2015 IEEE Symposium on Security and Privacy

Every Second Counts: Quantifying the Negative Externalities of Cybercrime via Typosquatting

Mohammad Taha Khan* taha@cs.uic.edu
Xiang Huo* xhuo4@uic.edu
Zhou Li† zhou.li@rsa.com
Chris Kanich* ckanich@uic.edu

*University of Illinois at Chicago †RSA Laboratories

***Abstract*—While we have a good understanding of how cybercrime is perpetrated and the profits of the attackers, the harm experienced by humans is less well understood, and reducing this harm should be the ultimate goal of any security intervention. This paper presents a strategy for quantifying the harm caused by the cybercrime of typosquatting via the novel technique of *intent inference*. Intent inference allows us to define a new metric for quantifying harm to users, develop a new methodology for identifying typosquatting domain names, and quantify the harm caused by various typosquatting perpetrators. We find that typosquatting costs the typical user 1.3 seconds per typosquatting event over the alternative of receiving a browser error page, and legitimate sites lose approximately 5% of their mistyped traffic over the alternative of an unregistered typo. Although on average perpetrators increase the time it takes a user to find their intended site, many typosquatters actually improve the latency between a typo and its correction, calling into question the necessity of harsh penalties or legal intervention against this flavor of cybercrime.**

I. Introduction

Choosing whether or not to use a security feature is fundamentally a tradeoff: will the cost, either monetarily or in terms of decreased usability, outweigh the benefit of being protected from harm due to a certain class of attack? Security flaws and fixes are constantly being discovered and implemented by researchers and practitioners alike. Oftentimes, the benefits of these improvements are quantified by number of bugs found, number of malicious programs detected on victims' computers, or number of stolen credentials discovered on an attacker's drop site. While these are all worthwhile metrics, the ultimate goal of cybersecurity efforts is to protect users from harm: malicious software is no doubt a harm, but how much meaningful impact does it have on a user's daily life? Lessening how much these attacks negatively impacts users' lives should be front and center in evaluating any system that claims to improve the system's security.

The losses due to cybercrime are not always monetary in nature. A stolen credit card can certainly lead to monetary loss, but that is not the only damage caused, as the user loses time updating saved card numbers or requesting a new card. This time could have been spent doing something of the user's choosing, which is likely to be more edifying than talking to a customer service representative. In fact, the suspected astronomically low success rate of cybercrime [1] implies that these losses of time are far more common than actual monetary loss; thus, it is possible that this loss of time is the dominating factor in any analysis of the true costs of cybercrime.

To enable meaningful improvements in cybersecurity, valid metrics include not only how much more difficult a defender has made the attackers' success, but how much the negative effects of these attacks harms legitimate users. To understand this improvement, we must first understand how much harm is coming to these users, not only as lost value but also as wasted effort necessary to clean up after the attacks.

From a technical vantage point, observing direct losses due to cybercrime is challenging: value extraction often does not have tight temporal locality with any particular break in, and it crosses multiple technical systems (HTTPS browsers, cybercriminal credential caches, ATM networks) before the value is extracted. Users' lost time, however, is far more observable: from a network vantage point, it is possible to see when an activity has taken place. Time is a valuable resource, and time stolen or wasted by dealing with a cyber attack is a worthwhile metric which can easily be compared across different types of cybercrime, different users, and different platforms.

Although many types of cybercrime involve compromising a user's machine and installing malware to steal data, there are other more venial crimes committed on the web, like typosquatting. Typosquatting is the registration of Internet domain names which are similar to those of established sites, in the hopes that a user typing that site's name makes a mistake and visits the typo domain rather than their intended target. Typosquatting has not been shown to be particularly harmful from a malware infection standpoint: [2] found that the malware incidence rate for typo sites was lower than for legitimate sites in the Alexa top 1,000,000 site list. Nonetheless, typosquatting certainly causing a modest amount of harm: a typosquatter, knowing that its site's visitor desires to visit a similarly spelled domain name, instead presents ads, a competing site, or sometimes even malware to its visitors. Like spam, this practice also leads to investment in anti-typosquatting products like defensive registration. Even though it is not a major cybercrime, typosquatting has several properties that allow us to precisely quantify the harm being done by the typosquatters: it is observable at the network level; as typosquatting sites often serve no purpose independent of their target, visiting one signifies user intent to visit the targeted site; and finally, with the correct vantage point one can pinpoint how much of each

DOI 10.1109/SP.2015.16

user's time is wasted by typosquatters. We can also determine the damage to website owners by understanding how many potential visitors might not have visited their site because of the confusion caused by the typosquatter's domain.

Contributions

This paper presents a metric for, and empirical quantification of, the harm caused to users due to typosquatting. In service of that goal, we also present a new methodology for identifying typosquatting domains which complements prior work.

User Harm Metric. We present a metric for quantifying user harm in the form of lost time due to cybercrime. While lost time is perhaps the most minor form of harm due to cybercrime, it is certainly the most common and widespread. Not only is it common, but with the correct measurement vantage point and analysis, it can be precisely quantified for a population of users while maintaining their privacy, and compare it to the cost of additional defenses.

Quantification of Harm. Beyond describing and presenting the metric for harm via lost time, we apply this metric to the phenomenon of typosquatting. Even though it is a minor form of cybercrime, typosquatting allows us to perform *intent inference* – as typosquatting sites do not perform any advertisement for their own misspelled domain name, visiting a typosquatting site is a priori equivalent to intending to visit the site being typosquatted upon. Thus, harm can be specifically quantified by two metrics: time lost for users and visitors lost for site operators. As visitors lost per site can have wildly varying revenue loss implications for different sites, we choose to focus on time lost as our main harm metric, and define the same amount of lost time between different individuals as equivalent. This equivalence also aids us in our goal of maintaining user privacy: because any identifying factors about the user are immaterial in the aggregate, we have no use for any personally identifying information and thus user privacy is maintained.

Passive Detection of Typosquatting Domains. In service of our previous contribution, we also created a new methodology for detecting typosquatting domains which boasts fundamentally higher accuracy than previous approaches. With a vantage point at the network level, a combination of DNS and HTTP traffic records allows us to examine all attempted visits to similar hosts, and to use aggregate evidence that visits to a given site are almost always followed by visits to a lexically similar site, without the converse being true. This phenomenon implies that the former site is typosquatting on the latter. To precisely quantify this effect, we introduce a *conditional probability model* for detecting typosquatting domains, which provides a new metric for judging the accuracy of a typosquatting detection: by identifying domains which mainly attract visitors via direct type-in, have a high bounce rate (proportion of visitors who leave the site soon after arriving), and are very often followed by a visit to a more popular site with a similar name, our methodology finds exactly those sites which fit the commonly accepted definition of typosquatting, without needing to worry about coincidentally similar domain names.

II. Background

Traditionally, analyses of user harm are done at a macro level, often by organizations with a vested interest in the results of the analysis. Industry estimates of economic harm that reach the hundreds of billions of dollars per year, very often alongside a link to purchase some sort of anti spam product [3]. We cover more reasoned estimates in Section III, however even those based on available macro-level data estimate the losses to American consumers at $20 billion annually, no small amount [4]. These macro level analyses are certainly useful to determine how much investment should be placed in cybersecurity efforts overall. However, to differentiate among forms of harm or between different perpetrators, these treatments are not sufficiently specific to be of practical use. With a clear metric and a method of observing cybercrime events, we can craft highly granular estimates of loss to users. These metrics can allow us to focus cybercrime research not only on what minimizes the technological impact (e.g. number of malware installs or number of credentials stolen), but also focus on what interventions will have the most positive impact on users' lives.

Most crimes cause harm in multiple ways: for instance, a cybercriminal who steals $100 not only removed $100 from a user's account, the victim then needs to spend additional time fixing whatever flaw the attacker used to gain access to the user's account so that the same attack does not happen again. Let us call this the negative externality of an instance of cybercrime. These can happen even when there is no direct monetary damage, for instance when a user must clean up adware or malware which didn't successfully steal anything. However, it is just as possible that the infection itself is benign from the point of view of the user: if a keylogger steals no credentials or an advertisement hijacker has its ads blocked by an ad blocker, the user has not suffered any material loss.

Harm to victims is the natural counterpart to attackers' successes: for instance, by sending spam, spammers waste others' time and resources without their consent; successful keyloggers can steal users' credentials and possibly money out of their bank accounts, but they can also slow down the victim computer or force the user to waste time cleaning up the infection. Perhaps the purest form of cybercrime is vandalism: harming or inconveniencing others for thrill or notoriety. On the opposite end of the spectrum, the effect of state-sponsored sabotage or espionage is incredibly difficult to detect, let alone value. The "upside" to the attacker in these cases is difficult to quantify, but the negative externalities suffered by Internet bystanders is real, and decreasing it is a noble cause.

Investigating the advantage conferred to the attacker can be particularly fruitful, and can lead to effective interventions which decrease how lucrative a given form of cybercrime is. However, cybercrime is very rarely if ever a zero sum game where the cybercriminal's gain is exactly equal to the user's loss. Cybercrime is perpetrated for different attacker motivations, whether vandalism, economics, or espionage. Fully understanding these motives, especially with respect to the

monetarily motivated cybercriminal, can elucidate new methods to protect users, not by exploiting new technical mechanisms, but by decreasing how lucrative the attack is through other means, like filtering credit card transactions to stem the flow of spam [5]. In the same way, we aim to elucidate new methods for effective defense through a holistic understanding of the specific harms caused by cybercrime.

III. Related Work

Related work generally falls into one of two categories: work meant to quantify user harm and work specifically looking at the phenomenon of typosquatting.

