Email Typosquatting

Janos Szurdi
Carnegie Mellon University
jszurdi@andrew.cmu.edu

Nicolas Christin
Carnegie Mellon University
nicolasc@andrew.cmu.edu

ABSTRACT

While website domain typosquatting is highly annoying for legitimate domain operators, research has found that it relatively rarely presents a great risk to individual users. However, any application (e.g., email, ftp,...) relying on the domain name system for name resolution is equally vulnerable to domain typosquatting, and consequences may be more dire than with website typosquatting.

This paper presents the first in-depth measurement study of email typosquatting. Working in concert with our IRB, we registered 76 typosquatting domain names to study a wide variety of user mistakes, while minimizing the amount of personal information exposed to us. In the span of over seven months, we received millions of emails at our registered domains. While most of these emails are spam, we infer, from our measurements, that every year, three of our domains should receive approximately 3,585 "legitimate" emails meant for somebody else. Worse, we find, by examining a small sample of all emails, that these emails may contain sensitive information (e.g., visa documents or medical records).

We then project from our measurements that 1,211 typosquatting domains registered by unknown entities receive in the vicinity of 800,000 emails a year. Furthermore, we find that millions of registered typosquatting domains have MX records pointing to only a handful of mail servers. However, a second experiment in which we send "honey emails" to typosquatting domains only shows very limited evidence of attempts at credential theft (despite some emails being read), meaning that the threat, for now, appears to remain theoretical.

CCS CONCEPTS

• **Security and privacy → Human and societal aspects of security and privacy**; *Network security*; • **Networks** → *Network measurement*;

KEYWORDS

Domain name, Typosquatting, Abuse, Measurement, Ethics

ACM Reference Format:
Janos Szurdi and Nicolas Christin. 2017. Email Typosquatting. In *Proceedings of IMC '17: Internet Measurement Conference* . ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3131365.3131399

IMC '17: Internet Measurement Conference , November 1–3, 2017, London, United Kingdom
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5118-8/17/11…$15.00
https://doi.org/10.1145/3131365.3131399

1 INTRODUCTION

Domain typosquatting is the act of registering a domain name very similar to an existing, legitimate, domain, in an effort to capture some of the traffic destined for the original domain. Domain typosquatting exploits the propensity of users to make typographical errors when typing domain names—as opposed to clicking on links—and is frequently used for financial profit. For instance, somebody registering `googe.com` would immediately receive large amounts of traffic meant for `google.com`. That traffic could then in turn be monetized, by showing ads or setting up drive-by-downloads. Domain typosquatting has been shown to be profitable [18, 24], while requiring no technical skill.

In some jurisdictions, domain typosquatting is considered illegal, and may trigger trademark infringement cases.[1] In 1999, ICANN, the authority which regulates domain names on the Internet, created the Uniform Domain Name Dispute Resolution Policy (UDRP) as a solution for trademark owners to claim cybersquatting or typosquatting domain names [21].

Thus far, most of the studies in the related literature have solely focused on web typosquatting, that is, domain typosquatting used to illicitly acquire "page views." However, domain typosquatting can be equally used with other target applications—`ssh`, `ftp`, email, and so forth.

This paper is the first in-depth study to focus on *email typosquatting*, in which miscreants could register domain names mimicking those of large email providers to capture emails. Even though typing mistakes may be fairly rare, typosquatting a large email provider (e.g., `gmail.com`) could remain a profitable endeavor by virtue of the number of emails passing through the service. Indeed, while most emails illicitly received would be of limited use to the attacker, some could contain sensitive information that could yield large payoffs for the attacker, and cause considerable losses to the victim.

We put this hypothesis to the test in this paper. Specifically, we register 76 email typosquatting domains, collect data from these domains for more than seven months (June 4, 2016–January 15, 2017), and—working in concert with our Internal Review Board (IRB)—design a protocol to process the emails we receive to determine the potential harm email domain typosquatting might inflict on users, as well as its potential benefits to attackers (Section 4). We discover that a number of actors already have the infrastructure necessary for bulk email domain typosquatting (Section 5). Extrapolating from our observations through regression analysis (Section 6), we find that setting up the necessary infrastructure costs attackers only in the order of a couple of cents per email, and that they can expect to receive hundreds of thousands of emails over a few months. However, by actively sending "honey emails" containing credentials, we discover, that even though a lot of these

[1] See, e.g., in the U.S., the Federal Trademark Dilution Act, or FTDA, and the Anticybersquatting Consumer Protection Act, ACPA.

emails are accepted, they are not actually read (Section 7), meaning that email typosquatting does not appear, for now, to be monetized.

2 RELATED WORK

Our work broadly inscribes itself in the general line of research in online crime measurements, which has been an extremely active area of research over the past fifteen years or so. Rather than providing a comprehensive overview of the entire area, we refer the reader to recent survey papers in the field [12, 28]. Here, we focus instead on the narrower body of work that has attempted to characterize the prevalence of domain typosquatting, and its economic effects, and outline how our work builds on more than fourteen years of research in the community.

Most typosquatting papers have focused on web typosquatting, which targets users who make a mistake while typing an URL in their browser. In 2003, Edelman undertook the first case study of one typosquatter who registered, at the time, thousands of domains [17]. Subsequently, a number of efforts [13–15, 31] proposed methods to detect typosquatting domains targeting popular websites, as ranked by the Alexa service [1], and to differentiate legitimate domains from typosquatting domains [27]. Some of these studies suggest that monetization is achieved through domain parking – the act of monetizing otherwise empty web pages with advertisements.

Moore and Edelman [24] discussed monetization of typosquatting, and showed that miscreants might be relying on Google AdWords to select which typosquatting domains to register. Along the same lines, Agten et al. [11] provided a longitudinal study of monetization strategies of typosquatting targeting Alexa's top 500 domains. More recently, Khan et al. [22] quantified the harm of typosquatting caused to *users*, and found that a typical user loses 1.3 seconds on average when visiting a typosquatting domain.

Different from this entire body of work, we broaden the scope of investigation to email typosquatting, which, from a technical standpoint shares many similarities with web typosquatting (low barrier to entry, low sophistication), but whose monetization strategies ought to be completely different—whereas web typosquatting primarily profits from advertisements, through "parking pages [30]," email typosquatting is likely to benefit from capturing credentials or sensitive information.

To the best of our knowledge, only one white paper looked at domain typosquatting beyond web typosquatting [19]. The authors registered domains that were similar to existing *subdomains*, with the exception of a missing dot—e.g., `caibm.com` as opposed to `ca.ibm.com`. They claim to have collected 120,000 mis-directed emails over six months, but do not report on the number of domains they registered, and do not discuss whether they filtered out spam. Our work attempts to provide a far more detailed picture of email typosquatting in the wild; in particular, we will observe that filtering out spam email is a crucial step in providing credible measurements of the attack's impact. We also investigate whether typosquatters *act upon* emails they receive.

3 TERMINOLOGY

Typosquatting actually involves a number of different concepts, which we discuss here.

Distance metrics. We use two metrics to characterize the distance between various domain names. The *Damerau-Levenshtein distance* [16] is the minimum number of operations (deletion, addition, substitution, or transposition of two neighboring characters). Papers on typosquatting often rely on Damerau-Levenshtein distance of one ("DL-1") to detect typosquatting domains. Moore and Edelman define the *fat-finger distance* as "the minimum number of insertions, deletions, substitutions or transpositions using letters adjacent on a QWERTY keyboard to transform one string into another." [24] A fat-finger distance of one (FF-1) implies a DL-1 distance. The *visual distance* measures how different the mistyped character *looks* compared to the original character. We use a set of heuristic rules to compute the visual distance, which incorporate how confusing alphabet letters with numbers (e.g., "o" and "O," "1" and "l") is more likely to happen than confusing two (different) letters or numbers.

Typosquatting domains. The *target domain name* refers to any domain name targeted by typosquatters. Previous work on web typosquatting usually relies on Alexa rankings [1] to identify target domains.

We adopt Szurdi et al.'s taxonomy [27] to clearly differentiate lexically close domains from true typosquatting domain names. *Generated typo domains* ("gtypos") are "domain names which are lexically similar (e.g. at DL-1) to some set of target domains." *Candidate typo domains* ("ctypos") are "the subset of registered domains within the gtypo set which have been registered." Finally, *typosquatting domains* are candidate typo domains that "(i) [were] registered to benefit from traffic intended for a target domain," and "(ii) that [are] the property of a different entity."

