2011.

[10] Blazquez, A., "Spam over IPv6." `https://labs.ripe.net/Members/blazquez/content-spam-over-ipv6`, 2010.

[11] Boykin, P. and Roychowdhury, V., "Personal email networks: An effective anti-spam tool," *IEEE Computer*, vol. 38, no. 4, pp. 61–68, 2005.

[12] Burges, C., "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[13] Caballero, J., Grier, C., Kreibich, C., and Paxson, V., "Measuring pay-per-install: The commoditization of malware distribution," in *Proc. 20th USENIX Security Symposium*, (San Francisco, CA), Aug. 2011.

[14] Caballero, J., Poosankama, P., Kreibich, C., and Song, D., "Dispatcher: Enabling active botnet infiltration using automatic protocol reverse-engineering," in *Proc. 16th Conference on Computer and Communications Security (CCS)*, (Chicago, IL), Nov. 2009.

[15] Chris Kanich and Christian Kreibich and Kirill Levchenko and Brandon Enright and Vern Paxson and Geoffrey M. Voelker and Stefan Savage,, "Spamalytics: an Empirical Analysis of Spam Marketing Conversion," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, (Arlington, VA), Oct. 2008.

[16] Clark, J. W. and McCoy, D., "There are no free iPads: An analysis of survey scams as a business," in *6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, (Washington, DC), Aug. 2013.

[17] Coull, S. E., White, A. M., Yen, T.-F., Monrose, F., and Reiter, M. K., "Understanding domain registration abuses," in *Proc. 25th International Information Security Conference*, (Brisbane, Australia), Sept. 2010.

[18] Cova, M., Kruegel, C., and Vigna, G., "Detection and analysis of drive-by-download attacks and malicious javascript code," in *Proc. 19th International Conference on World Wide Web*, (Raleigh, NC), Apr. 2010.

[19] Crocker, D., Hansen, T., and Kucherawy, M., *DomainKeys Identified Mail (DKIM) Signatures*. Internet Engineering Task Force, Sept. 2011. RFC 6376.

[20] Danzig, P., Obraczka, K., and Kumar, A., "An analysis of wide-area name server traffic: A study of the internet domain name system," *ACM SIGCOMM Computer Communication Review*, vol. 22, no. 4, p. 292, 1992.

[21] Dasgupta, A., Punera, K., Rao, J. M., and Wang, X., "Impact of spam exposure on user engagement," in *Proc. 21st USENIX Security Symposium*, (Bellevue, WA), Aug. 2012.

[22] Dhamija, R., Tygar, J. D., and Hearst, M., "Why phishing works," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, (Montreal, Canada), Apr. 2006.

[23] "DNSBL resource: Statistics center." `http://stats.dnsbl.com/`, 2008.

[24] "How to snatch an expiring domain." `http://www.mikeindustries.com/blog/archive/2005/03/how-to-snatch-an-expiring-domain`, 2005.

[25] "DomainTools." `http://www.domaintools.com`.

[26] "Backorder price wars: NameJet.com lowers minimum bid on pending delete domains to $59." http://www.thedomains.com/2011/03/17/, 2011.

[27] Drucker, H., Wu, D., and Vapnik, V. N., "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[28] Fan, R., Chang, K., Hsieh, C., Wang, X., and Lin, "LIBLINEAR: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, no. 2008, pp. 1871–1874, 2008.

[29] "FCrDNS lookup testing." http://ipadmin.junkemailfilter.com/rdns.php.

[30] Felegyhazi, M., Kreibich, C., and Paxson, V., "On the potential of proactive domain blacklisting," in *3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats*, (San Jose, CA), Apr. 2010.

[31] Friedman, J. and Popescu, B., "Gradient directed regularization," *Stanford University, Technical Report*, 2003.

[32] Friedman, J. and Popescu, B., "Predictive Learning via Rule Ensembles," *Annals of Applied Statistics*, 2008.

[33] Gao, Y. and Zhao, G., "Knowledge-based information extraction: A case study of recognizing emails of Nigerian frauds," in *Proc. 10th International Conference on Applications of Natural Language to Information Systems*, (Alicante, Spain), June 2005.

[34] "GeoIP API. MaxMind, LLC." http://www.maxmind.com/app/api, 2007.