A. Quantifying User Harm

While estimating the loss to end users experiencing typosquatting is a fairly new topic, the cost due to other malicious activities is well studied. Users' money loss due to spam [4], [6], Nigerian scams [7], fake online pharmacies [8], fake antivirus software [9] and general phishing activities [10] have been analyzed. Not only can attackers steal currency, they can also steal CPU cycles: Huang et al. show how bot owners use victim machines to mine bitcoin for profit [11]. Lost capital and computing resources works well for quantifying user harm in these instances, but are not applicable as metrics for quantifying typosquatting's harm because the main harm is lost website visitors (when a user gives up or believes erroneously that they have arrived at their intended destination) and lost time.

More generally, tangible harm to users is notoriously difficult to estimate: financial harm to end users may be marginal due to the incredibly low success rate of cybercrime [1], or overinflated due to the difficulty of conducting accurate surveys [12]. Research on attack mitigation and remediation implicitly focuses on the harm to users in lost time, for instance when investigating the difficulty of cleaning up web server compromises [13]. Having a better understanding of malware infection events and their negative externalities [14] or their epidemiological precursors [15] has given defenders a better view of how harm is experienced by different populations of users.

B. Understanding and Detecting Typosquatting Activities.

In 2003, Edelman first investigated typosquatting [16]. Since then, many approaches were proposed to detect typosquatting activities. Wang et al. designed a system called Strider Typo-Patrol to protect branded domains by monitoring neighboring domains with typos [17]. Similar approaches have been proposed by Banerjee et al. [18], Linari et al. [19] and Chen et al. [20] which all select popular domains and detect their typosquatting counterparts with small lexical distance. Recent work has broadened the investigation of typosquatting: Szurdi et al. [2] investigate typosquatting among less popular sites, Agten et al. [21] study a pool of typosquatting targets over time to look at the dynamics of the phenomenon, and yet others have investigated typosquatting using homophones [22] or bit flips in DNS requests [23]. To complement these efforts, we introduce the intent inference technique which leverages passive data collection to discover typos without requiring a similarity metric for bootstrapping the list of domains to consider.

More generally, intent inference with respect to domain registration within new top level domains has also been studied [24], showing that defensive registration within new top level domains is extensive, and very few of the domains host potentially legitimate content.

Along with the efforts spent detecting this threat, other research has focused on understanding typosquatters' strategy and measuring the damage posed by them. Moore et al. explored their monetization methods and found out that most of them rely on pay-per-click advertisements [25]. Furthermore, the Internet marketing consultancy Fairwinds Partners shows that typosquatting costs brand owners massively [26], but does not investigate end users' costs.

IV. Data Sources

This study makes extensive use of both active and passive measurements. Passive measurements are aggregated from two organizations with complementary vantage points, and active measurements are taken to expand the intelligence available about the suspected typosquatting domains and their targets.

A. Passive Sources

To conduct this study, we collect passive web browsing information from two data sources. The first data source is a set of http and DNS requests for non-existent domains collected at a passive tap on a large public U.S. university's network as part of their network security infrastructure. The HTTP portion of this dataset was anonymized pre-analysis to only include a salted hash of source IP address. DNS requests were collected between the local recursive resolver and the Internet so individual clients' identities are not divulged, and prior to analysis they were filtered to only include "non-existent domain" results. We refer to this dataset as the TAP dataset. Volume statistics regarding the TAP dataset appear in Table I, and the columns are defined as follows: "hosts" refers to the number of unique clients observed in the dataset, "events" includes the raw number of individual HTTP or DNS requests recorded, "visits" includes the number of groups of HTTP requests for HTML files partitioned by five seconds of no traffic to approximate the number of user actions, and "domains" includes the number of unique fully qualified domain names visited or requested.

While this dataset does not include any HTTPS data, during prior active measurement efforts [2] we observed that very few typosquatting websites are available via HTTPS, and thus we expect that their effect on the overall phenomenon will be minimal.

Network operations does not filter this traffic in any way between the university and the Internet. Because of the pre-analysis anonymization of the data, the Institutional Review Board determined this research to not be human subjects research due to its lack of personally identifying information and purely passive collection.

Dataset	Duration	Hosts	Events	Visits	Domains
PROXY	31 days	1.8M	7.39B	78.4M	0.58M
TAP(HTTP)	436 days	468K	7.88B	0.14B	7.8M
TAP(DNS)	219 days	1.3M	2.62B	0.13B	46.4M

TABLE I
VOLUME FOR PASSIVE DATASETS.

The second dataset is a one-month collection of logs of HTTP/HTTPS communications between internal machines at a technology services enterprise and external web sites. The logs are aggregated from proxies deployed at enterprise borders and requests for any non-existent domains are not included. Branches of this enterprise are set up on several continents and therefore this dataset provides us with global view on typosquatting events. While hundreds of fields are presented in each log, our analysis only uses timestamp, source address, destination URL, destination IP, referrer, and HTTP result code. This dataset is leveraged to calculate the conditional probability threshold (see Section VI) and measure user harm. A subset containing timestamp, anonymized source address, and destination domain (exclusive of domains internal to the corporation) was used for the first task. For the latter task, our analysis code is exported to the enterprise and run on one of its internal servers. Only the statistical result is returned and no personally identifiable information is revealed in this process. We refer to this dataset as the PROXY dataset and show volume statistics in Table I.

B. Active Sources

The final dataset is an active web scrape of all domains in the suspected typo or suspected target sets, totaling 13.5K domains. This crawl uses a javascript-aware crawling mechanism that allows recording of redirections through multiple intermediary sites and final rendered web pages, including any javascript based redirection attempts. Using a simplified version of the methodology from [27], this dataset allows us to differentiate between user site interaction and automated redirection, as well as allow us to collect page content for clustering purposes.

V. METHODOLOGY

To achieve the ultimate goal of characterizing user harm, we must achieve three separate sub goals: first, we must collect a set of typosquatting domains in use. Second, we must define a metric to quantify how much time is wasted by each typosquatting domain or each registered domain. Finally, we characterize both the overall harmful effect of typosquatting as well as the effect contributed by different aspects of both user activity and typosquatter behavior. An overview of our methodology is shown in Figure 1.

A. Passive Typosquatting Detection

Previous treatments of typosquatting (see Section III-B) have detected typosquatting domains largely through active investigation: by lexically comparing popular site domain names with all registered domains, one can find millions of

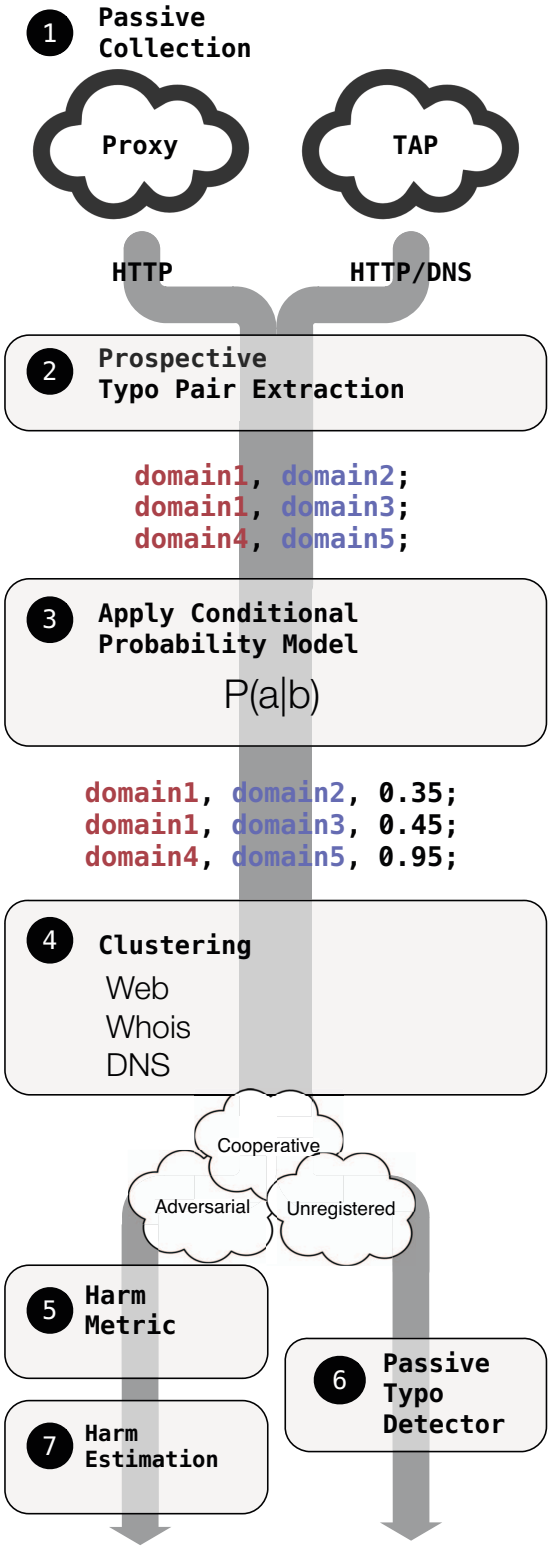

Fig. 1. Methodology overview.

potential typosquatting domains. However, without actively visiting these sites and characterizing the content presented, one cannot infer the true intent of the registrant. In many cases, further investigation shows a large portion of those sites are indeed sites with little unique content that are likely set up only to maximize economic return on user typos. However, active measurement by third parties cannot determine ground truth about these sites' typo status: for instance, without prior knowledge about its legitimacy, one might mistake `nhl.com` as a typo for `nfl.com`, as both sites offer content related to sports.