Misdirected email taxonomy. Typosquatting of email domains allows an attacker to capture a number of different emails. First, *receiver typo* emails are simply sent to the wrong address by the sender mistyping the recipient's email address. We only focus on typos in the domain name, and leave the issue of typos in the recipient name to future work.[2]

We also consider *reflection typo* emails. Those emails are the result of users mistyping their email address when registering for an online service. As a consequence, emails from the service are subsequently sent to the wrong address. While the harm caused would be likely negligible in the case of an online raffle, providing the wrong address to a financial services company might lead to leaks of confidential or sensitive information.

Last, we capture a completely different type of error with *SMTP typo* emails, which result from a user mistyping their SMTP settings in their email client. This type of error is pernicious, as *all* emails sent by the victim may be intercepted until the typo is fixed.

4 IN THE SHOES OF A TYPOSQUATTER

In this section we describe a seven-month experiment (June 4, 2016–January 15, 2017), during which we acted as email typosquatters ourselves, in an effort to gain insights into whether email typosquatting could be a potential problem or not. The idea is simple: by registering typosquatting domains, we can simply count the number of emails these domains—which we absolutely did not advertise or otherwise use, to avoid measurement confounds—receive, and infer whether email typos occur frequently or not, and if so, which kind

[2] For instance, we consider `alice@gmial.com`, but not `aliec@gmail.com`.

of typos seem more prevalent than others. This analysis will later be useful in attempting to derive more general projections, beyond the set of domains we registered, on the potential magnitude of the problem overall.

Because we are ultimately acting as attackers, our experimental setup is driven by ethical considerations. We start with a discussion of these ethical objectives, before turning to how our collection methodology attempts to fulfill these objectives. We then analyze the results of our data collection.

4.1 Ethical challenges and how to address them

Registering a set of typosquatting domain names ourselves provides a very precise view of the type of information users may accidentally leak. At the same time, 1) we need to tread carefully with possible trademark infringement, and, even more importantly, 2) we can potentially receive users' personal information.

Both issues are very serious and led us to design our protocol with the collaboration of our university's Internal Review Board (IRB), in an effort to minimize the risk to users, and to ourselves. The protocol was approved by our IRB, and our sponsor's IRB, before we started our experiments.

The trademark infringement part—which actually does not impact any users but us—was relatively quickly settled. We agreed to surrender any domain we registered to the legitimate owner of a trademark it could potentially infringe upon simple request. To date, we have not received any such requests.

While we elected to keep emails accidentally sent to our domains to carry out deeper analyses than could be done by simply keeping headers, we take three measures to protect the users who sent these emails. First, our storage infrastructure consists of a hardened server accessible only from our university network. Second, we automatically remove sensitive information using regular expression matching prior to storage. Finally, we encrypt all emails prior to storing them, using an encryption key kept *separately* from the server (i.e., on removable storage). To result in potential harm, accidental disclosure of the contents of the server would need to be accompanied by a leakage of our encryption key.

Even though our IRB protocol allows us to look at the content of the emails we receive, provided that we do our best effort to automatically sanitize personal identifiers prior to doing so, we wanted to minimize as much as possible such interactions. Initially, we were hoping to be able to derive the content of these emails purely programmatically—i.e., inferring the presence of leaks from regular expression matching on the body, classification of attachment names, etc. However, we received an enormous amount of spam email, which made it important to fine-tune and evaluate the spam filtering system we used. We eventually settled on looking at a small sample of 103 emails (out of several millions we received overall) that were classified as non-spam to evaluate the performance of our spam classifier, which is absolutely crucial to the rest of our analysis due to the imbalance of our dataset.

In other words, we adopted a utilitarian ethics view—while it is undesirable (but permitted by our IRB-approved protocol) to look at some of these email contents, we were satisfied that the small minority of emails we were manually analyzing would 1) not result in any risk to the users who sent (or were meant to receive) these emails, while 2) giving us stronger confidence in our results. We re-emphasize that potentially sensitive information (e.g., credit card numbers) was automatically scrubbed *prior* to our looking at these 103 emails.

4.2 Collection methodology

We next turn to discussing how we selected a set of domains to register, before delving into the details of our collection infrastructure. We then explain how we post-processed the data we acquired by presenting the layered filtering system we built to remove spam from our corpus.

4.2.1 Domain registration. When deciding on which domain names to register, we had a number of constraints to satisfy, and three main objectives in mind.

Constraints. Our first constraint is budgetary. While registering individual domains is reasonably cheap, in the order of $8–$20 per year depending on the registrar and top-level-domain being used, it is potentially time-consuming, and we have to limit ourselves to at most a couple of hundred domains. Our second constraint, which is far more serious, is that of availability. Unfortunately a number of the most interesting typo domains are already registered (either by the trademark owners themselves, or by typosquatters), so that we were forced to choose from what is available. However, the set of gtypos is a powerset of the set of target domains. In particular, for the top 10,000 domains according to Alexa rankings, there are millions of gtypos. Even though hundreds of thousands are already registered, we are still able to select a few dozen typosquatting domains that can hopefully produce representative outcomes.

Objectives. When we undertook this study, we had absolutely no idea of the amount of emails we would receive. Our first goal was thus to find typo domains that could be trusted to provide a representative, and measurable signal, if anything was to be measured. Our second goal was to compare different DL-1 typing mistakes (e.g., deletion and substitution), to be able to reason about respective impact of such mistakes. Third, we wanted to register a corpus of domains that would allow us to measure the different kinds of typos (receiver, SMTP, reflection) we had identified.

Strategy. To maximize the probability of receiving emails, we aimed to register typo domains targeting some of the most popular domains. To that effect, we selected target domains with a small Alexa rank in the email category (i.e., popular domains for email). To prune down the list of domains we register, most of the typo domains we generated have a fat-finger distance of one from the target domain.

This led us to select domains targeting top email providers such as Google, Microsoft, Yahoo, Apple, and Mailchimp. We complemented this list with some of the "second tier" e-mail providers such as Rediffmail Pro, GMX, AOL, Hushmail and ZohoMail.

We hypothesized that we would see more reflection typos on domains that advertise "disposable," instant email addresses. Accordingly, we registered typos of the 10 Minute Mail (`10minutemail.com`) and YOPmail (`yopmail.com`) domains.

To assert the risks linked to SMTP typos, we also registered typos linked to some of the most popular Internet Service providers

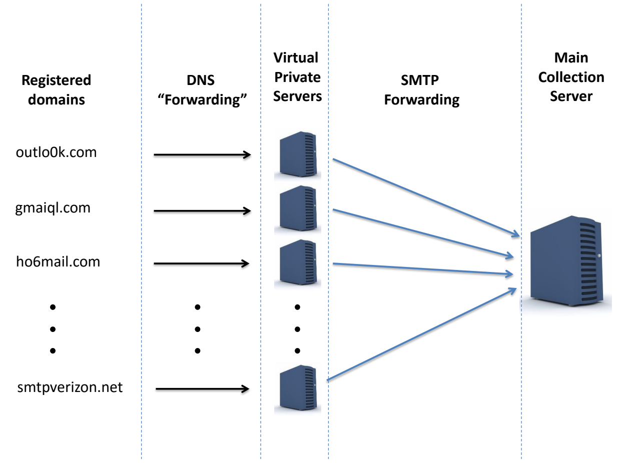

Figure 1: The design of the typo email collection infrastructure

which offer SMTP service to their users: AT&T, Comcast, Cox, TWC and Verizon.

We chose Paypal and Chase as potential sensitive (financial) domains and registered a few domains targeting SMTP typos on these domains.

For each of the target domains, we registered multiple typo domains to compare how different typing mistakes impact the amount of email received.

The complete list of 76 domains we registered, as well as additional information about these domains can be found in an online appendix.[3]

4.2.2 Collection infrastructure. Figure 1 shows a high-level overview of our data collection infrastructure. Each typo domain is assigned a different Virtual Private Server, which in turn forwards the data to our main collection server. This allows us to eschew a potential (but unlikely) issue, of people spamming us from looking up domains and flagging us as security researchers. In addition, to distinguish between different SMTP typo mistakes, we used a one-to-one mapping of our domain names to virtual private server IP addresses. This is because the SMTP protocol does not require the domain name of the SMTP server contacted to be included in the headers. We thus have to differentiate domains by IP addresses.