[35] Gianvecchio, S., Xie, M., Wu, Z., and Wang, H., "Measurement and classification of humans and bots in internet chat," in *Proc. 17th USENIX Security Symposium*, (San Jose, CA), July 2008.

[36] "Godaddy bulk domain search." http://www.godaddy.com/domains/searchbulk.aspx, 2012.

[37] Golbeck, J. and Hendler, J., "Reputation network analysis for email filtering," in *1st Conference on Email and Anti-Spam (CEAS)*, (Mountain View, CA), July 2004.

[38] Gomes, L. H., Castro, F. D. O., Almeida, R. B., Bettencourt, L. M. A., Almeida, V. A. F., and Almeida, J. M., "Improving spam detection based on structural similarity," in *Proc. SRUTI Workshop*, (Cambridge, MA), July 2005.

[39] Goodman, J., Cormack, G., and Heckerman, D., "Spam and the ongoing battle for the inbox," *Communications of the ACM*, vol. 50, no. 2, pp. 24–33, 2007.

[40] Grier, C., Thomas, K., Paxson, V., and Zhang, M., "@spam: The underground on 140 characters or less," in *Proc. 17th Conference on Computer and Communications Security (CCS)*, (Chicago, IL), Oct. 2010.

[41] Gu, G., Perdisci, R., Zhang, J., and Lee, W., "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. 17th USENIX Security Symposium*, (Vancouver, BC, Canada), Aug. 2008.

[42] Gu, G., Porras, P., Yegneswaran, V., Fong, M., and Lee, W., "BotHunter: Detecting malware infection through IDS-driven dialog correlation," in *Proc. 16th USENIX Security Symposium*, (Boston, MA), Aug. 2007.

[43] Hao, S., Feamster, N., and Pandrangi, R., "Monitoring the initial dns behavior of malicious domains," in *Proc. ACM SIGCOMM Internet Measurement Conference*, (Berlin, Germany), Nov. 2011.

[44] Hao, S., Syed, N., Feamster, N., Gray, A., and Krasser, S., "Detecting spammers with SNARE: Spatio-temporal network-level automatic reputation engine," in *Proc. 18th USENIX Security Symposium*, (Montreal, Quebec, Canada), Aug. 2009.

[45] Hao, S., Thomas, M., Paxson, V., Feamster, N., Kreibich, C., Grier, C., and Hollenbeck, S., "Understanding the domain registration behavior of spammers," in *Proc. ACM SIGCOMM Internet Measurement Conference*, (Barcelona, Spain), Oct. 2013.

[46] Ho, T. K., "Random decision forest," in *Proc. 3rd International Conference on Document Analysis and Recognition*, (Montreal, Canada), Aug. 1995.

[47] Hollenbeck, S., *VeriSign Registry Registrar Protocol Version 2.0.0*. Internet Engineering Task Force, Nov. 2003. RFC 3632.

[48] Hollenbeck, S., *Extensible Provisioning Protocol*. Internet Engineering Task Force, Aug. 2009. RFC 5730.

[49] Holz, T., Gorecki, C., Rieck, K., and Freiling, F. C., "Measuring and detecting fast-flux service networks," in *16th Annual Network & Distributed System Security Symposium*, (San Diego, CA), Feb. 2008.

[50] Hulten, G., Penta, A., Seshadrinathan, G., and Mishra, M., "Trends in spam products and methods," in *1st Conference on Email and Anti-Spam (CEAS)*, (Mountain View, CA), July 2004.

[51] Hulton, E. and Goodman, J., "Tutorial on junk email filtering," *Tutorial in the 21st International Conference on Machine Learning (ICML)*, 2004.

[52] "Functional and performance specifications, .com agreement appendix 7." `http://www.icann.org/en/about/agreements/registries/verisign/appendix-07-01mar06-en.htm`, 2006.

[53] "Add grace period limits policy." `http://www.icann.org/en/resources/registries/agp/agp-policy-17dec08-en.htm`, 2008.

[54] "Domain name registration process." `http://whois.icann.org/en/domain-name-registration-process`, 2014.

[55] "Internet world stats." `http://www.internetworldstats.com/stats.htm`, 2013.

[56] "Internet alert registry." http://www.cs.unm.edu/~karlinjf/IAR/, 2007.

[57] " iPlane." http://iplane.cs.washington.edu/data/data.html, 2012.