Although actively collecting site content and comparing sites with lexical similarity can detect typosquatting, there are two main shortcomings: the lexical model based on edit distance might not capture all user typos, and it cannot capture typos which are not yet registered. Due to computational constraints, we focus mainly on the second weakness. Typos of domains which are not yet registered are key to our analysis: while an analysis which sums up the amount of time wasted by visiting typosquatting sites would be able to characterize the extent of the problem, it would be unnecessarily pessimistic. A proper analysis must compare the time wasted by typosquatters to the time wasted by viewing a "website unavailable" error page in the browser: the overall time wasted by typosquatting is the difference between time lost to seeing the error page and time lost due to seeing the typosquatter's page, as the typo would happen either way. In fact, it is altogether possible that typosquatting as committed by at least some subset of the transgressors is a net positive for users: they may arrive at their intended destination more quickly after viewing the typosquatter's page compared to the browser error page.

The core of our typosquatting detection algorithm is passive detection of possible typosquatting events.h Through observing user activity at the http and dns request level, we are able to seed a list of typo domain and target domain pairs without actively inspecting their content (which is useful in the case of unregistered domains, as they have no content). While we validate our methodology using active techniques like web scraping and manual inspection, the combination of passive detection of typo pairs and the conditional probability model provides us with an orthogonal metric which measures the core facet of typosquatting: the fact that visits to lexically similar sites were unintended by the user, and thus were likely the result of a typo.

1) Conditional Probability Model: To bootstrap passive typosquatting detection, we search our datasets for all events that indicate directly typing a domain into the browser's URL bar: either a load of the root resource, or a DNS lookup that results in an "non existent domain" response. We only search for non existent domains in the DNS dataset because loads for existent domains will be served by higher fidelity HTTP level data. HTTP data is better for this purpose because requests are visible at the individual client level rather than the level of the recursive resolver, and do not suffer as much due to caching. Although negative results are also cached by recursive DNS resolvers, we expect that typos for all but the most popular sites will not be served by the negative result cache.

Once we have the list of candidate type-in events, we transform this set into all pairs of domains with a Damerau-Levenshtein edit distance of one which might have been performed by the same user within a reasonable amount of time, such that they are likely causally related. Our key insight to passively detect typosquatting is that the conditional probability of visiting the target domain will be much higher for typosquatting domains than for unrelated domains which just happen to have a small edit distance between them. That is, when a user visits a typosquatting domain, even if that domain does succeed in diverting the user to a different page, the intent to visit the target page still exists, and thus in the aggregate the chance that a user will eventually find their true intended site will be higher after a visit to a typo site than to an unrelated site. We validate our methodology by sampling a subset of site pairs at different conditional probability levels, and choose a conditional probability cutoff at sufficiently high accuracy (see Section VI-A1). We show the overall accuracy results of using our conditional probability model in Section VI-A2.

One challenge here is choosing some reasonable bound for the distance in time between two site visit events. For choosing the ideal time window to reflect true typo instances, we use an optimization scheme. We first parse our logs to extract all possible typos that occurred within 90 seconds and divide them into *N=9* ten-second bins. For each bin, we cumulatively calculate its accuracy A_k, for up to the k^{th} bin under consideration. An inherent constraint of our scheme requires the aggregation of all items in previous bins that occurred before the k^{th} bin. The optimization of the time window is based on the samples S_i in each bin, the fidelity a_i, which is the percentage of HTTP based typos in a bin and the overall loss in accuracy L_i as a cost of including that bin. Equation 1 below elaborates our accuracy calculation.

$$A_k = \sum_{i=1}^{k} \frac{a_i S_i}{\sum_{j=1}^{N} S_j} - L_i \tag{1}$$

We evaluate the accuracy of each of the 9 bins and find an optimal cutoff between the third and the fourth bin at approximately 33 seconds. This is defined as our optimal time window for observing both HTTP and NX domain based typos.

B. Harm Inference

We define harm in this case as a combination of time lost when attempting to visit the intended site, and visitors lost due to not reaching their intended site. Again, while trivial in comparison to monetary losses due to an instance of fraud or data loss due to vandalism, this form of loss is likely the most common among all cybercrime losses, and when summed up could cause substantial loss of time (in the case of lost time due to typos) or new visitors/customers (in the case of intended visitors who never visit the site they attempted to visit).

Choosing a specific definition for harm is fundamentally difficult: harm could just as easily be a number of seconds

lost, an interruption, or even the installation of malware or the loss of a password to a phishing site. We disregard the latter two as prior work has found only a vanishingly small amount of malice in the typosquatting ecosystem, less in fact than in the Alexa top million sites themselves [2]. Because our model focuses only on typo events, all users will be experiencing an interruption, and the time lost will be the difference in how many additional seconds of delay the user experiences between visiting the typo site and visiting their intended site. Although a raw number of seconds is easy to compare between two users, the cognitive load of that interruption, or its context, might not be the same between two different sites or two different users, or two different events: for instance, a lost second shortly before a paper deadline might be worth far more to a user than a lost second shortly after the deadline. To standardize our analysis, we assume that different seconds at different times or among different users are all equally valuable.

After using the conditional probability model to detect typosquatting domains, we can use the same set of visit pairs to quantify harm to site visitors and site operators. For lost time due to typosquatting, in essence we are comparing two distributions: the delay distribution of all typo pairs where the initial domain is registered by a typosquatter, and the distribution of all visit pairs where the initial visit was a "non-existent domain" DNS response.

C. Typo Event Characterization

Overall, a typo event can have one of a few different general classes, which we enumerate here. A visualization of typo events and their possible outcomes appears in Figure 2.

1) *Adversarial registration.* This is the activity most commonly thought of as "typosquatting:" registration of a lexically similar domain by an unaffiliated party which does not link directly to the target domain. In many cases, these sites are full of ads, and could advertise a competing service similar to the target domain's. In some cases, these will host malware.
2) *Cooperative/defensive registration.* This activity exists where a brandholder or an entity operating on its behalf registers a domain, and redirects users to their (presumably) intended target domain.
3) *Unregistered.* Mistyped domains can also simply not yet be registered. When DNS is not tampered with by service providers, attempting to visit these sites in modern browsers results in a "web page not available"/"server not found" error message from the browser. It is then the user's responsibility to interpret this generic message and find the domain they were attempting to visit.
4) *Unrelated visit.* Correlated loads may also simply be spurious false positives. We describe our methodology to detect spurious correlations during the evaluation of passive typo detection (Section VI-A2).

When characterizing these different types of typosquatting events, our main metrics are time lost and visits lost. For the "unregistered" type of typo event, all events should be the same - no matter which unregistered domain a user types in, the error page will look exactly the same. Likewise, for cooperative registration, the user might see an error page or be immediately redirected to the domain they wish to visit. Adversarial registration, however, will be affected by the content of the page: while a page full of generic ads may signal to a user that they have not found their desired page, a lookalike or competitor page advertising a similar service might greatly increase the time spent finding the intended page or the chance the user gives up without visiting their intended site. Characterizing this distribution will both allow us to present an accurate overall estimate for the harm caused by typosquatting, as well as single out different entities contributing to the harm caused by the phenomenon.

Furthermore, for each typo event which does not cause a visit loss, there are several avenues through which the user can find their desired site:

1) **Redirection**: perhaps the most simple, the lexically similar site can automatically redirect the user to their intended destination. Redirects using HTTP return codes can be followed in our analysis using the `Location:` header to confirm the redirection, and javascript based redirections can be followed using the `Referer:` header. We rely on our web scrape to detect javascript redirects because our vantage point in the network cannot differentiate between an automated redirection (which our scraper will follow) and user-initiated navigation (which our scraper will not cause).
2) **Direct navigation**: the user might find the site via correcting their mistake directly in the URL bar, which can be determined via the lack of a `Location:` header or `Referer:` header.
3) **Search engine**: the user can also search for the site on a search engine to verify correct spelling. While major search engines are now using HTTPS to secure their traffic, the `Referer:` header is still set to the domain of the referring site for Google and Bing, which account for the vast majority of all search traffic.
4) **Other navigation**: beyond search engines, users can still find the site through other means: either by clicking through links directly on the typo site they visit, or referring to some other non-search engine reference page.

Combining the HTTP request headers in our passive datasets and intelligence from the scraping dataset, we can identify which of these modalities the user used to find their intended web site, and furthermore we can determine how much each mode affects the overall time delay. As each of these implies a user successfully finding their intended site, visitor loss rate is not relevant for this analysis.

D. Typosquatter Characterization

Within the class of adversarial registrations, there are several different perpetrators, all attempting to maximize their own revenue via their investment in typo domains. Our hypothesis is that properties of the typo site will influence the visit loss rate and the time delay distribution. For much of this analysis, we follow the general methodology of Levchenko et al. [5].

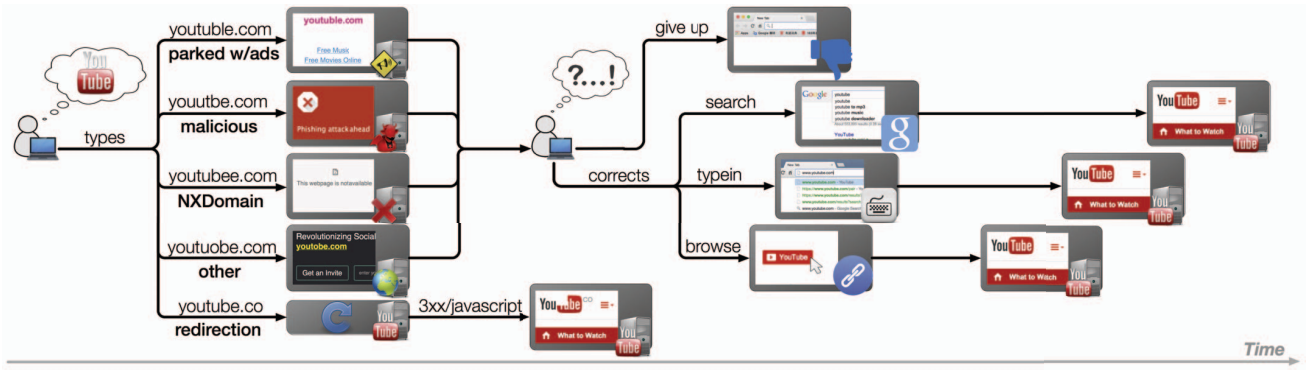

Fig. 2. Typosquatting event visualization. When a user mistypes their desired domain, one of many different events may happen. In the optimistic case the user is redirected to the correct site. Otherwise, the user will be redirected to several different types of sites, and has various options for finding their intended site.