Table 1 shows our DNS settings for each domain we registered. We include wildcard subdomains to collect typo domains sent to any subdomains of the domains we registered. We run Postfix on our main collection server, which we configure to accept any email sent to any email address. The username and the domain name can thus both be random strings. Our collection server never sends any email out, but ultimately forwards these emails to a processing and storage server (not represented in the picture).

Email processing pipeline. Figure 2 describes this email processing pipeline. When we receive an email we first feed it into SpamAssassin [7]. We do not discard email identified as spam, and instead simply flag it as such. We then tokenize the email into header, body and attachments, save header information, and run both the body and any attachments through a text extraction module (Textract [6]), which operates on a variety of different file formats, even performing optical character recognition on some image files.

Table 1: DNS settings for an example typo domain.

FQDN	TTL	TYPE	priority	record
*.exampel.com.	300	MX	1	exampel.com.
exampel.com.	300	MX	1	exampel.com.
*.exampel.com.	300	A	NA	1.1.1.1
exampel.com.	300	A	NA	1.1.1.1

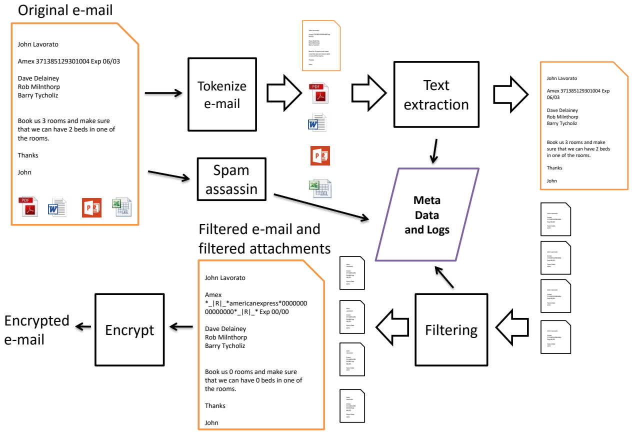
Original e-mail
John Lavorato
Amex 371385129301004 Exp 06/03
Dave Delainey
Rob Milnthorp
Barry Tycholiz
Book us 3 rooms and make sure that we can have 2 beds in one of the rooms.
Thanks
John
Tokenize e-mail
Text extraction
Spam assassin
Meta Data and Logs
Filtered e-mail and filtered attachments
John Lavorato
Amex *_|R|_*americanexpress*000000000000000*_|R|_* Exp 00/00
Dave Delainey
Rob Milnthorp
Barry Tycholiz
Book us 0 rooms and make sure that we can have 0 beds in one of the rooms.
Thanks
John
Filtering
Encrypt
Encrypted e-mail

Figure 2: The typo email filtering system used.

Filtering out sensitive information. We send the text output into a filtering system based on regular expression matching. The idea is to flag when sensitive information is found in an email, while immediately discarding it to protect user privacy. We use the HIPAA list of personal identifiers [3] as a baseline for our set of sensitive information. We replace personal identifiers by salted hashes whenever possible; as an added precaution, we replace *all* digits in the text by zeroes.

We use the public Enron email corpus [25] (May 7, 2015 version) to test how well our regular expression matching heuristics are performing. Table 2 shows the precision (ratio of true positives over true and false positives) and sensitivity (ratio of true positives over true positives and false negatives) for each type of sensitive information. In our context, these metrics are more useful than the more widely used "accuracy" metric. Indeed, because the number of emails containing private identifiers is small overall (and indeed, this is also true of the Enron corpus), we have an imbalanced dataset; as a result, an algorithm that always outputs "no sensitive information was found" would have a high accuracy.

Each score in Table 2 is computed based on sampling 20 random emails per type of sensitive information found in the dataset (except for social security numbers, for which we only had 13 examples

[3] See https://www.andrew.cmu.edu/user/nicolasc/publications/Szurdi-IMC17-appendix.pdf.

Table 2: Precision and Sensitivity of our regular expression based filtering module.

Sensitive info	F1-score	Prec.	Sens.
Credit card number	0.96	0.93	1.00
Social Security number	0.88	0.78	1.00
Employer id. number	0.94	0.89	1.00
Password	0.50	0.33	1.00
Vehicle id. number	1.00	1.00	1.00
Username	0.74	0.59	1.00
Zip	1.00	1.00	1.00
Identification number	0.67	0.75	0.60
Email address	0.99	1.00	0.98
Phone number	0.89	0.83	0.95
Date	1.00	1.00	1.00

available), manually labeling them, and comparing them to what our algorithm produced. The results show a high recall for most sensitive information, except for Identification numbers. The sensitivity for identification numbers is low, because our definition of an identification number is very broad. To validate our results further (beyond the biased sample produced by our algorithm), we sampled an additional 100 random emails from the Enron dataset and manually labeled them. Due to the imbalanced nature of sensitive data, we only found phone numbers, emails and dates in this sample. The sensitivity remains high however—0.91, 1.00 and 0.98 for phone, date and email respectively.

Once all of this processing is done, we encrypt each part (header, body, attachment) and most of the log files for storage on our collection server.

4.3 Email classification

After running our experiment for a few days, it became obvious we were receiving very large amounts of spam, which would completely bias any analysis if left unfiltered. Spam can come from miscreants noticing our servers accept any email (even though they don’t relay to any party but our collection server), or from users mistyping their own email address (reflection typo) and being subsequently added to promotional lists. Some of our domains might have also been previously registered, and could still appear in certain promotional lists.

We thus turned to building a filtering and classification module, which not only filters out spam, but also classifies reflection typo emails that result from a single typo (e.g., making a typo while signing up for a mailing list). Our classification module consists of five layers, which act as a funnel: each email marked as spam in a given layer is not further considered.

Layer 1: Detecting erroneous header fields. Emails in which the name of the SMTP server relaying the mail to our collection server does not match the name of one of our registered domains is immediately classified as spam. The sender’s address should also not belong to one of our domains, since we do not send any email. Conversely, spammers often pose as sending from the same domain as the intended recipient. Thus, any email in which the sender appears to be one of our domains is classified as spam. In receiver or reflection typo emails (but not in SMTP typo emails), the recipient’s email address should belong to one of our typo domains.

Table 3: Evaluation of Spamassassin on four datasets

Dataset	Precision	Recall
TREC [8]	0.98	0.79
CSDMC [2]	0.98	0.87
SpamAssassin [7]	0.97	0.84
Untroubled [9]	–	0.23

Layer 2: SpamAssassin. We run SpamAssassin on all incoming email. Table 3 shows our evaluation of SpamAssassin in local mode with the default thresholds on four different datasets. While precision is good, the low recall indicates we need additional filtering. We immediately remove all emails with ZIP or RAR attachments and consider them as spam—we indeed receive large amounts of such emails, and every single one of them we manually inspected was spam.

Layer 3: Collaborative spam filtering. If a sender sends us spam once, we consider all of the emails from that sender, across all of our domains, to be spam. Furthermore we apply bag-of-words analysis to the email body. If the analysis yields more than 20 words, we flag all other emails with a matching bag-of-words as spam. This filtering step should have high precision, because it is highly unlikely that two emails would be spam and ham, respectively, if both emails use the same corpus of words.

Layer 4: Detecting reflection typos. Emails that have survived the first three layers might not be spam, but still be the product of automated systems. For instance, a user might have made a typo while signing up for a certain service, and subsequently received notifications to that erroneous address. We automatically classify these emails, using a set of regular expression heuristics. If an “unsubscribe-list” header field is present; “bounce” or “unsubscribe” appears in the `Sender:`, `From:`, or `Reply-To:` fields; or if any two of `From:`, `Reply-To:`, or `Return-Path:` have different values, we classify the email as a reflection typo. We additionally search for strings including “unsubscribe,” “remove yourself,” and other similar content in the body to flag email containing such strings as reflection typos. Finally, we also filter out emails sent from system users, e.g., “postmaster,” “root,” or “admin.”

Layer 5: Frequency-based filtering. Finally, the last layer filters out receiver typo emails (but not SMTP typos) for which the sender address, the recipient email address, or the email body appear too often in our corpus. The insight here is that true typo emails ought to be unique, rare instances. We selected thresholds for these frequencies based on the distribution of these features to include the most common frequencies and to exclude outliers. We set the receiver address frequency threshold to be 20, and both the sender address and content thresholds to 10. Details of these distributions (which motivate these thresholds) can be found in our online appendix.

