PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration

Shuang Hao* Alex Kantchelian† Brad Miller§ Vern Paxson†◇ Nick Feamster‡
*University of California, Santa Barbara †University of California, Berkeley
§Google, Inc. ◇International Computer Science Institute ‡Princeton University
shuanghao@cs.ucsb.edu {akant,vern}@cs.berkeley.edu
bradmiller@google.com feamster@cs.princeton.edu

ABSTRACT

Miscreants register thousands of new domains every day to launch Internet-scale attacks, such as spam, phishing, and drive-by downloads. Quickly and accurately determining a domain's reputation (association with malicious activity) provides a powerful tool for mitigating threats and protecting users. Yet, existing domain reputation systems work by observing domain *use* (*e.g.*, lookup patterns, content hosted)—often too late to prevent miscreants from reaping benefits of the attacks that they launch.

As a complement to these systems, we explore the extent to which features evident at domain registration indicate a domain's subsequent use for malicious activity. We develop PREDATOR, an approach that uses only time-of-registration features to establish domain reputation. We base its design on the intuition that miscreants need to obtain many domains to ensure profitability and attack agility, leading to abnormal registration behaviors (*e.g.*, burst registrations, textually similar names). We evaluate PREDATOR using registration logs of second-level .com and .net domains over five months. PREDATOR achieves a 70% detection rate with a false positive rate of 0.35%, thus making it an effective—and early—first line of defense against the misuse of DNS domains. It predicts malicious domains when they are registered, which is typically days or weeks earlier than existing DNS blacklists.

CCS Concepts

•Security and privacy → Intrusion/anomaly detection and malware mitigation; •Networks → *Network domains;*

Keywords

Domain Registration; Reputation System; Early Detection.

1. INTRODUCTION

The Domain Name System (DNS), the Internet's lookup service for mapping names to IP addresses, provides a critical service for Internet applications. Attackers, however, abuse the DNS service to direct victims to Web sites that host scams, malware, and other malicious content [18, 53]. To mitigate these threats, operators routinely build reputation systems for domain names to indicate whether they are associated with malicious activity. A common mode for developing reputation for DNS domain names is to develop a *blacklist* that curates "bad domains". A network operator who wishes to defend against an attack may use a domain blacklist to help determine whether certain traffic or infrastructure is associated with malicious activity.

Unfortunately, curating a DNS blacklist is difficult because of the high rate of domain registrations and the variety of attacks. For example, every day around 80,000 new domains are registered in the .com zone, with a peak rate of over 1,800 registrations in a single five-minute interval. Establishing domain reputation is thus a continual, ongoing process that must be automated, based on the features of the DNS domains. Existing DNS reputation systems primarily use characteristics of DNS lookups to distinguish legitimate domains from malicious ones [3, 4, 5]. Other systems have derived domain reputation by crawling Web pages to discover malicious content [35, 55]. Unfortunately, these existing systems have a number of limitations. (1) The particular resources and vantage points they employ (*e.g.*, URL crawlers, spam traps) provide limited visibility into various attacks across time, and thus may miss malicious activities of certain domains. (2) Existing systems derive reputation for domain names primarily *after* malicious activity is already underway, delaying detection. By the time existing reputation systems add a domain to a blacklist, the domain may have already been used in widespread abuse (*e.g.*, spam or phishing campaigns, drive-by downloads). (3) Because existing reputation systems cannot prevent malicious domain registrations, attackers continue to register new malicious domains even as their old domains become blacklisted, to sustain scam campaigns.

The ideal time to establish the reputation of DNS domains is thus *at registration time*, before the miscreants can profitably use them. Towards this goal, we design **PREDATOR** (Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration), a proactive reputation system that can accurately and automatically identify malicious domains at time-of-registration, rather than later at time-of-use. Such a proactive reputation can enable early detection of potentially malicious DNS domains, thus benefiting many stakeholders: (1) Network operators can take appropriate actions to protect their networks. For example, email servers can preemptively greylist [34] (*i.e.*, temporarily reject) emails that contain suspicious newly-registered domains and request re-delivery after a period (by which time the network operators can collect more evidence to make final decisions, such as examining the content that the domain hosts). (2) Registries or registrars can require more strict documentation or verification (*e.g.*, validation of payment instruments) before they approve registrations of domains with low predicted reputation. (3) Law enforcement and

CCS'16, October 24–28, 2016, Vienna, Austria

ISBN 978-1-4503-4139-4/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2976749.2978317

security professionals can prioritize investigation efforts and take down malicious sites early [51]. For example, in 2010, LegitScript[1] and registrar eNom have announced an agreement to devote efforts to taking down rogue Internet pharmacy domains [8].