Our simplifying assumption is that, if we can infer common operation via clustering of site properties (page structure, whois information, or infrastructure like DNS or HTTP hosting), all sites run by the same entity will contribute similarly to delay and visitor loss.

VI. Results

Our results are split into three main sections: how many registered typo domains were detected; the time loss and visitor loss metrics we use to quantify harm; and a characterization of the different typosquatting operations by active and passive measurements, as well as their individual effect on the loss metrics.

A. Passive Detection

Overall, we consider loads or attempted loads of the root resource for 36.7M unique fully qualified domain names. We include requests for nonexistent domains as attempts to load the root resource of said domain. Detailed data regarding these loads is seen in Table I, however this value is lower because of the data sanitization tasks outlined in Section VI-B. Using records of the form *(domain, anonymous user id, timestamp)*, we generate pairs of domains (d_1, d_2) such that each load was performed by an individual user within 33 seconds of each other, and the Damerau-Levenshtein edit distance between the two domains is one. After this filtering step, we arrived at 61,274 unique pairs of domains.

1) Pair filtering: While many of these pairs of correlated, domains like `nhl.com` and `nfl.com`, for which one is unlikely to be a typo of another even though they are lexically similar, will no doubt show up in this dataset. Thus, we apply the conditional probability model as described in Section V-A1. Figure 3 presents the distribution of conditional probabilities as such: for all pairs of prospective typos, we graph the probability that a user visits the latter domain after visiting the former. Domain pairs like the aforementioned sports league domains are unlikely to be correlated: in our dataset, a request for `nhl.com` is only followed by a load of `nfl.com` .08% of the time, and the reverse rate is even lower, below 0.01%. However, visits to the site `eba.com` are followed by visits

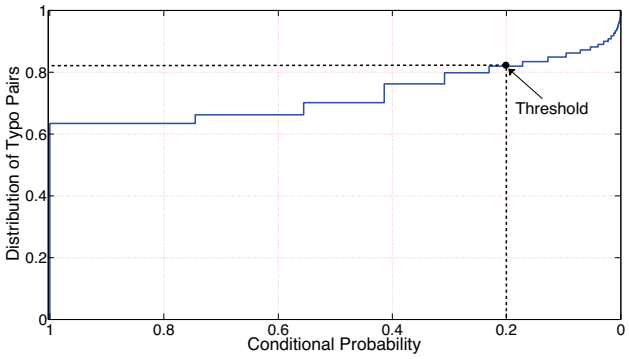

Fig. 3. Conditional probability distribution for user intent modeling.

to `ebay.com` 90% of the time. Thus, visits to `eba.com` are likely to be typos.[1]

To evaluate the accuracy of the conditional probability model, we manually inspect a random sample of the typo pairs using a methodology similar to that in [2]. We segment the data into ten bins and manually inspect 20 samples from each bin. As shown in Figure 4, there is a distinct drop in accuracy at 20% probability, which we set as our threshold for typo detection. Because the volume in the high accuracy bin at 90% or above holds the lion's share of all prospective typos, the overall accuracy at our chosen threshold is 86.5%.

2) Evaluation: Before using our passive typosquatting detection approach to quantify harm, we must first validate the approach itself. Here, we compare the results of our algorithm with that of prior work which takes an active approach to identifying typosquatting.

Obviously, passive detection will detect a much lower absolute number of domains than previous approaches: methods relying on zone file inspection will see all domains registered in a given DNS zone and can use a set of all small edit distance domain pairs as prospective typos. However, just as obviously,

[1] `eba.com` was the highest non-zero probability adversarial registration in our dataset, however upon manual inspection it appears to be unrelated to the target site. We discuss this effect in Section VII-C.

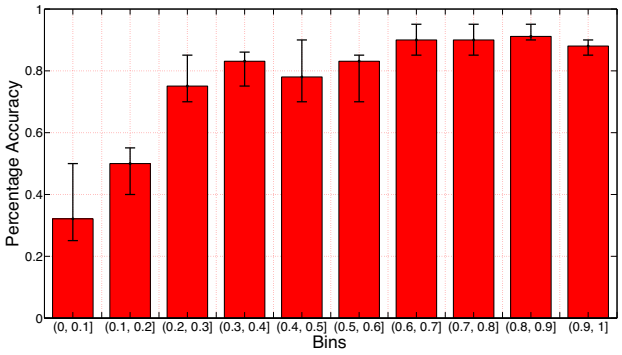

Fig. 4. Sampled typo accuracy per conditional probability threshold.

active detection cannot detect how many mistypes happen on non-existent domains, nor can they see how common they are or how much time delay happens between the typo lookup and the intended site lookup. Active detection also cannot find domains outside of the given zone, as the zone file for other top level domains are not shared with researchers; they would need to resort to active crawling of the domain name space for all possible edits. In fact, we find that 5% of all typos perpetrated by users were outside of the same TLD.

Furthermore, at high conditional probability thresholds, the accuracy of the passive detection method is 95%, compared to the accuracy of prior work like [2] which achieves 86% accuracy for random typo samples. The conditional probability model allows a typo detection or defense mechanism to tune a confidence variable based on how often loading a prospective domain correlates with a given target. Also, as this method is complementary to prior work, combining the approaches could improve both coverage and accuracy.

B. Data Sanitization

Beyond the conditional probability model of user behavior, we performed several other data cleaning tasks to arrive at a more accurate estimate of the underlying phenomenon. Several classes of web requests can fit our filters correctly, but are actually artifacts of phenomena besides users mistyping domains. Here we list the heuristics we used to remove those requests from our dataset.

We preprocessed the initial set of typo pairs to remove out a fair amount of false positives in the data. A majority of these instances were domains having an edit distance of 1 in the subdomain. A common occurrence in the dataset was the `www` subdomain followed by a redirection to a `ww1` subdomain. For instances of domain pairs that had two words such as `hello-world.com` followed by `helloworld.com` would appear as a typo pair although the additional - was not a result of user mistyping but rather a user misunderstanding. To cater these situations, we excluded all typos that had an edit distance of 1 as a result of a different digits in the subdomain or a difference of a hyphen in the typo and target domain strings.

Another efficient heuristic that we applied aimed to filter out pairs that had subdomains differing by letters with small alphabetic distances between each other. This eliminated instances such as `a.example.com` followed by a request to `b.example.com`. As a result of applying these heuristics, the number of true typos went considerably up to 63% of typo pairs having a conditional probability of 1.

As our DNS logs lacked user level granularity, we also filter out extremely popular sites as we cannot be confident that the DNS requester and the HTTP requester are one in the same. Because in our dataset a "non-existent domain" result could have been elicited by any user, pairs consisting of a non-existent domain and a sufficiently popular domain, we cannot be sure that the same user elicited both of those requests, e.g. because a third party loaded the popular site before the user who mistyped their target. Thus, choose a threshold for domain popularity and discard all pairs for which we cannot be suitably sure that both events in the pair come from the same user. The popularity metric was defined as the number of times different users requested the same domain within our optimized time window. To evaluate domain popularity we used a representative subset of our HTTP logs. Using that subset to evaluate popularity, we only considered unregistered domain typos for which their respective targeted domains had a 20% or lesser chance of appearing twice within our optimized time window. In other words, because a DNS request for an unregistered typo could be generated by any user within our user base, we only include those for which there is a 80% or greater chance that there was only one visitor to that domain within the next 33 seconds.

As a part of the sanitization process we were able to classify defensive domain registrations in the "other navigation" category of cooperative registrations in Table III. This also provides an explanation for a subset of our cooperative typos incurring larger delays than a normal automatic redirection. In the case of defensive registrations, the landing page contained a message of an "Invalid URL" that caused the user to either retype the domain or follow a link on that website. The most mistyped subdomains were `wordpress`, `tumblr`, and `blogspot` URLs, that were either nonexistent blogs or benign pages the user did not intend to visit. As these domains were registered with the same registrar as their target domains, we classified them as cooperative registrations as opposed to adversarial.

C. Quantifying Harm

User time loss is the first metric we quantify. Harm cannot be evaluated in a vacuum: the lost time due to a typosquatter's activities should be compared against alternatives like the domain being cooperatively registered or unregistered. In Table II, we quantify the proportion of each class; in Figure 5 we also present the cumulative distribution of delay incurred by events in each class.

In the aggregate, we can describe the time lost due to typos in the absolute, as well as relative to what could have happened otherwise. Among all three classes of typos, users at the TAP location lost 1,205,060 seconds, or just under two weeks

Domain class	Cooperative	Adversarial	Unregistered
Unique domains	2.4K	11.1K	20.9K
Unique visits	40.7K	40.4K	67.4K
Average delay (s)	2.87	9.58	10.38
Average loss (%)	3.30	16.81	11.53

TABLE II
DOMAIN VOLUME, VISIT VOLUME, AND HARM BY DOMAIN CLASS.

of time. Of course, this is not the effect of typosquatting, but simply the overall effect of mistyping domains. To compute the actual harm caused by typosquatters, we can form two different estimates: one where we compare the delay experienced due to adversarial registration to the delay caused by a domain being unregistered, and another where we compare the adversarial delay with the cooperative delay. These, respectively, form the lower bound on time lost where the defender spent no extra resources on otherwise unregistered domains, and an upper bound on time lost where the defender registers *all* possible typos that users ever type. Computing these bounds, we see that the lower bound is approximately -8.98 hour and the upper bound is approximately 75 hours. A negative value for the lower bound provides us with the insight that on average, adversarial domains actually help users reach their intended websites faster than if that domain is left unregistered. This is certainly not the expected result, but hardly a surprising one: because the dataset is so large, less than nine hours difference between one effect and another is so low as to be roughly equivalent, leading us to hypothesize that in the common case, users correct their error in roughly the same amount of time whether they see a "server unavailable" error message or unexpected content.