Performance analysis. To ensure that our spam filtering performs decently, we conducted small manual analysis of receiver typo emails. We randomly selected 5 emails (collected between June 6 and September 16, 2016) for each domain name where we expected to

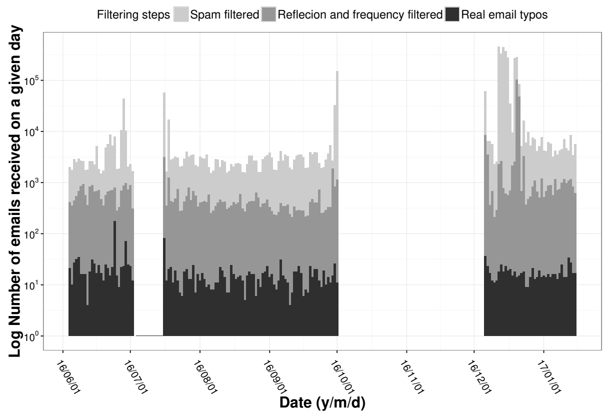

Figure 3: The number of receiver typo emails received daily during our data collection. Emails are in three categories: spam, auto and frequency filtered emails, and true typo emails. The plot is *not* stacked, and is in logarithmic scale on the y-axis.

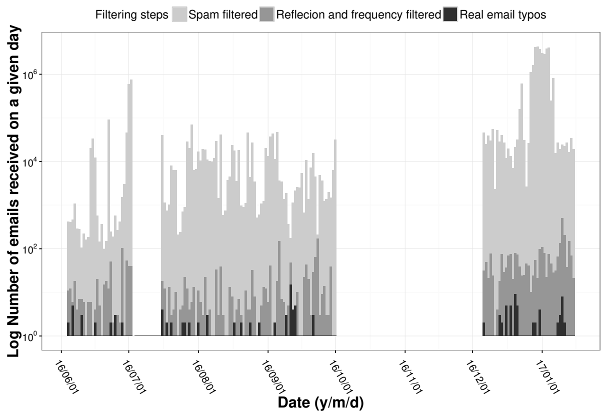

Figure 4: The number of SMTP typo emails received daily during our data collection. Emails are in three categories: spam, auto and frequency filtered emails, and true typo emails. The plot is *not* stacked, and is in logarithmic scale on the y-axis.

receive receiver typo emails. One researcher analyzed the emails to decide whether they are spam emails or not. In total, the researcher labeled 77 emails and found that 80% of them were not spam emails. (Detailed, per domain results can be found in the online Appendix.) We additionally analyzed 26 emails that arrived by domains where we did not expect to receive anything but SMTP typos, yet, were classified as receiver typos by our system. 25 of these 26 emails turned out to have been correctly identified as receiver typos.

4.4 Analysis

We next turn to the analysis of the emails our infrastructure collected over more than seven months. In this entire discussion, we report numbers projected over a full year. Indeed, there were minor differences in data collection period for each domain (due, e.g., to the infrastructure being partially overwhelmed on certain days), so that we need to normalize all numbers to a common scale. Given that the study was over seven months, we hypothesize that any daily, weekly, monthly, and most seasonal effects are accounted for in our collection. In short, when collect x emails, we report the number $y = x \cdot 365/d$ where d is the number of days we actually collected data for that domain.

4.4.1 Email volume. Figure 3 and 4 represent the total email count, per day, we received during our collection, broken down between receiver typos (Figure 3) and SMTP typos (Figure 4). Collection gaps correspond to times during which our infrastructure was malfunctioning (in particular due to being overwhelmed with spam, and crashing as a result, with little hopes of recovering two months worth of data). We receive SMTP typo emails sparsely in small batches which perfectly characterizes what we expected. Users rarely make SMTP typo mistakes and when they do, then they quickly recognize the error and correct it. On the other hand, receiver typos occur with a near-constant rate.

Projecting from the seven months of data collection, our infrastructure receives 118,894,960 emails per year. Based on the email header, 16,233,730 are candidates to be receiver or reflection typo emails and 102,661,230 are candidates to be SMTP typo emails.

However, most of these emails turn out to be spam—only 7,260 emails per year pass all of our filters. Correcting, based on our manual analysis, would bring that number further down to 6,041 emails/year being either receiver or reflection typos.

For SMTP typo candidates we found that 5,147 emails/year are sent to us by automated agents; 5,555 of the candidate SMTP typos emails per year are frequency filtered and 415 are not. However SMTP typos, by their very nature, may lead a single user to send large amounts of email (if only for a short time), which could lead frequency filtering to produce false positives. Hence we estimate our infrastructure receives between 415 and 5,970 SMTP typo emails/year.

Surprisingly to us, we received a non-negligible number of receiver typo emails (over 700 emails/year) to domains that we had specifically designed to catch SMTP typing mistakes (for instance, `mx4hotmail.com`). These emails do not appear to be spam (as discussed above, we looked into 26 of them), but we are not sure what is causing this behavior.

4.4.2 Per-domain analysis. We next turn to discussing whether some domains receive more typos than others, and why.

A small fraction of domains received most of the receiver typos. Out of the 31 domains registered to collect receiver typo emails, 27 domains targeted email providers, excluding temporarily email address providers (10minutemail.com and yopmail.com) or bulk email sending services (sendgrid.com and mailchimp.com). Figure 5 shows that out of these 27 domains only two domains received the majority of the total receiver typo emails and 12 domains received 99% of all emails. This finding reinforces our intuition that some typosquatting domains are orders of magnitude better than others.

SMTP typos are infrequent compared to receiver typos. We receive an order of magnitude less SMTP typo emails than receiver

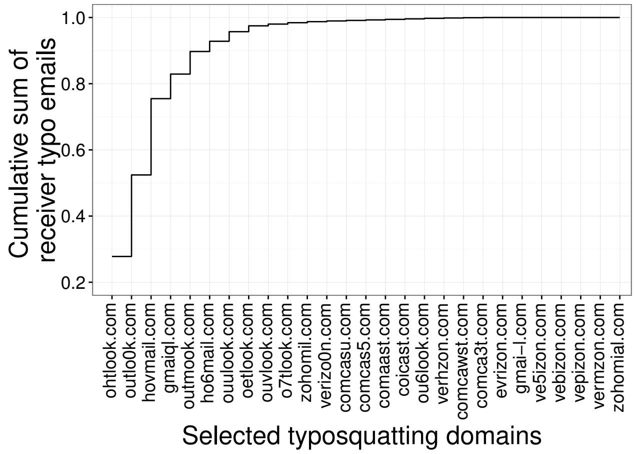

Figure 5: Cumulative sum of emails received by our typosquatting domains.

typo emails. So, SMTP typosquatting has questionable profitability, compared to what receiver and reflection typo mistakes could offer. However, there is no harm, to the typosquatter, in simply collecting these emails on domains they would have already registered.

We define as the *persistence* of an SMTP typo for a given user the time difference between the first and last email received from that particular user. For 70% of our users, we received only one email due to a SMTP typo mistake, so that the persistence is undefined (i.e., taken to be equal to zero by convention). 83% of SMTP typos lasted less than a day and 90% less than a week. The maximum persistence was 209 days. When an SMTP typo persisted for this long it can be for one of two reasons: the same user made the same mistake several times, or these emails were spam our filtering system did not catch. 90% of SMTP mistakes caused the users to send four or less emails to our servers. As discussed earlier, emails filtered out during the frequency filtering step might include SMTP typo mistakes; however without manual inspection of their content, we cannot draw conclusions about these emails.

Visual distance, target popularity, and keyboard distances are important features. Typosquatting domains targeting more popular target domains (`gmail.com`, `outlook.com`, `hotmail.com`), unsurprisingly receive significantly more receiver and reflection typo emails. More interestingly, for a given target domain, FF-1 domains always receive the most emails if the typing mistake is not totally obvious (`evrizon.com`, `ohtlook.com` and `outlo0k.com`). In other words, visual distance seems more important than keyboard distance. Figure 5 confirms that the top two domains are DL-1 and FF-1 typos of two of the three most popular email providers, with low visual distance from the real domain.

We only found a statistically significant correlation between the popularity of the target domain and the number of reflection and receiver typo domains received. This is not surprising since the popularity of the target domain outweighs the other attributes, and without an explanatory variable we cannot expect to see significant correlation with other attributes of the target domain.