[58] Johansen, L., Rowell, M., Butler, K., and McDaniel, P., "Email communities of interest," in *4th Conference on Email and Anti-Spam (CEAS)*, (Mountain View, CA), July 2007.

[59] John, J. P., Moshchuk, A., Gribble, S. D., and Krishnamurthy, A., "Studying spamming botnets using Botlab," in *Proc. 6th USENIX NSDI*, (Boston, MA), Apr. 2009.

[60] Jung, J. and Sit, E., "An empirical study of spam traffic and the use of DNS black lists," in *Proc. ACM SIGCOMM Internet Measurement Conference*, (Sicily, Italy), Oct. 2004.

[61] Jung, J., Sit, E., Balakrishnan, H., and Morris, R., "DNS performance and the effectiveness of caching," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, (San Fransisco, CA), Nov. 2001.

[62] Kanich, C., Weaver, N., McCoy, D., Halvorson, T., Kreibich, C., Levchenko, K., Paxson, V., Voelker, G. M., and Savage, S., "Show me the money: Characterizing spam-advertised revenue," in *Proc. 20th USENIX Security Symposium*, (San Francisco, CA), Aug. 2011.

[63] Kantchelian, A., Tschantz, M. C., Ling Huang, P. L. B., Joseph, A. D., and Tygar, J. D., "Large-margin convex polytope machine," in *Advances in Neural Information Processing Systems (NIPS)*, (Quebec, Canada), Dec. 2014.

[64] Karlin, J., Forrest, S., and Rexford, J., "Autonomous security for autonomous systems," *Computer Networks*, vol. 52, no. 15, pp. 2908–2923, 2008.

[65] Klensin, J., *Simple Mail Transfer Protocol*. Internet Engineering Task Force, Oct. 2008. RFC 5321.

[66] Konte, M., Feamster, N., and Jung, J., "Dynamics of online scam hosting infrastructure," in *Passive & Active Measurement (PAM)*, (Seoul, South Korea), Apr. 2009.

[67] Konte, M., Feamster, N., and Jung, J., "Fast flux service networks: Dynamics and roles in hosting online scams," in *Passive & Active Measurement (PAM)*, (Atlanta, GA), Apr. 2011.

[68] Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G. M., Paxson, V., and Savage, S., "Spamcraft: An inside look at spam campaign orchestration," in *2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats*, (Boston, MA), Apr. 2009.

[69] Lam, H. and Yeung, D., "A learning approach to spam detection based on social networks," in *4th Conference on Email and Anti-Spam (CEAS)*, (Mountain View, CA), July 2007.

[70] Leontiadis, N., Moore, T., and Christin, N., "Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade," in *Proc. 20th USENIX Security Symposium*, (San Francisco, CA), Aug. 2011.

[71] Lepinski, M., *BGPSEC Protocol Specification*. Internet Engineering Task Force, Oct. 2014. Internet-Draft.

[72] Levchenko, K., Chachra, N., Enright, B., Felegyhazi, M., Grier, C., Halvorson, T., Kanich, C., Kreibich, C., Liu, H., McCoy, D., Pitsillidis, A., Weaver, N., Paxson, V., Voelker, G. M., and Savage, S., "Click trajectories: End-to-end analysis of the spam value chain," in *Proc. IEEE Symposium on Security and Privacy*, (Oakland, CA), May 2011.

[73] Levine, J., *DNS Blacklists and Whitelists*, Feb. 2010. RFC 5782.

[74] Liu, H., Levchenko, K., Felegyhazi, M., Kreibich, C., Maier, G., Voelker, G. M., and Savage, S., "On the effects of registrar-level intervention," in *4th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, (Boston, MA), Mar. 2011.

[75] Lychev, R., Goldberg, S., and Schapira, M., "BGP security in partial deployment: Is the juice worth the squeeze?," in *Proc. ACM SIGCOMM*, (Hong Kong, China), Aug. 2013.

[76] "MAAWG email metrics program: The network operators perspective, report 15." `http://www.maawg.org/sites/maawg/files/news/MAAWG_2011_Q1Q2Q3_Metrics_Report_15.pdf`, 2011.

[77] McCoy, D., Pitsillidis, A., Jordan, G., Weaver, N., Kreibich, C., Krebs, B., Voelker, G. M., Savage, S., and Levchenko, K., "PharmaLeaks: Understanding the business of online pharmaceutical affiliate programs," in *Proc. 21st USENIX Security Symposium*, (Bellevue, WA), Aug. 2012.