However, in the upper bound case where those domains had all been registered by the average cooperative entity, the adversarial registrants would have caused 75 hours of delay throughout the course of our dataset. Even though this is a sizeable loss summed over a few seconds here or a few seconds there, one should consider it in the context of all web usage. As we do not have an accurate estimate of the time spent using the web on this network, we instead create a very rough estimate from facts about the campus and U.S. web use. With a very conservative estimate of an average of 10,000 people (less than one third of total enrollment and academic staff) on campus on a typical day using the web for .7 hours per day (half of the U.S. average of 1.40 hours per day [28]), this factors out to 175 man-years of web use over the course of our full fidelity dataset (219 days). Using even the upper bound on time wasted by typosquatting, this factors out to 4.22 seconds wasted due to typosquatting per 24 hours of web use. Although this is an infinitesimal figure, it will hopefully be of value when performing comparative analyses with other forms of cybercrime.

1) Characterizing Harm: Within each domain class, four methods for arriving at the intended site are possible: being redirected automatically by the web server or page code, performing a search for the intended domain via a search engine, directly correcting or re-typing the domain into the URL bar, or browsing to the intended page. The proportions of each discovery method are shown in Table III; the distribution of delays associated with each discovery method is shown in Figure 6.

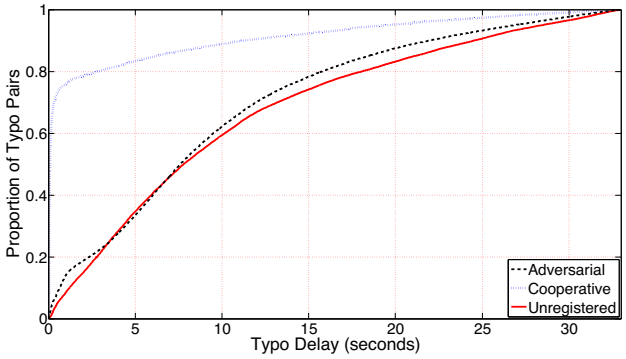

Fig. 5. Cumulative distribution of visit delay among cooperative registration, adversarial registration, and unregistered domains.

Discovery Method	Redirect	Search	Typein	Browse
Cooperative	38712	N/A	N/A	2045
Adversarial	N/A	1797	38403	243
Unregistered	N/A	11604	54760	1088

TABLE III
DISCOVERY METHOD BY DOMAIN CLASS (IN NUMBER OF VISITS).

Even before splitting the adversarial registrations into different classes, the overall delay caused by either unregistered visits or visits to adversarial registrations is greater than cooperative registrations: that is, defenders can save their visitors time by registering popular typos and forwarding their visitors to the correct domain name. However, in the absence of this defense, we actually find that the overall delay caused by adversarial registrations is *less* than that when a visitor types in an unregistered typo. This evidence is a first hint that the phenomenon of adversarial typosquatting taken as a whole actually saves users time rather than wasting their time.

Discovery method has a significant effect on the delay between visiting a typo and finding the resulting site. However, other factors can also influence the amount of time it takes a user to find their intended site. In Figure 7, we partition the data between "mobile" users (on tablets and phones) and "traditional" users on non-mobile devices within the TAP dataset. To perform the partition, we consider an event as mobile if the string "Android," "iPad," or "iPhone" exists in the HTTP request's User-Agent string (which covers most modern mobile devices); anything else is considered a "traditional" browser. For finding typos by correcting a typo in the address bar (right side subfigure), we see a minimal difference between the four modalities split up by mobile/traditional and unregistered/adversarial domains. When searching, however, unregistered domains are uniformly easier to correct with a search, likely an effect caused by pre-resolution which we discuss in Section VI-G1. The traditional laptop and desktop computing environments where

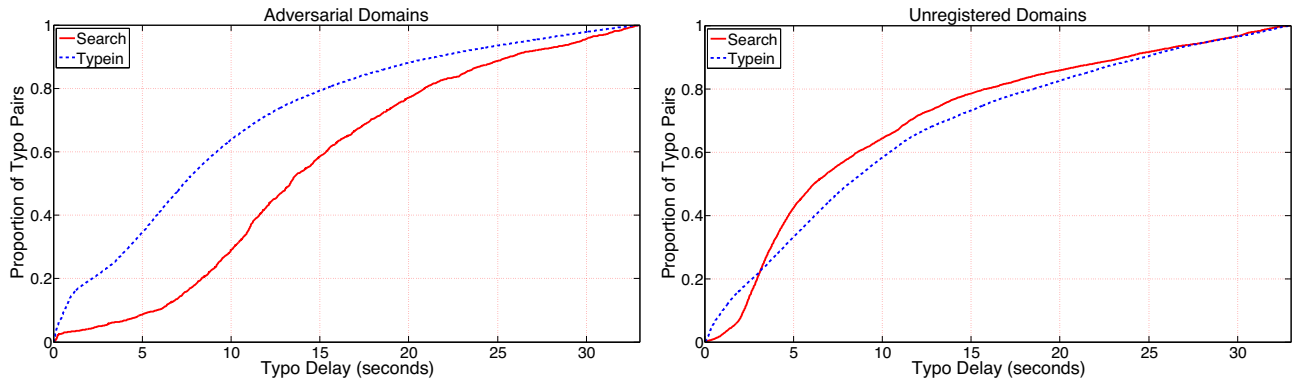

Fig. 6. Cumulative distribution of visit delay by discovery method.

user actions like navigating to the address bar or typing on the keyboard are measurably faster in each modality, however there is only a slight edge when manually correcting a typo.

D. PROXY Dataset Measurements

To validate our approach on the TAP dataset, we repeated our analysis for the PROXY dataset. This population had a nearly distinct set of typos–of the 5,722 registered domain pairs present in the PROXY dataset, only 300 overlapped with the TAP dataset. That this network is only meant to be used for work purposes (and not recreational uses e.g. by students in dorms or between classes) can partly explain the sparsity of this overlap. Furthermore, the PROXY dataset was collected from a corporation with a global presence, and the TAP dataset includes only users at a large public university on the East coast in the United States. Even so, Figure 8 mirrors the effect seen in Figure 6 for the TAP dataset (and likewise for the more generic domain class Figures 9 and 5), showing that the distinct delay characteristics of the different discovery methods outweigh any effect caused by current task or user population.

E. Malicious Typosquatting Domains

To determine what proportion of adversarially registered domains were suspected of malicious intent, we used the VirusTotal and Google Safe Browsing scanners to test whether any of these domains were blacklisted. Among all 11.1K adversarial domains seen in the dataset, 33 were listed on VirusTotal (with 3 or more detections) and 9 on Google Safe Browsing (all listed as malware), either indicating low detection rate, or low incidence of actionable malice on typosquatters. This finding agrees with the results of [2] which actually found that domains in the Alexa top 1,000,000 were more likely to be marked malicious than detected typo domains.

F. Typo Characterization

Beyond focusing on the content of the sites, we can also characterize the typos people make. First among these questions is whether the most popular sites receive the most typos, or do less popular sites also receive typos? To investigate, we graph the Alexa rank of the target domain against the number

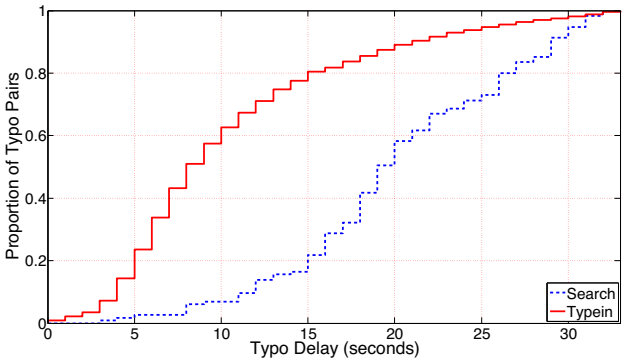

Fig. 8. Cumulative delay distribution by discovery method for the PROXY dataset.

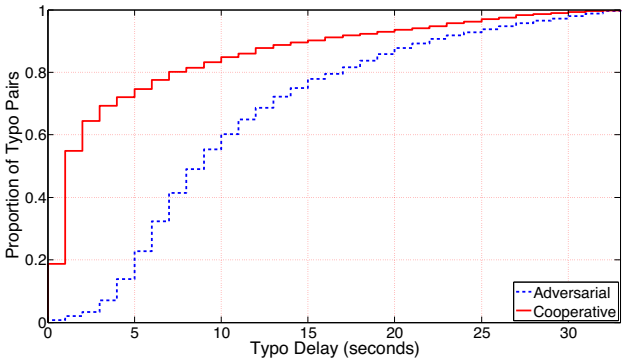

Fig. 9. Cumulative delay distribution by domain class for the PROXY dataset.

of visits in Figure 10. Perhaps unsurprising, the lion's share of typos happen on the 100,000 most popular sites. However, there is a distinct long tail effect, with a full 15% of all typos targeting sites with rank below 1,000,000.