PREDATOR is based on the intuition that, to make domain purchase as economical as possible miscreants must register large quantities of domains—typically in bulk—to ensure that they can remain agile as individual domains are blacklisted or taken down [33, 52]. The timing and targets of attacks may vary, but the domain registrations nonetheless exhibit telltale signatures, including the types of domains that they register and the temporal registration patterns they exhibit, that both distinguish them from benign registration behavior and are relatively non-malleable, due to the fact that attackers require access to large numbers of inexpensive domains to operate successfully.

Although a time-of-registration DNS domain reputation system offers many potential benefits, developing such a reputation system is difficult. Unlike other DNS reputation systems that can observe the ways a domain is used in practice (*e.g.*, by observing lookup patterns or content that the domain hosts), a reputation system that operates at registration time has much more limited information available for developing reputation. We identify features based on (1) the delegated registrars and the hosting infrastructure, (2) structural characteristics of the name itself, (3) previous registration history, and (4) correlations with registration bursts. Another challenge is the lack of full ground truth to evaluate performance; thus, our best hope is to compare the reputations that PREDATOR derives to those of existing blacklists. Of course, the reputation information in existing blacklists is not only late, it is also often incomplete. We also assess how PREDATOR can detect non-blacklisted but spam-related domains by sampling and manually checking those domains.

Our results show that PREDATOR can accurately determine the reputation of new domains with a low false positive rate (0.35%) and a good detection rate (70%). Although the performance is not perfect, *the benefits of establishing domain reputation at registration time are clear*: (1) PREDATOR can achieve early detection, often days or weeks before existing blacklists, which generally cannot detect domain abuse until an attack is already underway. The key advantage is to respond promptly for defense and limit the window during which miscreants might profitably use a domain. (2) As a first line of defense, PREDATOR can reduce the suspect domains to a pool of 3% of all new domains, while capturing 70% of the domains that will subsequently appear on well-known blacklists. Thus, our system enables prioritizing domains for subsequent, detailed analysis, and to find more malicious pages given a fixed amount of resources (*e.g.*, via URL crawlers or human-involved identification). (3) We show that PREDATOR can complement existing blacklists and capture additional domains hosting spam-related content that other blacklists miss.

We make the following contributions:

- We develop an approach to establish domain reputation based on the features evident at the time of domain registration, which provides early detection of potentially malicious domains.
- We identify and encode 22 features that help distinguish domain registration behavior characteristic of abuse from legitimate registration behavior, 16 of which have not been identified or analyzed in previous work.
- We incorporate these features into a state-of-the-art supervised learning algorithm and implement a prototype version of our proactive time-of-registration domain reputation system, PREDATOR.
- We perform a comprehensive empirical evaluation of PREDATOR using five months of logs of second-level .com and .net domain registrations, demonstrating its effectiveness for complementing existing blacklists.

[1]LegitScript is a service to verify and regulate online pharmacies, recognized by the National Association of Boards of Pharmacy.

2. BACKGROUND

The registration process of second-level domains involves three major participants: *registrants* (persons or companies that seek to acquire domain names), *registrars* (*e.g.*, GoDaddy), and *registries* (*e.g.*, Verisign). A registrant usually applies online to a registrar, which is an organization accredited by ICANN to contract with registries to sell domains. The registrar represents the registrant in submitting registration requests to the registry, which manages the relevant top-level domain (TLD) [25]. The registry allocates the domain name, adds the registration data to a central database, and publishes the DNS records in the top-level domain nameservers. The updates operate in close to real time [22, 23] and have a short interval between registration requests and domain activation in the zone. For every domain registration, the registrar charges the registrant and pays a fee to the associated registry. The annual cost for registrants to register a .com domain ranges from about $8.50 to $25 [45, 46]. The registry in turn pays a fee to ICANN for new domain names in the TLD zone.