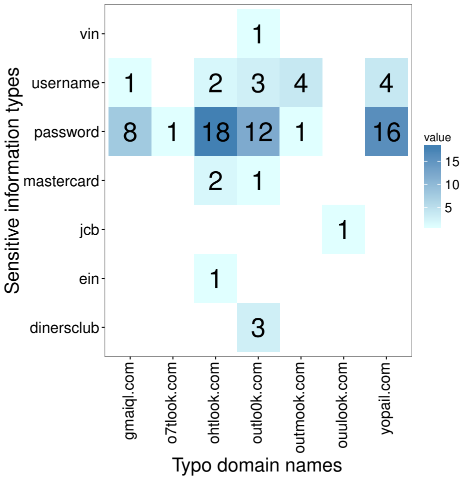

Figure 6: Heatmap of sensitive information of real typo emails. The heatmap shows the frequency of a sensitive information type for a given typosquatting domain.

4.4.3 What does a typosquatter receive? Figure 6 shows among the true typo emails which ones received what kind of sensitive information. Unsurprisingly, `yopmail.com` typo domains to receive a fair amount of usernames and password since their emails are often used for temporary registration.

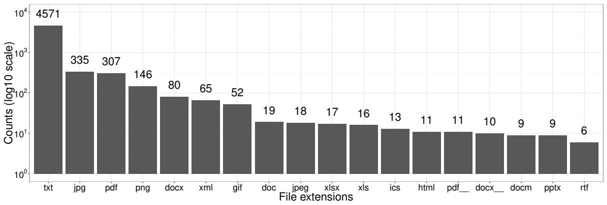

Figure 7: Frequency of extensions among true typo emails.

Attachment analysis. Figure 7 shows the attachment extensions' distribution for all receiver typo emails we received. The distributions of extensions for spam emails and true typo emails significantly differ. Without filtering the emails we received have a significantly higher proportion of file types that are easier to exploit such as `.doc`, `.docm`, `avi`, `.xls` and `.xlsm`. (Recall we discard ZIP and RAR files during our filtering process.)

Out of a randomly selected 109,151 unique file hashes we found 323 in the VirusTotal database [29]. 304 of the hashes were found to be malicious and 17 were benign. All emails containing these malicious attachments were categorized as spam by our filtering system. (The benign hashes likely do not contain personal, sensitive information since they have already been observed elsewhere in the VirusTotal database.)

The dangers of reflection typos. We found that one particular email address at `zohomil.com` received a lot of emails with CVs and work search related subjects and attachments. It turns out that somebody included a mistyped email address in various job postings on multiple pages—a nasty variant of a reflection typo.

Table 4: SMTP support of typosquatting domains

Support status	Count	% total	% analyzed
No MX or A record found	651,439	15.5	23.7
No info	144,1725	34.4	-
No email supp.	28,3636	6.8	10.3
Supp. email, no STARTTLS	1,693	0.0	0.1
Supp. STARTTLS with errors	257,952	6.2	9.4
Supp. STARTTLS w/o errors	155,6773	37.1	56.6

5 THE EMAIL TYPOSQUATTING ECOSYSTEM

We complement the results from our experiment playing the role of a typosquatter with a more “passive” analysis, in which we attempt to estimate whether email typosquatting does occur in the wild, and who the actors are.

5.1 Methodology

To gain an better understanding of the typosquatting ecosystem we first looked at the set of ctypo domains registered in the wild. We generated all possible DL-1 variations of Alexa’s top one million domain on November 5, 2016 [1]. We considered the set of ctypo domains, i.e., the domains that are actually registered, and collected the MX and A records of these ctypo domain names, on November 7, 2016. The SMTP protocol specifies that, in absence of an MX record, the A record of the domain name should be used as the mail server’s address [23]. We clustered ctypos together based on their DNS settings to see any evidence of concentration in the typosquatters’ infrastructure. If there was no MX record found for a domain name we used the corresponding IP address for clustering.

We further analyze whether these domains actually run an SMTP server using data downloaded from zmap.io [10] on October 29, 2016. We checked the IP addresses obtained from requesting the A record for those domains for which an MX record was found. If there was no MX record, we used the A record directly.

We also attempted to collect WHOIS information for all ctypo domains between December 22 2016 and January 24 2017. We used PyWhois [4], and Ruby Whois [5] for querying and parsing WHOIS information. While a lot of the information is probably fake, it can nevertheless be useful in clustering domains by owners (e.g., while Mickey Mouse is unlikely to register typosquatting domains, repeatedly seeing the name Mickey Mouse as a technical contact for typosquatting domains might be evidence of common ownership).

More precisely, to cluster registrants of typosquatting domains we use an approach similar to Halvorson et al. [20]. We use six fields of the WHOIS record: registrant name, organization, email address, phone number, fax number and mail address. We consider two domain names to be registered by the same entity or group of entities, if four of the six fields match. Naturally, this means we cluster only domains for which at least four WHOIS fields were available. Using a .com zone file, we find domain name servers that serve a significantly higher proportion of typosquatting domains than should be expected.

5.2 Analysis

SMTP support for typosquatting domains . Table 4 shows SMTP support for typosquatting domains. 22.3% of typosquatting domains are not capable of receiving emails, 34.4% did not yield any information, and 43.3% support SMTP.

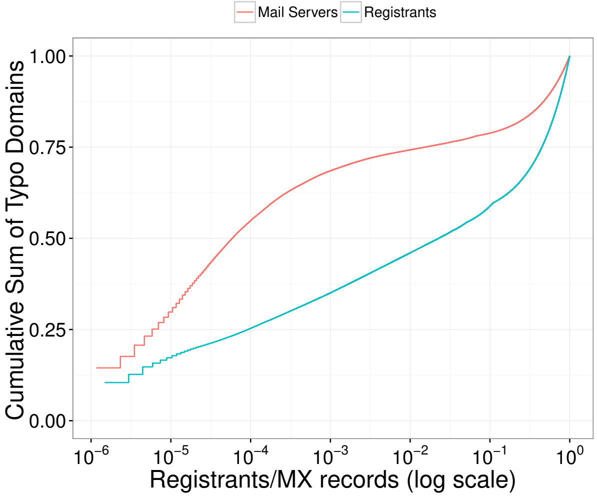

Figure 8: Cumulative sum of typosquatting domains by mail servers and registrants. Mail servers and registrants are ordered by the number of domains served/owned, in decreasing order.

Typosquatting registrants. Using the clustering technique described above, Figure 8 shows the concentration among registrants (excluding those protected by WHOIS proxy services) who filled out at least four of their WHOIS registration fields. The *x*-axis is the fraction of all registrants. The top 14 registrants own 20% of typosquatting domains. A mere 2.3% of all of the registrants in appear to own the majority of typosquatting domains. At the same time, there is a heavy long tail for the ownership of the rest of the domains.

Most of the registrants that operate a large number of typosquatting domains have SMTP servers active on most of their domains. The top three registrants are actually companies whose business appears to be holding domain names for sale. While questionable, this practice is not evidence of active malice. On the other hand, many of the other registrants do not seem to focus on domain resale, but do operate SMTP servers, which is suspicious. The online appendix contains a list of the top typosquatting registrants.

Suspicious name servers. A number of name servers are used by a significantly higher ratio of typosquatting domains compared to benign domains. In general, the average ratio of typosquatting domains over benign domains is about 4% – par for the course for large organizations that may not be able to check very carefully the activities of all of their customers. However, a number of name servers far exceed that ratio, and can be viewed as catering to typosquatters. The candidate typosquatting ratio of all .com domains is as high as 89% for one such name server. Further adding to the suspicion, half of these name servers are registered behind privacy proxies, and a majority of their domains have active SMTP servers. Full details can be found in the online appendix.

MX record concentration. As Figure 8 shows, not only do a lot of typosquatting domains support mail, but many of them point to only a few MX records. The top eleven SMTP servers handle mail for more than one third of typosquatting domains and 51 for the majority. Less than one percent of the SMTP servers supports more than 74% of domains. In other words, a few providers might have the chance to defend against (or be held responsible for, in case they are colluding with the miscreants) potentially dangerous and privacy invasive email typosquatting.

SMTP and mail typos. Some typosquatters deliberately target SMTP subdomains (e.g., registering `smtpgmail.com` to `smtp.gmail.com`) and webmail domains (e.g., by registering `mailgoogle.com`, targeting `mail.google.com`). We found 41 SMTP and 366 mail typosquatting domains registered, targeting Alexa's top 10,000 `.com` domains and Alexa's top 500 `.com` domains in the email category.