Drilling down to local popularity, we gauge site popularity based on the number of visits to the site from the local campus population. In Figure 11, we see that while the most popular sites are subject to a moderate amount of typos, it is the middle

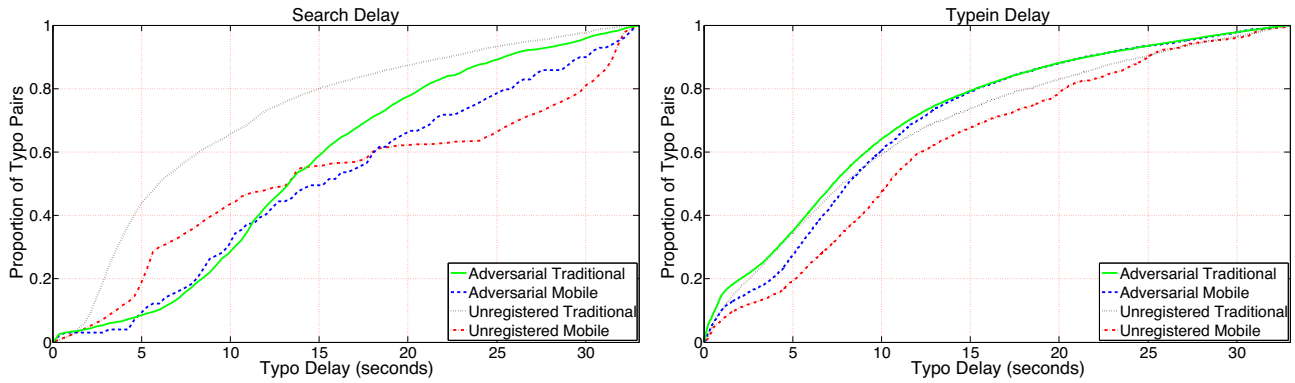

Fig. 7. Cumulative delay distribution by device type.

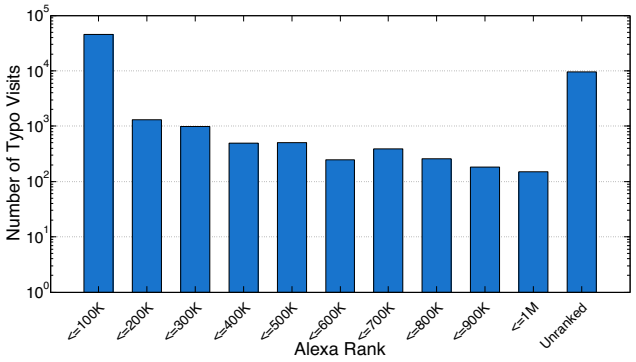

Fig. 10. Number of typo visits by target site's Alexa rank.

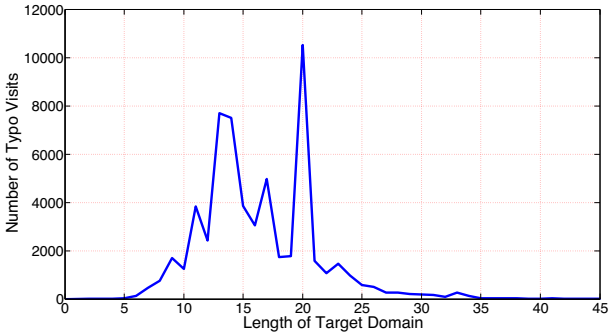

Fig. 12. Number of typo visits by target site's domain name length.

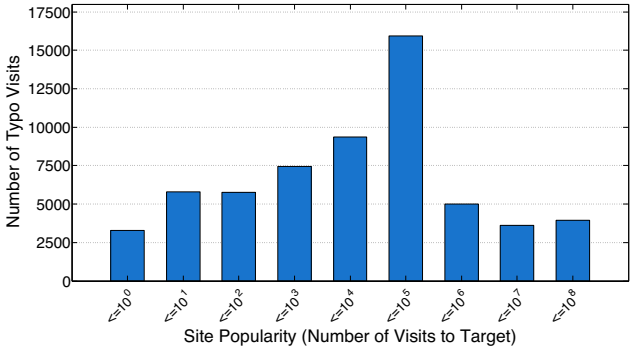

Fig. 11. Number of typo visits by target site's local popularity.

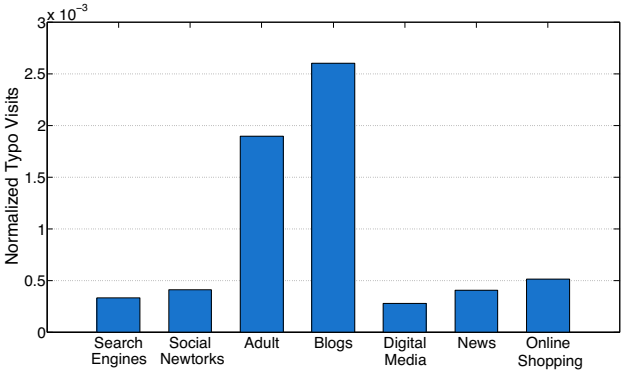

Fig. 13. Number of typo visits by target site's category.

of the overall popularity distribution that sees most typos (there are, of course, far more sites in the 100,000 visit range than in the 100,000,000 visit range).

We can also investigate whether longer domain names are more likely to be mistyped, as shown in Figure 12. This figure shows a smooth peak around length 12, in line with Alexa's top 100,000 having an average length of 13 and the entire 1,000,000 having an average length of 15. The outlier at length 20 is `ratemyprofessors.com`, no doubt a popular site on a college campus. Furthermore, the most popular typo is `ratemyprofessor.com`, which is not only an edit distance one typo, but also a very reasonable semantic mistake.

Finally, we can also inspect typos by their site category, as determined via `urlblacklist.com`. Search engines are higher than one might expect here as a typo, even if they are one of the most useful tools online. Our hypothesis here is that users in search of a given site will first manually type in the name of the search engine directly to the URL bar, then search for the URL in the search engine, and then finally go to the search result. Not only does this boost the number of visits

Count	Registrar name
1080	GODADDY.COM LLC
666	ENOM INC.
603	FABULOUS.COM PTY LTD.
561	.CO Internet S.A.S.
435	NETWORK SOLUTIONS LLC.
393	TUCOWS DOMAINS INC.
374	PDR LTD. D/B/A PUBLICDOMAINREGISTRY.COM
357	INTERNET.BS CORP.
338	NAMEKING.COM INC.
336	Public Interest Registry

TABLE IV
TOP TEN REGISTRIES FOR ADVERSARIAL TYPO DOMAIN NAMES.

to the search engine, typo correction provided by the search engine prevents many other typos that might have happened during the search query.

G. Typosquatter Characterization

Not all typosquatters cause the same effects: here we explore the different typo page clusters and their effect on visit delay. In this section we focus on adversarial registrations to detect and characterize individual organizations perpetrating typosquatting.

1) Infrastructure Clustering: First, we cluster the adversarial typosquatting based on network features like WHOIS registrant, DNS provider, and hosting provider.

Delay clustering. One interesting phenomenon to discuss here is the large proportion of incredibly quick corrections for unregistered domains as seen in Figure 6: 30% of users find their destination domain via correcting their typo within five seconds. A reasonable explanation exists however: modern browsers pre-resolve domain names as they are typed by end users, thus a user who types `example.com` slowly enough will cause their browser to attempt to look up `example.co` to speed up the eventual page visit.[2] As shown in Table IV, the `.co` registrar is among the most popular registrars for registered typos, and its presence in Figure 14 (as the outlier in the upper left hand corner) shows that it is likely that these domains are being looked up as part of pre-resolution of `.com` domain names rather than as actual user typos.

As shown in Figure 6, it seems unreasonable to expect that over 40% of all users who correct an unregistered domain typo via search do so in under five seconds. Closer inspection of browser operation provides a clue here: the address bar in modern browsers provides both address entry and search engine functionality. The browser cannot a priori tell the difference between someone searching for a string with no spaces or simply entering a host name. Thus, for example, if a user types a domain name with a nonexistent TLD, the browser can default to searching for the desired domain. Because modern search engines provide spelling correction, the correct domain is likely to be the first result on the search engine results page, only a click or tap away.

Beyond domain registrar information from whois, we can also inspect the name server responsible for resolving the typo

[2]This effect, along with its privacy implications, is investigated in [29].

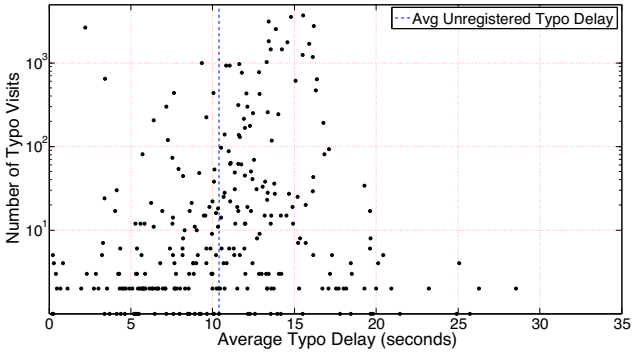

Fig. 14. Scatterplot of average typo discovery delay and number of typos, grouped by domain registry.

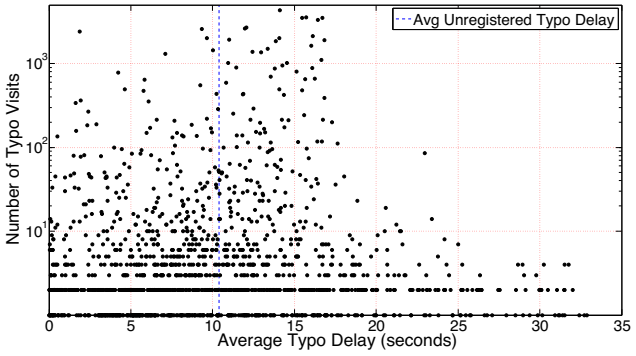

Fig. 15. Scatter plot of average typo discovery delay and number of typos, grouped by DNS provider.

domains; the top ten DNS providers are listed in Table V and a scatter plot of number domains and average delay is presented in 15. While there are no distinct trends to point out here, the heavy hitters are all well known domain parking providers; as the monetization and hosting mechanisms involved in domain parking are fundamentally very similar to those used in typosquatting, this is an unsurprising result.