The registration time periods range from one to ten years. Upon domain expiration, if registrants choose to renew, the domains remain in the zone. Otherwise, the domain expires, is removed from the zone, and becomes available for registration again. We categorize domains as *brand-new* (registered for the first time) or *re-registration* (the domain has previously appeared in the zone, and now a registrant registers it again after its expiration). We further characterize re-registration domains as *drop-catch* (re-registered immediately after its expiry) or *retread* (some time has passed since its previous removal from the zone). We can obtain information regarding expiring domains via registries or third-party sites [42, 59].

3. CASE STUDY: SPAMMER DOMAINS

As an illustration of how spammers exhibit distinctive registration behavior, consider the following instance at registrar Moniker. Figure 1 shows the counts of .com domain registrations from registrar Moniker on one day of March 2012. The x-axis gives the hour of the day (Eastern time). The y-axis shows the count of .com domain registrations for every five-minute epoch. The black bars indicate the counts of domains subsequently appearing on blacklists (including Spamhaus [49], URIBL [56], and a spam trap that we operate), while the white bars are the numbers of domains that are not blacklisted later. These instances provide insight into the characteristics of spammer domain registrations, such as: How many domains are registered together?; What do the names look like?; and How long do the existing blacklists take to block those domains? We briefly explore these questions using a case study.

(1) *Domain registrations occur in bursts*: Table 1 shows the detailed statistics of the five registration spikes in Figure 1. A five-minute epoch may have tens or even hundreds of domains registered for later spam activities. Presumably miscreants exploit the cheaper prices and management ease of bulk registration provided by registrars. For example, when registering over 100 .com domains, Moniker offers a 5% discount (as well as a 25% discount on privacy protection) [41], and GoDaddy provides a 36% discount (lowering the annual price from $12.99 to $8.29) [17]. Because miscreants likely often operate

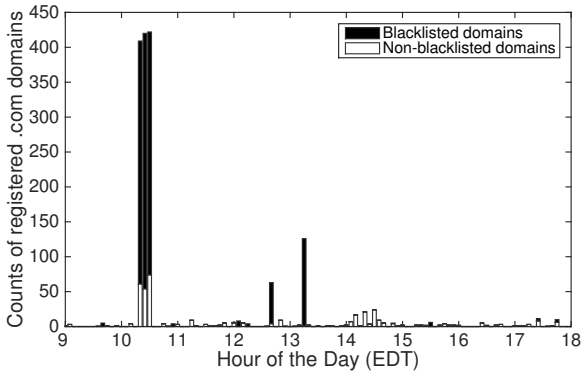

Figure 1: Domains registered by `Moniker` every 5 minutes on a day of March 2012. Each bar represents the count of .com domains registered per 5-minute epoch.

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

Table 1: 5-minute epochs with spammer domain registrations from `Moniker`. The epochs correspond to the five registration spikes in Figure 1.

on thin margins and rely on volume for profitable operation, these discounts are significant.

(2) *Domains registered together are often at similar stages in the domain life-cycle*: Spammer domains in the same spike tend to exhibit the same life-cycle type, brand-new (*i.e.*, first-time registration) or retread (*i.e.*, re-registration with some time after previous expiration), as shown in Table 1. The registration spikes during 10:15–10:30AM (retread), and those during 12:35–1:15PM (brand-new) are likely from different attackers. This phenomenon indicates that in addition to making up names themselves, miscreants track domain names that are due for expiration in the `WHOIS` database (available from publicly released sources [42, 59]) and collect recently expired names in bulk.

(3) *Domains registered together may be similar to one another*: When we take a closer look at the spammer brand-new domains registered in the same epoch of 1:10–1:15PM, we observed that many names appeared similar to each other. Table 2 displays examples of names sharing the same substrings. We highlight the common substrings in bold. The names algorithmically generated by miscreants manifest characteristic lexical patterns.

Table 2 shows how many days pass until various spammer domains are blacklisted; in some cases, these delays may be several weeks, or even several months. Yet, the characteristic registration behaviors that we have outlined provide opportunities to detect and blacklist these domains earlier than existing techniques (*i.e.*, at registration time) because their registration behaviors are so distinctive. The following sections outline our general approach, as well as how we extract and encode the relevant features for establishing domain reputation.