The SMTP typos include domains `smtpgmail.com`, `smtpoutlook.com` and `smtplive.com` targeting the biggest email providers. This could plausibly be defensive registrations. However, they are *privately* registered, which is inconsistent with trademark protection—in our experience, defensive registrations usually point at the legitimate owner or their agent, not at a private registration service.

6 EXTRAPOLATING FROM OUR EXPERIMENTS

In this section, we combine the observations gleaned by through our experiment (Section 4 and our analysis of the typosquatting ecosystem (Section 5) to attempt to extrapolate our findings on an admittedly limited set of domains to the whole Internet.

6.1 Toward a projection

We use a seed of 25 of our typosquatting domains targeting 5 email domains: `gmail.com`, `hotmail.com`, `outlook.com`, `comcast.com`, and `verizon.com`. These domains are highly popular email services, and using the information from our small foray into typosquatting might help us best understand the potential magnitude of email typosquatting in the wild.

Specifically, we attempt to project our results to other typos of email domains. To do so, we rely on three hypotheses

- **(H1)** Typing mistakes are equiprobable among users of different email providers.
- **(H2)** Sending an email is a two-step process. Users type in the email address. Second, users verify the address and potentially correct any mistakes.
- **(H3)** The number of emails sent to a typosquatting domain is proportional to the number of emails sent to the target domains.

Based on these hyotheses, we build a simple model to estimate the expected number of emails sent to a given typo domain

$$E_{ij} = E_i \cdot P_{t_{ij}} \cdot (1 - P_{c_{ij}}) \,,$$

where E_i is the expected number of emails (over a fixed time period, e.g., a year) sent to email addresses in domain i, E_{ij} is the expected number of emails sent to email addresses in domain j, where the DL distance between i and j is either zero or one.

$P_{t_{ij}}$ is the probability of user typing j instead of i. (This includes typing the correct domain.) $P_{c_{ij}}$ is the probability of the user correcting the mistake after typing j instead of i.

Directly validating this model is impossible, because $P_{t_{ij}}$ and $P_{c_{ij}}$ are unknown, and different for different domains, even in the case of similar typing mistakes. Instead, we build on this simple model to devise a linear regression model used to predict E_{ij} based on features characterizing the process of typing mistakes.

First, we use Alexa's monthly unique visitors to estimate E_i for email domains (e.g., `gmail.com`, `outlook.com`). We assume E_i is proportional to the number of active users of domain i.[4] We add three features to incorporate $P_{c_{ij}}$ into our model: the visual distance, the length of the target domain and position of the mistake, and the fat-finger distance.

One drawback of our approach is that we were not able to register domains of popular email providers with deletion or transposition typos. Thus we used Alexa's data on typosquatting domains of the 40 most popular target domains, to estimate the difference in probability between different typing mistakes. We collected Alexa's data from October 27, 2016 to October 30, 2016 [1].

Furthermore, we removed typosquatting domains receiving outstanding traffic among typos of the same target domains, because those domains are probably not malicious, and just happen to be accidentally close to the target domain. We used the median of all absolute deviations from the median (MAD, [26]) to detect such outliers. We estimate the 95% confidence interval for the mean of the different typing mistakes to estimate how different their average traffic is. We will use these results to estimate the number of emails received by deletion and transposition typo domains.

6.2 Regression results

The five target domains—`gmail.com`, `hotmail.com`, `outlook.com`, `comcast.com` and `verizon.com`—are targeted by 1,211 typosquatting domains (excluding defensive registrations, and our own 25 domains).

We build a linear regression model, by transforming the dependent variable to square root space. We select the following three features: the target domain's Alexa rank (log transformed), the square root of our visual distance heuristic (between the target and the typo domain) normalized by the length of the original domain and the fat-finger distance between the target and the typosquatting domain (zero or one). The R^2 value of the fit is 0.74. Running a leave-one-out cross-validation test the R^2 value drops to 0.63.

Our model finds that the 1,211 typosquatting domains registered by others should receive approximately 260,514 emails per year, with a 95% confidence interval ranging between 22,577 and 905,174 emails per year. Figure 9 shows based on the AWS Alexa data collected that deletion and transposition typo mistakes are significantly more frequent than addition and substitution mistakes. Taking this information into account, our modified regression analysis yields an expected number of emails received by typosquatters equal to 846,219 with a 95% confidence interval ranging between 58,460 and 4,039,500.

[4] This assumption does not hold in the general case, when web popularity may be very different from email usage; but we assume it is reasonable in the case of the webmail domains we are looking at.

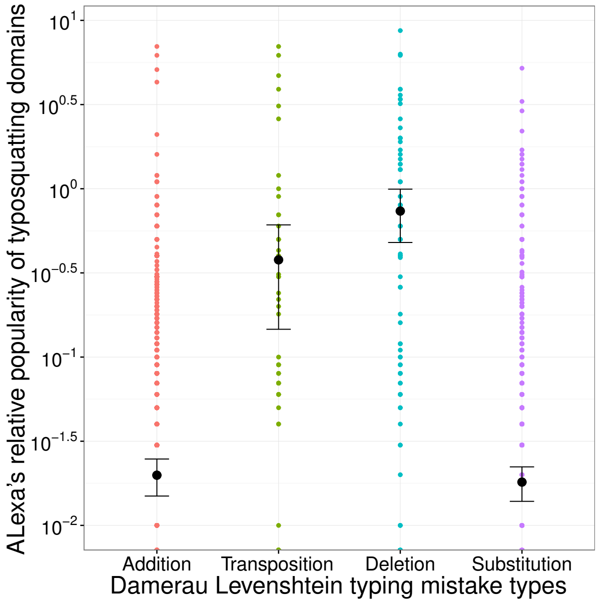

Figure 9: The average relative popularity of typosquatting domains separated by the type of typing mistakes: addition, deletion, substitution, transposition. We also marked the average popularity and the 95% confidence interval for each type of mistake.

Economic implications. Registering a .com domain costs about about USD 8.5 per year. Using this price in the model above, a typosquatter owning these domains can acquire an email for less than two cents. (This computation excludes spam.) From our own experience, by keeping our five top performing typosquatting domains we could collect "legitimate," non-spam emails for less then a penny a piece (excluding marginal costs, such as those of running a server, and keeping storage).

However, we conjecture that the domains registered by us were mostly available, because they are less profitable than other typosquatting domains. In other words, we would not be surprised if our calculations only provided a relatively conservative estimate on the number of emails typosquatters actually receive when registering typosquatting domains targeting popular email service providers.

The very small set of emails we manually analyzed appear to contain a wide variety of sensitive information that cannot be exploited by itself, but can aid miscreants to perform targeted attacks. For instance, six of the 103 emails we analyzed manually appeared to contain digital receipts, which contain considerable personal information that could be used for subsequent spearphishing campaigns or other scams; some other emails included information (car registration, visa documents, resumes, adult side registration, medical records) that could plausibly be used for identity impersonation, spear-phishing, or even intimidation.

7 IN THE SHOES OF A TYPOSQUATTING VICTIM

We have discussed the potential threat of email typoquatting and the existing ecosystem that appear to support it. However, are typosquatters actually doing anything with the emails that they are able to collect? To answer this question, we run an additional experiment, in which we now play the role of a potential victim, and deliberately email known typosquatting domains with "honey emails." This experimental protocol, like the collection protocol earlier described, was vetted and approved by our IRB.

7.1 Experimental design

Honey email design. We designed our honey emails to 1) signal back to our servers when opened and 2) to include seemingly sensitive information (e.g., login credentials), whose access we can monitor.

Our emails included a 1x1-pixel tracking image residing on a VPS we operate. HTML clients might try to download this image upon opening the email, but this is not always the case. For instance, depending on its default configuration, the Thunderbird email client may not automatically download such embedded images. Shortly stated, presence of a signal indicates that the email has certainly been opened, but absence of a signal is not proof that the email was not opened.

We included sensitive information in the form of honey tokens and honey accounts. A honey token is a file attachment that signals back upon being opened. After experimenting with both PDF and DOCX, we discovered that DOCX readers tend to allow external access by default more commonly than PDF readers.

Our honey accounts consisted of email accounts at two major email providers and a shell account on a VPS we control. The wording and headers of each email were designed to mimic real-life interactions between users. (We piloted these emails with members of our research group, to confirm they looked plausible, and were not caught by spam filters.) In total we used four different email design templates, and we made sure to send one typosquatter registrant one of each email designs exactly once. Further, we only sent one email to each typosquatting domain.