[78] Meyer, T. A. and Whateley, B., "SpamBayes: Effective open-source, Bayesian based, email classification system," in *Proc. of 1st Conference on Email and Anti-Spam*, (Mountain View, CA), July 2004.

[79] "Microsoft releases new threat data on Rustock." `http://blogs.microsoft.com/blog/microsoft-releases-new-threat-data-on-rustock/`, 2011.

[80] Mockapetris, P. V., *Domain Names – Concepts and Facilities*. Internet Engineering Task Force, Nov. 1987. RFC 1034.

[81] Mockapetris, P. V., *Domain Names – Implementation and Specification*. Internet Engineering Task Force, Nov. 1987. RFC 1035.

[82] "Moniker bulk registration." `https://www.moniker.com/bulkdomainname.jsp`, 2012.

[83] "NameJet domain name aftermarket." `http://www.namejet.com/pages/downloads.aspx`, 2012.

[84] Niu, Y., Wang, Y.-M., Chen, H., Ma, M., and Hsu, F., "A quantitative study of forum spamming using context-based analysis," in *18th Annual Network & Distributed System Security Symposium*, (San Diego, CA), Feb. 2011.

[85] Pandrangi, R., Feamster, N. G., and Hao, S., "Systems and methods for identifying malicious domains using internet-wide dns lookup patterns." U.S. Patent 8,713,676 B2, Apr. 2014.

[86] Park, Y., Jones, J., McCoy, D., Shi, E., and Jakobsson, M., "Scambaiter: Understanding targeted Nigerian scams on craigslist," in *21st Annual Network & Distributed System Security Symposium*, (San Diego, CA), Feb. 2014.

[87] Pathak, A., Hu, Y. C., and Mao, Z. M., "Peeking into spammer behavior from a unique vantage point," in *1st USENIX Workshop on Large-Scale Exploits and Emergent Threats*, (San Francisco, CA), Apr. 2008.

[88] Paxson, V. and Floyd, S., "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 2005.

[89] "PlanetLab." `http://www.planet-lab.org/`.

[90] Quinlan, J., "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[91] Rahman, M. S., Huang, T.-K., Madhyastha, H. V., and Faloutsos, M., "Efficient and scalable socware detection in online social networks," in *Proc. 21st USENIX Security Symposium*, (Bellevue, WA), Aug. 2012.

[92] Rajab, M. A., Monrose, F., and Terzis, A., "A multifaceted approach to understanding the botnet phenomenon," in *Proc. ACM SIGCOMM Internet Measurement Conference*, (San Diego, CA, USA), Oct. 2007.

[93] Ramachandran, A. and Feamster, N., "Understanding the network-level behavior of spammers," in *Proc. ACM SIGCOMM*, (Pisa, Italy), Sept. 2006.

[94] Ramachandran, A., Feamster, N., and Vempala, S., "Filtering spam with behavioral blacklisting," in *Proc. 14th Conference on Computer and Communications Security (CCS)*, (Alexandria, VA), 2007.

[95] Ramachandran, A., Dagon, D., and Feamster, N., "Can DNSBLs keep up with bots?," in *3rd Conference on Email and Anti-Spam (CEAS)*, (Mountain View, CA), July 2006.

[96] Rao, J. M. and Reiley, D. H., "The economics of spam," *Journal of Economic Perspectives*, vol. 26, no. 3, pp. 87–110, 2012.

[97] "Rogues and registrars: Top 10 list." `http://blog.legitscript.com/2012/10/rogues-registrars-top-10-list-october-2012`, 2012.

[98] "List of ICANN accredited registrars." `http://www.icann.org/registrar-reports/accredited-list.html`, 2013.

[99] "Root zone database." `http://www.iana.org/domains/root/db`, 2012.

[100] Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E., "A Bayesian approach to filtering junk e-mail," in *AAAI Workshop on Learning for Text Categorization*, (Madison, Wisconsin), July 1998.

[101] Samosseiko, D., "The partnerka—what is it, and why should you care?," in *Proc. of Virus Bulletin Conference*, (Geneva, Switzerland), Sept. 2009.