4. PREDATOR ARCHITECTURE

We design PREDATOR to infer each domain's reputation immediately after registration. The decision process does not need to examine the Web content on the domains or wait until users are exposed to attacks. Our goal is for PREDATOR to act as a first layer of defense

Domains (highlight common strings)	*Blacklist delay*
ask**lender**home.com	92 days
askhome**lender**snow.com	51 days
ask**lender**shome.com	32 days
askhomes**lender**.com	24 days
askhome**lender**.com	12 days
askhome**lender**s.com	6 days
ask**lender**today.com	5 days
financilsart.com	122 days
financilss.com	71 days
financilsssky.com	19 days
financilsspro.com	18 days
financilspro.com	17 days
financilssart.com	9 days
financilssky.com	7 days
strokecarebeat.com	65 days
strokecaregreen.com	14 days
strokesoft.com	11 days

Table 2: Example of brand-new spammer domains registered in one single epoch (1:10–1:15PM EDT, per the bottom row of Table 1) from `Moniker`. The domain names with common strings are grouped and ordered by the blacklist delay.

to mitigate malicious URLs or domains that host spam-advertised sites.

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

Training mode. Based on the domain registration information, we extract three types of statistical features:

- *Domain profile features* (Section 5.1) focus on the current registration. The features can be derived from the public `WHOIS` information and the domain names.
- *Registration history features* (Section 5.2) are based on previous registration history. The features can be acquired from third-party services such as DomainTools [10], or they are available at registrars and registries.
- *Batch correlation features* (Section 5.3) examine the domains registered from the same registrar and within the same epoch. The information is available at registrars or registries.

We use prior knowledge to label a set of known spammer and non-spammer domains. With the labeled domains, the learning module takes the extracted features and uses a supervised learning technique to build a classifier. Section 6 describes the design of PREDATOR's classification approach in detail.

Operation mode. Upon a new domain registration, we extract the corresponding features and incorporate them into the classifier. The classifier assigns a reputation score by aggregating the weights learned in the training mode. If a domain is registered to launch malicious activities (*e.g.*, spam campaigns), we expect to assign a low reputation score. On the other hand, we want to assign a high score if a domain is for legitimate Internet services. If the score is lower than a threshold, we generate a detection report to flag the domain as malicious. Network operators or users can take advantage of the

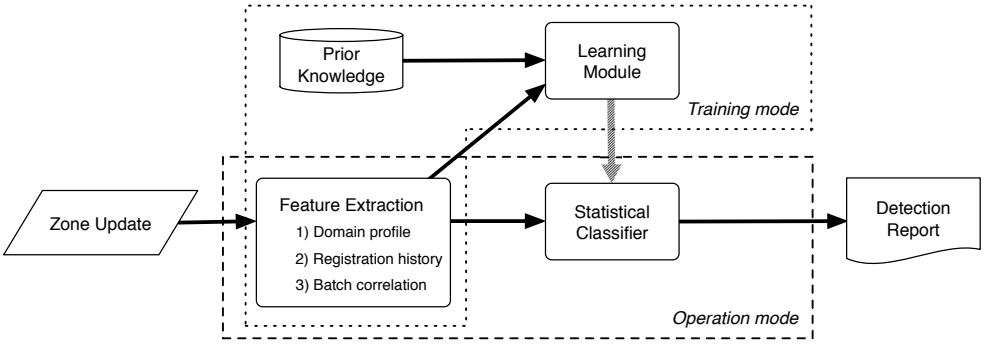

Figure 2: A high-level overview of PREDATOR**.**

early warning to limit the time that miscreants might profitably use a domain for malicious activities (in some cases, they may prevent the attacks from occurring in the first place).

5. IDENTIFYING CHARACTERISTIC FEATURES

We now expand on our earlier observations, exploring other characteristic domain registration behavior of miscreants and encoding these behaviors into features. We categorize the features into three groups: domain profile features, registration history features, and batch correlation features. Each feature is *categorical*, *continuous*, or *ordinal*. Categorical features typically arise when representing features that are nominal in nature. Such features do not present a meaningful notion of order, as values represent logically separate concepts. Continuous values naturally correspond to continuous-range type of features, and ordinal features denote the countable ordered ones (like integers). Most non-boolean features are of these two types.