Our first email design included login information for a major email service provider. The second design included login information for a shell account under our control. The third design included a link to a tax document shared through a major document sharing service, where we could monitor accesses. Our final design had a DOCX attachment with (fake) payment information.

Sending emails. We ran two measurement experiments.

Email probes. The first experiment had for objective to determine how many typosquatting domains actually accept email – the idea is that this gives us a rough idea of how many are deliberately set up as email typosquatting domains, as opposed to web typosquatting domains that happen to also target email domains. To that effect, we started with a pilot, in which we sent out a small number of 164 honey emails between May 2, 2017 and May 6, 2017. We selected a low number of target domains (and a low sending rate) to avoid alerting typosquatters to our measurements. However, most of our emails bounced, or resulted in a timeout or network error.

After this pilot, we ran a larger measurement to test how many typosquatting domains accepted *any* of our emails. To that effect, on May 15, 2017, we sent out 152,985 benign emails to 50,995 typosquatting domains, including domains of registrants owning the most typosquatting domains, domains linked with a name server frequently used for typosquatting domains, typos of the three major email domains (`gmail.com`, `hotmail.com`, `outlook.com`), and finally candidate typosquatting domains that use the most popular WHOIS privacy service.

Each domain selected listens on (some of) the SMTP server ports, according to Zmap. To verify which one, we sent three emails – one each to ports 25 (no authentication), 465 (SSL) and 587 (STARTTLS).

The emails in this experiment were designed to look like test email without any sensitive information in them. Here, we sent emails from our own virtual private servers. This allowed us to determine whether emails were actually received and/or read in a client that retrieves external resources.

Honey tokens. We then conducted a second set of measurements, in which the goal was to determine if emails were not only received, but also read and/or acted upon. Here too, the experiment started with a *honey token* pilot measurement limited to 738 domains out of these 50,995 typosquatting domains, to ensure that the infrastructure worked as it was supposed to and to run a conservative measurement unlikely to be detected by miscreants—indeed, most typosquatters, even those who operate myriad domains, received at most one email from us. We selected these 738 domains by 1) purposefully limiting ourselves to at most four domains per registrant we could identify, and 2) selecting these four domains based on their Alexa rank and the type of typing mistake. We sent out one honey email containing sensitive information to each of these typosquatting domains on May 15, 2017. All emails in this pilot were sent through a major email provider to make them less conspicuous and to avoid spam filters.

Following this pilot, on June 15, 2017, we ran a far more aggressive measurement, in which we sent all four different honey emails designs to *all* 7,269 typosquatting domains which had accepted our emails in the first set of experiments. Here, due to the size of the test (close to 30,000 emails), we used our own servers, rather than a major email provider, to send out these emails. During this test, while we only sent four (different) emails per domain exactly once, we potentially sent out the same email multiple times to the same individuals – since some typosquatters own more than one domain.

We logged access attempts to the “honey” shell account until July 1, 2017; and accesses to the other resources our honey tokens were pointing to until September 14, 2017.

7.2 Results

Typosquatting domains and email acceptance. Table 5 presents the results of our first experiment, in which we monitored whether our honey emails were accepted. 1,170 publicly registered domains accepted our emails without any error message. Based on our access logs, three of these domains, including two (`outfook.com`, and `uutlook.com`) that seem to be clear typosquatting domains, appear to have read our emails. On the other hand, we experienced a large percentage of network errors and timeouts for the majority of publicly registered domains.

Table 5: Error message count received when running the initial test for the honey email experiment.

	Number of typo domains	
	Public reg.	Private reg.
No error	1,170	6,099
Bounce	1,567	1,160
Timeout	17,923	6,976
Network Error	7,901	6,584
Other error	93	1,522
Total	28,654	22,341

6,099 of our emails were accepted on domains using WHOIS privacy proxy services, which overall presented far less errors. 19 of these emails were read based on our logs. We discovered that 6 of these domains were clear typosquatting domains, 8 were legitimate domains that just happened to look like typosquatting domains, and 5 could be either way. Glancing at the time difference between emails were sent and when emails were opened seems to suggest that these emails might have been read by humans – rather than by automated processes – as it frequently took several hours before the email was opened. Furthermore, some of these emails were opened several times, sometimes days after they were first opened

Interestingly, some of these domains appear to be targeting potentially sensitive sectors, such as banking (e.g., `disvover.com`, `bankofamericqa.com`), ...), adult sites (e.g., `nuaghtyamerica.com`), or email providers (e.g., `comcacst.com`).

Table 6: Distribution of the mail exchange server usage for the domains that accepted our emails.

MX domain	Total	%	CDF	Private?
`b-io.co`	3,171	43.6	43.6	Yes
`h-email.net`	1,344	18.5	62.1	Yes
`mb5p.com`	732	10.1	72.2	Yes
`m1bp.com`	635	8.7	80.9	Yes
`mb1p.com`	558	7.7	88.6	Yes
`hostedmxserver.com`	225	3.1	91.7	Yes
`hope-mail.com`	176	2.4	94.1	Yes
`m2bp.com`	94	1.3	95.4	Yes
`google.com`	61	0.8	96.2	No
`googlemail.com`	34	0.5	96.7	No

Table 6 shows that 95% of the domains which accepted our emails without errors rely on eight mail server domains, which are all privately registered.

Honey tokens and honey accounts. While the pilot measurement – sending data to 738 domains only – did not result in any signal being sent back to us, our larger measurement to all 7,269 suspected typosquatting domains resulted in 15 emails being apparently opened and/or read by someone, and two honey tokens being accessed. Here too, we saw a lag of several hours between the time we sent emails and the time they were opened or read, suggesting human involvement.

Specifically, on June 16, 2017 a potential typosqatter read the "tax document" we had uploaded to a known document sharing service. The domain we sent this honey email to was a legitimate service once, but for the past two years it has been operating as a parked domain. Logs provided by the document sharing service indicate that the document was opened half an hour after we sent it, and was viewed for 28 seconds from Caracas, Venezuela using a Windows desktop computer. We also saw that 9 days later someone read our email from another IP also from Caracas, Venezuela and 14 days later from Orlando, Florida.

Likewise, on June 16, 2017 a potential typosquatter tried to gain access to our honey shell account from an IP in Poland. This specific email did not show up in our logs as having been viewed, presumably due to the miscreant not opening inlined images.

While interesting, we caution that these two anecdotes are far from providing evidence of systematic email collection and monetization by typosquatters—in fact, given the number of emails we sent, it seems that these practices are the (rare) exception rather than the norm.

This overall negative result may be explained by several factors. First, the risk involved with getting caught might be higher then the expected benefit of the sensitive information we sent them. Second, it is possible that typosquatters do not even realize that they are collecting these emails; plausibly, the SMTP servers could have been turned on by default, and not wilfully. After all these domains might have been registered primarily for *web* typosquatting, with email typosquatting being an afterthought, if a thought at all.

8 DISCUSSION AND LIMITATIONS

A major limitation of this study is that it only considers domain typosquatting, and not username typosquatting. For instance `aliec@gmail.com` might receive a lot of email meant for `alice@gmail.com`. However, without the collaboration of the email service provider, doing an analysis of username typosquatting is impossible.

Our data collection experiments show that there is potential danger, but, contrary to web typosquatting, the "expected" risk to consumers is far less obvious – most of the time, the risk is probably very low, but in a few cases, depending on the specific content that is being sent, might lead to disastrous outcomes (contrary to web typosquatting).

While we have seen only scant evidence of credential abuse in the wild when we posed as victims, we have on the other hand discovered highly suspicious registration patterns. These may be a by-product of web typosquatting, but we cannot rule out that the situation will not change; the infrastructure appears to be certainly already in place, even though this may be accidental.

Web vs. email typosquatting. Web typosquatting is one of the easiest attacks to carry out, because it requires almost no technical knowledge. As our measurements show, some parties are seemingly interested in exploiting typing mistakes and have the ability to collect emails from potential victims. Yet, they don't appear to act upon these emails, even though there is plenty of evidence (from our data collection) that many people actually could fall victim to this kind of attack.