One interesting point to notice is that, if these clusters represent different typosquatters, all perpetrators whose average delay is below 10.38 seconds are actually *helping* the user save time by getting them to their destination faster. Because the difference between the average delay due to typosquatting and the average delay due to a browser error page is so small, it is not hard for typosquatters to actually improve over that average.

Success clustering. Just as different typosquatters cause users to take more or less time to find their intended destination, so do different typosquatters cause a varying amount of harm to the intended destinations: Figures 16 and 17 plot the intended site discovery rate against the quantity of visits grouped by registrar or DNS provider cluster. Note that due to our 20% threshold on the conditional probability model, there is a hard cutoff on loss rate. While there are no pronounced effects

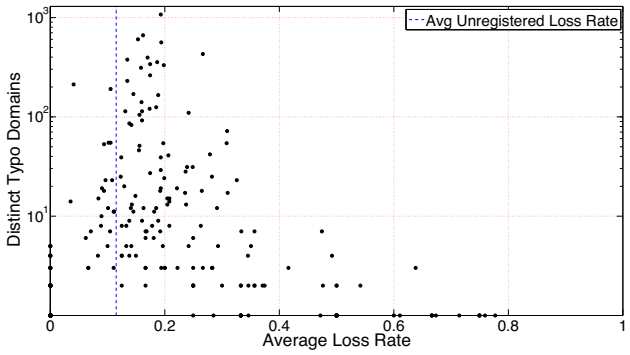

Fig. 16. Scatterplot of average typo discovery success and number of typos, grouped by registrar.

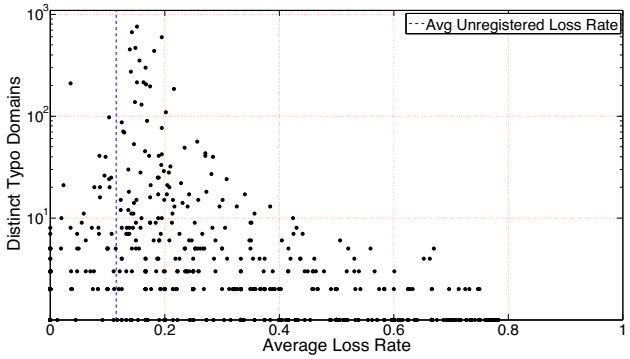

Fig. 17. Scatterplot of average typo discovery success and number of typos, grouped by DNS provider.

here, we can see that a healthy amount of clusters (including some with high volume) average a success rate above that of unregistered domains, showing that not only can some typosquatters lead their visitors to their intended destination faster, they also have a higher success rate.

Count	Nameserver
757	dsredirection.com
671	above.com
593	domaincontrol.com
466	parkingcrew.net
453	internettraffic.com
439	sedoparking.com
351	dnsnameserver.org
301	hastydns.com
272	rookdns.com
215	parklogic.com

TABLE V
TOP TEN NAMESERVERS (AGGREGATED BY REGISTERED DOMAIN) FOR ADVERSARIAL TYPO DOMAIN NAMES.

VII. DISCUSSION

A. Limitations

Note that the passive dataset only has HTTP data, and no HTTPS data. When correlating pairs of typo and target domains, any target domains which are visited directly via their HTTPS URL will be invisible to our collection apparatus. Thus, we will miss some successful typo corrections and slightly under-count the successful discovery rate for target sites which are found in this manner, and the delay from this typo event will not be factored into the average delay.

As stated earlier, a strength of this approach is that it could be used for typos which differ by more than Damerau-Levenshtein distance one. However, the volume of traffic we inspected made it infeasible to search for typos which differ by more than one character.

Because the conditional probability model allows the detection of typos based on user intent rather than active inspection of the sites in question, the coverage of all typos will be far less than an exhaustive, active methodology. In addition, our offline methodology requires inspecting a massive amount of user traffic. While these are weaknesses in using the conditional probability model as an online typo detection approach, the additional data regarding user harm would not be available without passive data, and would not enable analysis of the negative externalities of this form of cybercrime.

B. Revenues and Negative Externalities

To estimate the negative externality ratio for typosquatting, we must convert our time and visit loss into dollars and cents, as well as make estimates for the costs and revenues of typosquatters. Here we use reports of online advertisement prices and revenue from Johnston [30], [31] as very rough estimates of costs and revenues for site visits and ad clicks.

For visit gains and losses, "cost per click" advertising can be used as a proxy for the value of a visit which is "successful," and "cost per impression" advertising can be used as a proxy for an unsuccessful visit i.e. a visit to a typosquatter's site that does not cause a visit to the intended site. Thus, we can estimate the value of the revenue to the typosquatter as $r_c\alpha + r_i(1-\alpha)$ where r_c is the revenue per click, r_i is the revenue per impression, and α is the visit loss rate.

We can estimate that visit loss is zero-sum: visitors who wish to perform a given action should be "worth" equivalent revenue whether they end up at the legitimate site or the typosquatter's site. Thus, loss to the intended site owner is $c_c\alpha$ (the cost per click times the visitor loss rate), as this would be the cost to the owner to otherwise attract the lost visitors. Note here that c_c is the cost per click to the advertiser rather than r_c, which is the revenue to the publisher (the ad broker retains the spread between these two values); we make the estimate of a 50/50 split in the case of both c_c and c_i again based on [30], [31]. We ignore any negative effect of visit delay on the revenues of the site owner. Using our average net visit loss rate of 5.28%, for every thousand typos which end up at an adversarial typosquatter's site, the intended site

owner loses $48.58 and nets the typosquatter $2.70. Note that this is the marginal loss for the intended site and marginal revenue for the typosquatter: we ignore the fixed costs of the typosquatter including domain registration and hosting, as well as any difference in the marginal costs incurred (or not incurred) due to computational resources used for each visit.

Considering only the values due to visit loss or gain, the negative externality ratio would be 18:1, in line with the estimates for non-violent crime [4]. However, the time loss is also significant: for every 24 hours spent browsing the Internet, a user loses 4.22 seconds to typosquatting. Because statistics for web use are readily available, we focus on the United States for these computations. An Internet user who browses the web for 1.40 hours a day five days a week would lose 64 seconds to typosquatting per year; using the median hourly wage of $16.87 in the United States as a proxy for the value of this user's time [32], we arrive at a per capita loss of $.29 due to lost time.

We can also estimate the externality ratio of the defenders (who purchase domains to prevent typosquatters from doing the same) and the attackers (who also have to purchase domains to ply their craft). In our entire traffic trace, we see 11.1K adversarial domains and 2.4K cooperative domains. Assuming that the cost of of domain registration is roughly equivalent for different registrars, the ratio between attacker effort and defender effort is 4.62:1, far lower than nonviolent crime. This effect implies that typosquatting is likely here to stay: if the profits from typosquatting are sufficient (and, unless this investment was largely undertaken with speculation in mind, we can assume it is), then the 2.5K unregistered domains experiencing typo traffic in our dataset indicate that there is certainly sufficient fertile ground upon which to perpetrate yet more typosquatting, without even accounting for newly popular domains for which typos have not yet been registered. Although this analysis neglects the registrations which were not visible in our dataset, it does serve as a metric for the amount of successful effort each party has put in to perpetrating or defending against typosquatting.

Due to the numerous "fudge factors" in each estimate, these numbers should be taken with a healthy dose of caution. However, the final negative externality ratio is approximately 18:1 (the lost time does not significantly effect the calculation), showing that typosquatting is far less societally damaging than spam based on Rao's estimate of 100:1 [4] and in line with other nonviolent crime. Dealing in absolute values, the USC "How Much Media" estimates that approximately 160 billion hours are used annually browsing and searching the web [28]: thus, typosquatting wastes 262 man-years of time, or $38 million in lost "productivity" using the median hourly wage cited above. Although as a headline value that certainly appears large, in the context of a gross domestic product approaching 17 trillion dollars, this value is vanishingly small. In addition, this is using the upper bound for typosquatting, which presupposes that defenders would register every typo in our dataset, which is somewhat unrealistic. In the case where site owners do not perform defensive registration, there would actually be a *savings* of 31 man years of time, which likewise corresponds to over $4.5 million in gained "productivity." As these estimates come down on either side of helpful or harmful, perhaps it is best to consider typosquatting squarely in a gray area with no clear-cut benefit or harm to society.

This analysis can also guide site owners in a proper course of action. Because the loss rate and delay for users arriving at a given site after mistyping it is roughly equivalent between an unregistered domain and a typosquatting domain, site owners should not register typos because they fear typosquatters will further delay or steal their visitors. The upper bound implies that one can decrease delay and loss of visitors with defensive registrations; however, these defensive registrations are effectively defending against the act of mistyping, rather than the act of typosquatting.

C. Future Work

Extensions to our methodology and analysis could further explore typosquatting via the conditional probability model and harm inference. Chief among them is generalizing typo detection so that it does not rely on Damerau-Levenshtein edit distance. In theory, the conditional probability model should be able to find all typos, not just those with small edit distance. We attempted to apply the conditional probability model to all root document loads instead of just those with a small edit distance, but the accuracy metric (as described in Section VI-A1 for the short edit distance version) was unacceptably low, and the runtime was far larger. To improve this approach, we might include heuristics for detection from other typosquatting work, or incorporate the information used for clustering (Section VI-G) during the detection step rather than only at the pair clustering step.

While passive typo detection can infer visitor intent, it cannot detect site operator intent: as in the `eba.com`/`ebay.com` example, not all common typos are obviously typosquatters monetizing someone else's brand, even if they benefit from it. Because of this, intent inference should not necessarily be used as a method to block typo domains, but rather could be used as a typo suggestion similar to the approach taken by search engines do when they detect typos. Future work could perform more extensive analysis of the bounce rate from these domains, or the content (or lack thereof) of the pages being hosted to build a more confident gauge of the owner's intent.