Table 3 summarizes the features that PREDATOR uses. The first column shows the feature categories, the second describes the detailed features, the third indicates whether the features are categorical, continuous, or ordinal, and the last column lists previous work that has analyzed the feature. Out of 22 features, 16 have not been previously considered for registration-based detection; none have ever been incorporated into a classifier for time-of-registration reputation. We now describe each of these features in detail, discussing both the domain knowledge and intuition we used to identify the feature as well as how we ultimately encoded it. We observe similar patterns from .com and .net zones across different time periods (2012 and 2015, respectively).

5.1 Domain Profile Features

Domain profile features are those that we extract from the domain name or from the public `WHOIS` information regarding the current registration.

Registrar (categorical). End users need to select the delegated registrars to register domains. Miscreants presumably choose particular registrars due to the prices or policies from different ones. We find 79% of .com spammer domains were registered through only ten registrars. Similarly, 84% of .net spammer domains were attributed to ten registrars (five of the registrars overlapped). We map registrars to a group of binary features. A given dimension is set to 1 if and only if the corresponding registrar hosts the domain. There are 906 such categorical registrar features. Since a domain has a single designated registrar; only one such feature is set to 1 and the others are 0.

Authoritative nameservers (categorical). Without authoritative nameservers, people could not resolve the domains in the zone. The nameservers indicate how miscreants manage their domains (after choosing registrars). Miscreants may use self-resolving nameservers, where the nameserver is responsible for resolving its own domain (*e.g.*, the nameserver for `example.com` is `ns.example.com`, as opposed to `ns.third-party.com`) [13]. Although some other nameservers belong to major hosting companies that often host legitimate domains, they provide a finer-granularity indication of spammer domain registrations. The nameserver assignment usually happens close to domain registrations and might change with time. We collect authoritative nameservers for the domains within five minutes of domain registrations (*i.e.*, the epoch that we have defined) from the zone update files and map them to a set of binary features, where 1 means the nameserver resolves the domain. Because a domain could have multiple nameservers or nameserver changes, more than one attribute could have value 1.

Nameserver IP addresses and ASes (categorical). Multiple nameservers can resolve to the same IP address, and different IP addresses can originate from the same Autonomous Systems (ASes). Both IP addresses and ASes indicate underlying hosting infrastructure, which provides a means for identifying portions of the Internet with a higher prevalence of hosting spammer domains. We convert them to a group of binary attributes. Similar to the nameserver feature, more than one attribute could have a value of 1. In our experiment, we derive the IP addresses of the nameservers from the zone update files. For a newly registered domain, we obtain the nameservers within five minutes of the domain registration, and retrieve all IP addresses ever associated with the nameservers within one year before the domain registration. We use the mapping dataset from iPlane to map the IP addresses to Autonomous System numbers [29]. We focus on the IP addresses of the authoritative nameservers, available at registration time, while previous work mainly investigated IP addresses directly resolved for the domains [3, 5], usually configured later close to attack launch time (*e.g.*, using fast flux [24]).

Registration time (categorical). Miscreants need to repeatedly register domains for turnover in spam activities; this behavior may reveal certain temporal patterns. In `WHOIS` information, "Creation Date" indicates when a domain was registered. We have equivalent registration times from the zone update data, and extract hour-of-the-day and day-of-the-week. The hour of the day can reflect the time zones of registrants' locations, and the day of the week can capture the registrants' workflows. We extract the information according to Eastern Standard Time (*i.e.*, UTC -0500), although this choice is not significant because the purpose is to capture repeated temporal patterns of domain registrations. The hour of the day corresponds to 24 categorical features, and the day of the week maps to seven features.