Reflecting more on this negative result, web typosquatting only needs the ability to register a domain and the subscription to a parking service, and is thus accessible to any miscreant. On the other hand, email typosquatting requires deeper technical expertise. First, the collection infrastructure is not straightforward to set up. Second, spam filtering is equally complex—as we saw in our own experiment, spam filters alone might not be very reliable. To add insult to the injury, the payoff is far more uncertain (low occurrence, high payoff) than in the web typosquatting case (high occurrence, low payoff), and the risk of getting in trouble (e.g., if abusing financial credentials) is much higher.

Possible defenses. What if the situation were to change, and typosquatters actually used emails received for profit? Our results in Section 4.4 shown that far more emails are received by typosquatting domains targeting top email service providers compared to middle sized providers. This trivially means that large providers registering their typosquatting domains defensively would have the biggest impact per defensive registration and also it would be the most cost effective per user. While for a small company it might be financially burdensome to register hundreds of domains (not mentioning the legal costs in case the domains are owned by someone else), for major companies, a few thousand dollars a year should be a negligible cost. It is not unprecedented for a large company to acquire typosquatting domains in bulk even if legal lawsuit is needed. Facebook a few years ago won a lawsuit summing to $2.8 million against typosquatters, recovering 105 typo domains. UDRP and ACPA provide frameworks for brand owners to acquire typosquatting domain names, in case they are already owned by typosquatters. Similarly these costs should be low compared to the potential harm for the financial sector such as banking domains.

Besides defensive domain registrations, typo correction tools could also help to reduce the potential harm from typosquatting. Typo correction could be integrated into any input field: at SMTP setup phase, registrations, email recipient, or when giving contact information in online forms.

Policy interventions could also be viable. For instance, the Chinese registry raised the registration price and requiring identification for `.cn` domains. Raising the cost of domain registration and requiring identification for registration would definitely drive most of the typosquatters out of business. However these intervention would potentially have a high collateral damage on legitimate domain owners. Another approach would be for ICANN and registrars to periodically remove typosquatting domains. This however is unlikely to happen due to incentive misalignments, namely that this would require a great effort from this parties who do not suffer from this activity and at the same time their revenue would decrease.

9 CONCLUSION

We conducted a measurement study of email typosquatting, based on our own data collection, and an examination of the whole ecosystem. We conclude that the profitability of a typosquatting domain depends on three main factors: popularity of target domain, edit distance from target domain, and visual distance from the target domains. We observed that receiver and reflection typo emails are an order of magnitude more frequent than SMTP typo emails. Among the emails received we found users accidentally sending us email containing highly sensitive personal data. We also observed that some registrants own thousands of email typosquatting domains, that these domains support SMTP. Furthermore, some of the name servers (and registrars) used by tens of thousands of typosquatting domains appear to be cesspools, with a 5–10 higher typosquatting domain ratio than normal. Even though typosquatters have the infrastructure to collect private emails in bulk literally for pennies each, we found that, with very rare exceptions, they do not actually misuse sensitive information sent to them. We conjecture this may be due to incentives being in favor of web typosquatting—shortly stated, it is not worth bothering with a more complex attack with a more uncertain payoff—but cannot guarantee the situation will not change. Certainly, the potential for monetization by a determined actor is there, and proactive defenses ought to be considered.

10 ACKNOWLEDGMENTS

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. This work benefited from extensive discussions with Michael Stroucken, Mahmood Sharif and the Carnegie Mellon University cybercrime group. We also thank our anonymous reviewers, including those on the student shadow PC, for their useful feedback, and Thorsten Holz for shepherding this paper.

REFERENCES

[1] Alexa Web Information Service. http://aws.amazon.com/awis/.

[2] CSMining group: CSDMCS 2010 spam dataset. http://csmining.org/index.php/spam-email-datasets-.html.

[3] HIPAA Protected Health Information Identifiers (45 CFR 164.14). http://www.ecfr.gov/cgi-bin/text-idx?SID=e58a563f56b8cf8e6511be534d364a64&node=se45.1.164_1514&rgn=div8. Last accessed: September 30, 2017.

[4] Python WHOIS parsing tool. https://bitbucket.org/richardpenman/pywhois.

[5] Ruby WHOIS parsing tool. https://whoisrb.org/.

[6] Textract. https://textract.readthedocs.io/en/stable/. Last accessed: September 30, 2017.

[7] The Apache SpamAssassin Project. http://spamassassin.apache.org/.

[8] Trec spam dataset. http://trec.nist.gov/data/spam.html.

[9] Untroubled.org spam archive. http://untroubled.org/spam/.

[10] Zmap: Internet-Wide Scan Data Repository. https://scans.io/.

[11] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015)*. Internet Society.

[12] Ross Anderson, Chris Barton, Rainer Böhme, Richard Clayton, Michel JG Van Eeten, Michael Levi, Tyler Moore, and Stefan Savage. 2013. Measuring the cost of cybercrime. In *The economics of information security and privacy*. Springer, 265–300.

[13] Anirban Banerjee, Dhiman Barman, Michalis Faloutsos, and Laxmi N Bhuyan. 2008. Cyber-fraud is one typo away. In *Proc. IEEE INFOCOM 2008*, 1939–1947.

[14] Anirban Banerjee, Md Sazzadur Rahman, and Michalis Faloutsos. 2011. SUT: Quantifying and mitigating URL typosquatting. *Computer Networks* 55, 13 (2011), 3001–3014.

[15] Guanchen Chen, Matthew F Johnson, Pavan R Marupally, Naveen K Singireddy, Xin Yin, and Vamsi Paruchuri. 2009. Combating Typo-Squatting for Safer Browsing. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*. IEEE, 31–36.

[16] Fred Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM* 7, 3 (1964), 171–176.

[17] Benjamin Edelman. 2003. Large-Scale Registration of Domains with Typographical Errors. http://cyber.law.harvard.edu/people/edelman/typo-domains/. (Sep 2003).

[18] Benjamin Edelman. 2010. Estimating Visitors and Advertising Costs of Typo Domains. http://www.benedelman.org/typosquatting/pop.html. (2010).

[19] Godai group. 2011. Doppelganger Domains. http://godaigroup.net/wp-content/uploads/doppelganger/Doppelganger.Domains.pdf. (Sept 6 2011).

[20] Tristan Halvorson, Janos Szurdi, Gregor Maier, Mark Felegyhazi, Christian Kreibich, Nicholas Weaver, Kirill Levchenko, and Vern Paxson. 2012. The BIZ top-level domain: ten years later. In *Passive and Active Measurement*. Springer, 221–230.

[21] ICANN. 1999. Uniform Domain Name Dispute Resolution Policy (UDRP). http://www.icann.org/en/help/dndr/udrp. (1999).

[22] Mohammad Taha Khan, Xiang Huo, Zhou Li, and Chris Kanich. 2015. Every second counts: Quantifying the negative externalities of cybercrime via typosquatting. In *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 135–150.

[23] John Klensin. 2008. Simple mail transfer protocol. (Oct. 2008). IETF RFC 5321.

[24] Tyler Moore and Benjamin Edelman. 2010. Measuring the perpetrators and funders of typosquatting. In *Financial Cryptography and Data Security*. Springer, 175–191.

[25] The CALO project. [n. d.]. Enron email dataset. ([n. d.]). https://www.cs.cmu.edu/~./enron/. Last accessed: September 30, 2017.

[26] Peter J Rousseeuw and Mia Hubert. 2011. Robust statistics for outlier detection. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1, 1 (2011), 73–79.

[27] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. 2014. The Long" Taile" of Typosquatting Domain Names.. In *USENIX Security*. 191–206.

[28] Kurt Thomas, Danny Huang, David Wang, Elie Bursztein, Chris Grier, Thomas Holt, Christopher Kruegel, Damon McCoy, Stefan Savage, and Giovanni Vigna. 2015. Framing Dependencies Introduced by Underground Commoditization. In *Proceedings (online) of the Workshop on Economics of Information Security (WEIS)*.

[29] VirusTotal. [n. d.]. VirusTotal - Free Online Virus, Malware and URL Scanner. ([n. d.]). https://www.virustotal.com/.

[30] Thomas Vissers, Wouter Joosen, and Nick Nikiforakis. 2015. Parking Sensors: Analyzing and Detecting Parked Domains. In *Network and Distributed Security Symposium*. http://www.internetsociety.org/sites/default/files/01_2_2.pdf

[31] Yi-Min Wang, Doug Beck, Jeffrey Wang, Chad Verbowski, and Brad Daniels. 2006. Strider typo-patrol: discovery and analysis of systematic typo-squatting. In *Proc. 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*.