Fully quantifying the negative externalities of cybercrime can show defenders a better picture of how harmful these activities are to the Internet's human users. However, a comparative treatment would be even more useful. The user harm metric of lost time could be extended to other forms of cybercrime like spam, fake antivirus, or ransomware. The latter two scams have direct financial components, but the time spent performing remediation is a significant component of the loss incurred. Identifying and removing malware can be a frustrating and time consuming process: more complex intent inference, perhaps aided by a search engine query stream, could enable quantifying the human component of harm caused by malware.

User harm and passive DNS observation can also be combined to quantify the effects of recursive resolvers performing NXDomain wildcarding. NXDomain wildcarding is the practice of operating a recursive name server and returning a result for domains which would normally have none; the entity controlling that server can then serve ads based on the user's typo [33]. While this activity is certainly a compromise of the integrity of the DNS system, its effect on users is unclear: on one hand, users who see a page of ads and search results might be able to find their intended destination more quickly than a user who sees a browser error page. On the other hand, the page of ads and possibly competing search results might decrease (or, possibly, increase) visitor loss for the intended destination site. Recall from Figure 6 that unregistered domains have a high rate of intended site discovery below 5 seconds possibly because of browser URL bar search behavior: if this effect extends to wildcarding "search" sites, these sites' questionable behavior could actually be a net win for the user. Verifying this circumstantial evidence would show that while NXDomain wildcarding is hostile toward the integrity of the DNS, it might not be hostile to user experience.

VIII. Conclusion

This paper's ultimate goal is to characterize harm to users, and uses the time wasted by typosquatting to quantify the harm caused by an individual flavor of cybercrime. While overall typosquatting's negative externalities add up to 262 man-years of time lost in the United States per year, its externality ratio is much lower than spam's and is in line with other forms of nonviolent crime. When compared against the alternative of an unregistered domain's error page, the harm caused by typosquatting overall is no longer as clear cut, as some perpetrators actually help users achieve their goals more quickly than others. More generally, we show that it is possible to precisely quantify the harm to users via one strain of cybercrime, a measurement which should allow researchers and practitioners to efficiently allocate their effort when deciding which cybercriminal enterprises to combat.

IX. Acknowledgements

We thank the anonymous reviewers and our shepherd Nicolas Cristin for their thoughtful comments and assistance in improving the paper. We also thank our data partners, including George Mason University, and those who assisted in the collection and anlysis of the data, including Damon McCoy, Angelos Stavrou, and Chaitanya Yavvari. This work was made possible by National Science Foundation grant CNS 1351058.

References

[1] D. Florêncio and C. Herley, "Where do all the attacks go?" in *Economics of Information Security and Privacy III*. Springer, 2013, pp. 13–33.

[2] J. Szurdi, B. Kocso, G. Cseh, M. Felegyhazi, and C. Kanich, "The Long "Taile" of Typosquatting Domain Names," in *Proceedings of the 23rd USENIX Security Symposium*, 2014.

[3] Nash Networks Inc, "Strangling the internet: Why spam matters and what it costs," http://www.nashnetworks.ca/why-spam-matters-and-what-it-costs.htm, 2009.

[4] J. M. Rao and D. H. Reiley, "The Economics of Spam," *The Journal of Economic Perspectives*, pp. 87–110, 2012.

[5] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Félegyházi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, D. McCoy, N. Weaver, V. Paxson, G. M. Voelker, and S. Savage, "Click Trajectories: End-to-End Analysis of the Spam Value Chain," in *Proceedings of the IEEE Symposium and Security and Privacy*, 2011.

[6] C. Kanich, N. Weaver, D. McCoy, T. Halvorson, C. Kreibich, K. Levchenko, V. Paxson, G. M. Voelker, and S. Savage, "Show Me the Money: Characterizing Spam-advertised Revenue," in *Proceedings of the USENIX Security Symposium*, Aug. 2011.

[7] C. Herley, "Why do nigerian scammers say they are from nigeria?" *WEIS*, 2012.

[8] D. McCoy, A. Pitsillidis, G. Jordan, N. Weaver, C. Kreibich, B. Krebs, G. M. Voelker, S. Savage, and K. Levchenko, "Pharmaleaks: Understanding the business of online pharmaceutical affiliate programs," in *Proceedings of the 21st USENIX Conference on Security Symposium*, ser. Security'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 1–1. [Online]. Available: http://dl.acm.org/citation.cfm?id=2362793.2362794

[9] B. Stone-Gross, R. Abman, R. A. Kemmerer, C. Kruegel, and D. G. Steigerwald, "The Underground Economy of Fake Antivirus Software," in *Tenth Workshop on Economics of Information Security (WEIS)*, June 2011.

[10] C. Herley and D. Florencio, "A profitless endeavor: Phishing as tragedy of the commons," in *Proc. New Security Paradigms Workshop*. Association for Computing Machinery, Inc., September 2008.

[11] D. Y. Huang, H. Dharmdasani, S. Meiklejohn, V. Dave, C. Grier, D. McCoy, S. Savage, N. Weaver, A. C. Snoeren, and K. Levchenko, "Botcoin: Monetizing Stolen Cycles," in *Proceedings of the Network And Distributed Security Symposium (NDSS)*, 2014.

[12] D. Florêncio and C. Herley, "Sex, lies and cyber-crime surveys," in *Economics of Information Security and Privacy III*. Springer, 2013, pp. 35–53.

[13] M. Vasek and T. Moore, "Identifying risk factors for webserver compromise," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, vol. 8437. Springer, March 2014. [Online]. Available: http://lyle.smu.edu/~tylerm/fc14rf.pdf

[14] ——, "Do malware reports expedite cleanup? An experimental study," in *Proceedings of the 5th USENIX conference on Cyber Security Experimentation and Test*, ser. CSET'12. Berkeley, CA, USA: USENIX Association, 2012. [Online]. Available: http://lyle.smu.edu/~tylerm/cset12.pdf

[15] T.-F. Yen, V. Heorhiadi, A. Oprea, M. K. Reiter, and A. Juels, "An epidemiological study of malware encounters in a large enterprise," in *Proceedings of the 21st ACM Conference on Computer and Communications Security*, ser. CCS'14, Nov. 2014.

[16] B. Edelman, "Large-scale registration of domains with typographical errors," http://cyber.law.harvard.edu/people/edelman/typo-domains/, Sep 2003.

[17] Y.-M. Wang, D. Beck, J. Wang, C. Verbowski, and B. Daniels, "Strider typo-patrol: discovery and analysis of systematic typo-squatting," in *Proc. 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, 2006.

[18] A. Banerjee, M. S. Rahman, and M. Faloutsos, "SUT: Quantifying and mitigating url typosquatting," *Computer Networks*, vol. 55, no. 13, pp. 3001–3014, 2011.

[19] A. Linari, F. Mitchell, D. Duce, and S. Morris, "Typo-squatting: The curse" of popularity," in *WebSci'09: Society On-Line*, 2009.

[20] G. Chen, M. F. Johnson, P. R. Marupally, N. K. Singireddy, X. Yin, and V. Paruchuri, "Combating typo-squatting for safer browsing," in *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*. IEEE, 2009, pp. 31–36.

[21] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis, "Seven months' worth of mistakes: A longitudinal study of typosquatting abuse," in *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015)*, 2015.

[22] N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, and W. Joosen, "Soundsquatting: Uncovering the use of homophones in domain squatting," in *Information Security - 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings*, 2014, pp. 291–308.

[23] N. Nikiforakis, S. Van Acker, W. Meert, L. Desmet, F. Piessens, and W. Joosen, "Bitsquatting: Exploiting bit-flips for fun, or profit?" in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. Republic and Canton of Geneva, Switzerland:

International World Wide Web Conferences Steering Committee, 2013, pp. 989–998.

[24] T. Halvorson, K. Levchenko, S. Savage, and G. M. Voelker, "Xxxtortion?: inferring registration intent in the. xxx tld," in *Proceedings of the 23rd international conference on World wide web*. International World Wide Web Conferences Steering Committee, 2014, pp. 901–912.

[25] T. Moore and B. Edelman, "Measuring the perpetrators and funders of typosquatting," in *Financial Cryptography and Data Security*. Springer, 2010, pp. 175–191.

[26] FairWinds Partners LLC, "The cost of typosquatting," FairWinds Partners LLC, Tech. Rep., 2010. [Online]. Available: http://www.fairwindspartners.com/our-resources/perspectives/the-cost-of-typosquatting/

[27] C. Neasbitt, R. Perdisci, K. Li, and T. Nelms, "Clickminer: Towards forensic reconstruction of user-browser interactions from network traces," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 1244–1255. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660268

[28] J. E. Short, "How Much Media? 2013," http://classic.marshall.usc.edu/assets/161/25995.pdf, 2013.

[29] S. Krishnan and F. Monrose, "Dns prefetching and its privacy implications: When good things go bad," in *Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*. USENIX Association, 2010, pp. 10–10.

[30] M. Johnston, "What Are Average CPM Rates in 2014?" http://monetizepros.com/blog/2014/average-cpm-rates/, 2014.

[31] ——, "How Much Can Publishers Make from Taboola and Outbrain?" http://monetizepros.com/blog/2014/which-sponsored-content-network-is-best-taboola-outbrain/, 2014.

[32] Bureau of Labor Statistics, U.S. Department of Labor, "Occupational Employment Statistics," www.bls.gov/oes/, 2014.

[33] N. Weaver, C. Kreibich, and V. Paxson, "Redirecting DNS for Ads and Profit," in *USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, San Francisco, CA, USA, August 2011.