Registration period (ordinal). A user can register a domain for one to ten years. "Expiration Date" in `WHOIS` shows when the domain

Category	*Feature*	*Type*	*New?*
Domain profile	Registrar	Categ.	[13, 20]
	Authoritative nameservers	Categ.	[13, 20]
	IP addresses of nameservers	Categ.	✓
	ASes of nameserver IP addresses	Categ.	✓
	Daily hour of registration	Categ.	✓
	Week day of registration	Categ.	✓
	Length of registration period	Ord.	✓
	Trigrams in domain name	Categ.	✓
	Ratio of the longest English word	Cont.	[20]
	Containing digits	Categ.	✓
	Containing "–"	Categ.	✓
	Name length	Ord.	✓
	Edit distances to known-bad domains	Cont.	✓
Registration history	Life cycle	Categ.	[20]
	Dormancy period for re-registration	Ord.	[20]
	Previous registrar	Categ.	✓
	Re-registration from same registrar	Categ.	✓
Batch correlation	Probability of batch size	Cont.	[20]
	Brand-new proportion	Cont.	✓
	Drop-catch proportion	Cont.	✓
	Retread proportion	Cont.	✓
	Name cohesiveness	Cont.	✓

Table 3: Summary of PREDATOR **features. We include references to previous work that has proposed similar features in the context of proactive registration-based detection. The "New?" column highlights features that this work explores for the first time; where a feature was previously studied or identified, we provide a reference.**

is about to expire. The user owns the domain and can renew it any time before its expiration. Longer registration terms mean higher up-front fees. We find that almost all spammer domains have one-year initial registration terms, presumably since spammers tend to abandon their domains due to blacklisting. We use the years between domain registration and its potential expiration as one feature.

Lexical patterns (containing multiple features). Registrants of benign domains tend to choose easy-to-remember names. On the other hand, to acquire large numbers of domains for attack campaigns, miscreants generate random-looking names or produce similar names by following certain rules. We compute the features to capture facets of the linguistic structure of domain names. Though some of these features are not strictly new, previous work has not investigated the lexical patterns of new domains across entire zones. When analyzing the naming patterns, we only use the name in the second-level domain and ignore the trailing TLD (like ".com").

1) *Trigram* (categorical). We use the standard trigram approach to represent a domain name and to examine the character sequence. A domain name can only consist of the characters of letters, digits, and hyphens [6, 40]. (Internationalized domain names use the same character set to encode Unicode characters [32].) Since DNS systems treat domain names in a case-insensitive manner, we convert the names into lower case to process. We have a group of $37^3 = 50,653$ binary features that represent all the possible trigrams on the allowed alphabet of 26 letters, 10 digits and the hyphen character. We set a given feature to 1 if and only if the corresponding trigram appears in the domain name; otherwise, we set it to 0.

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

3) *Containing digits* (categorical). We observe that spammer domains (18% for .com and 42% for .net) are more likely to use numerical characters than non-spammer ones (10% for .com and 12% for .net). Possible reasons might be that miscreants add digits to derive numerous names from the same word affix or generate random names from a character set containing digits. We include a binary feature to indicate whether the domain name contains any digits.

4) *Containing "–"* (categorical). Similarly, miscreants can insert "–" to break individual words or concatenate multiple words, though our data did not show large differences regarding this attribute between spammer and non-spammer domain names. We include a binary feature to indicate whether the domain name contains any hyphens.

5) *Name length* (ordinal). Miscreants may create domains based on a specific template, such as random strings of a given length. We use the length (number of characters) of the domain name as a feature.

6) *Edit distances to known-bad domains* (continuous). We examine the similarity of a domain to a set of known-bad domains. We derive a set of previously known spammer domains based on the prior knowledge, compute the Levenshtein edit distances to the currently considered domain, and divide these edit distances by the length of the domain name for normalization. In our experiment (Section 7), we use the data from a separate month to extract known-bad domains, which do not overlap with any training or testing data. We take the five smallest normalized edit distances as features, which have values between 0 and 1 (we have experimented with various numbers of edit distances, and the detection performance remains similar). Although calculating edit distances is computationally expensive, it remains tractable given the pace of domain registrations. In our experiment, the size of the known-bad .com domain set is 66,598. On average, the calculation of edit distances for one domain to the set of known-bad domains requires 0.13 seconds on a 2.40GHz CPU machine; calculating these values for one month of new .com domains takes four days.

5.2 Registration History Features

Registration history features are based on previous registration history of a domain. If a domain has appeared in the zone before, we possess registration history, such as previous registrar and deletion time. Most of such features can be obtained from third-party services such as DomainTools [10] and Who.is [63], or they are available at registrars and registries. Due to the limitations of our data, we only consider the features regarding the most recent previous registration.