[102] Shalev-Shwartz, S., Singer, Y., and Srebro, N., "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, (New York, NY, USA), pp. 807–814, ACM, 2007.

[103] "McAfee SiteAdvisor." `https://www.siteadvisor.com/`.

[104] "SORBS: Spam and open relay blocking system." `http://www.au.sorbs.net/`.

[105] "Email spam record activity." `http://www.guardian.co.uk/technology/2011/jan/10/email-spam-record-activity`, 2011.

[106] "Cyber attack that sent 750k malicious emails traced to hacked refrigerator, TVs and home routers." `http://kdvr.com/2014/01/20/cyber-attack-traced-to-hacked-refrigerator-tvs-and-home-routers/`, 2014.

[107] "Massive spam campaign: Still attempting to spread malware." `http://blog.appriver.com/2014/02/massive-spam-campaign/`, 2014.

[108] "Spam botnets: The fall of Grum and the rise of Festi." `http://www.spamhaus.org/news/article/685/`, 2014.

[109] "SpamCop." `http://www.spamcop.net/`.

[110] "Spamhaus." `http://www.spamhaus.org/`.

[111] Springael, J. and Nieuwenhuyse, I. V., "A lost sales inventory model with a compound Poisson demand pattern," *Working paper*, July 2005.

[112] Stein, T., Chen, E., and Mangla, K., "Facebook immune system," in *Proc. 4th Workshop on Social Network Systems*, (Salzburg, Austria), Apr. 2011.

[113] "Symantec intelligence report." `http://www.symantec.com/theme.jsp?themeid=state_of_spam`, 2012.

[114] "Symantec Internet security threat report." `http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_2011_21239364.en-us.pdf`, 2012.

[115] "Symantec Internet security threat report." `http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v18_2012_21291018.en-us.pdf`, 2013.

[116] Thomas, K., Grier, C., Ma, J., Paxson, V., and Song, D., "Design and evaluation of a real-time URL spam filtering service," in *Proc. IEEE Symposium on Security and Privacy*, (Oakland, CA), May 2011.

[117] "URI." http://www.uribl.com/.

[118] "Verisign who was service." http://www.icann.org/en/registries/rsep/verisign-whowas-01jul09-en.pdf, 2009.

[119] "Verisign domain countdown." http://domaincountdown.verisignlabs.com, 2011.

[120] "Verisign domain report: The domain name industry brief." http://www.verisigninc.com/en_US/innovation/dnib/index.xhtml, 2012.

[121] "Virbl." http://virbl.bit.nl/.

[122] Wang, D. Y., Savage, S., and Voelker, G. M., "Cloak and dagger: Dynamics of web search cloaking," in *Proc. 18th Conference on Computer and Communications Security (CCS)*, (Chicago, IL), Oct. 2011.

[123] Wilson, T., "Researchers link Storm botnet to illegal pharmaceutical sales." http://www.darkreading.com/security/security-management/211201114/index.html, 2008.

[124] Wu, B. and Davison, B. D., "Identifying link farm spam pages," in *Proc. 14th International Wordl Wide Web Conference (WWW)*, (Chiba, Japan), May 2005.

[125] Wu, B. and Davison, B. D., "Detecting semantic cloaking on the web," in *Proc. 15th International Conference on World Wide Web*, (Edinburgh, Scotland), May 2006.

[126] Xie, Y., Yu, F., , Achan, K., Panigrahy, R., Hulten, G., and Osipkov, I., "Spamming bots: Signatures and characteristics," in *Proc. ACM SIGCOMM*, (Seattle, WA), Aug. 2008.

[127] Zhang, G. G. J. and Lee, W., "BotSniffer: Detecting botnet command and control channels in network traffic," in *16th Annual Network & Distributed System Security Symposium*, (San Diego, CA), Feb. 2008.

[128] Zhang, L., Zhu, J., and Yao, T., "An evaluation of statistical spam filtering techniques," *ACM Transactions on Asian Language Information Processing*, vol. 3, no. 4, pp. 243–269, 2004.

[129] Zhao, X., Pei, D., Wang, L., Massey, D., Mankin, A., Wu, S. F., and Zhang, L., "An analysis of BGP multiple origin AS (MOAS) conflicts," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (IMW)*, 2001.

[130] Zupan, J., *Clustering of Large Data Sets*. Research Studies Press, 1982.