Life cycle (categorical). As mentioned in Section 2, we categorize domains as brand-new, drop-catch, and retread, depending on the registration history. Although the life-cycle type itself may not be a strong indicator whether the domain is registered for spam-related activity, it often provides discriminative information when combining with other features, such as the life-cycle types of the other domains registered from the same registrar and around the same time. In total, the life-cycle categories map to three binary features and only one of them is set to 1.

Dormancy period for re-registration (ordinal). The usual re-registration domains tend to include some that expired a long time ago. On the other hand, spammers intentionally collect expired domains from publicly released sources [42, 59], which concentrate on recently expired domains. We observe that 30% of non-spammer retread domains were re-registered within 90 days, while 65% of

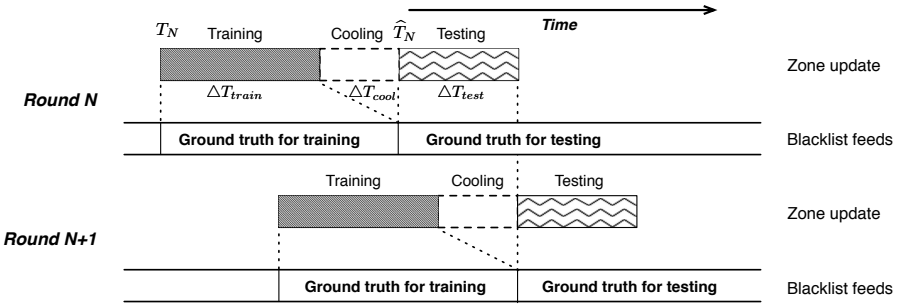

Figure 3: Sliding windows to train detection models.

spammer retread domains in .com and 37% of spammer retread domains in .net were re-registered within 90 days. We take the seconds between its last expiry and current registration as a feature. Regarding brand-new domains, the feature of dormancy period is not applicable, and we assign a default value (0 in our experiment).

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

Re-registration from the same registrar (categorical). We add features to explicitly indicate whether the registrar of a re-registration domain is the same registrar of the previous registration. We observe that in the .com zone[2] 18% of non-spammer retread domains use the same previous registrar, while only 4% of spammer retread domains do so, presumably because miscreants choose particular registrars (Section 5.1), which are unlikely to be those used by prior legitimate registrants. We use a dummy value to deal with brand-new domains.

5.3 Batch Correlation Features

Batch correlation features examine domains within the same tuple ⟨registrar, five-minute epoch⟩, which we define as a batch. The batch information is initially known by registrars or registries.

Probability of batch size (continuous). Miscreants often register domains in large batches, presumably due to cheaper price of bulk registration or management ease. We identify the qualitatively different registration behavior by using the model of compound Poisson process, defined as in Hao et al. [20]. A low-probability batch size from the model indicates an abnormally large registration spike. We use the probability as a feature in our system.

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

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

[2]Our data has no previous registrar information on .net, per Section 7.1.

6. CLASSIFIER DESIGN

This section introduces the Convex Polytope Machine (CPM), a supervised learning algorithm that we use (including our rationale for selecting this algorithm); the process of building the detection models; and the derivation of feature importance based on the models.

6.1 Supervised Learning: CPM

We want a classifier that can quickly train over large sets of data and achieve high accuracy. While linear Support Vector Machines (SVM) [12, 47] or comparable linear methods are often used in such high performance settings, nonlinearities in our data raise difficulties for SVM-style approaches. Instead, we employ a state-of-the-art supervised learning algorithm, the Convex Polytope Machine (CPM) [30]. CPM maintains an ensemble of linear sub-classifiers, and makes its final decision for incoming instances based on the maximum of all of their scores. More formally, suppose $\mathbf{x} \in \mathbb{R}^d$ is an instance of d features, and $\mathbf{w}^1, \ldots, \mathbf{w}^K \in \mathbb{R}^d$ represent the weights of the K sub-classifiers. We derive the score of $\mathbf{x}$ as:

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

The prediction score of $f(\mathbf{x})$ reflects how likely a domain is registered for spam-related activity. Geometrically, a CPM defines a convex polytope as the decision boundary to separate the two instance classes. In our application, it appears that this richer, non-linear decision boundary gives us high classification accuracy. Training of a CPM can be efficiently achieved by using the gradient descent technique [47]. To assess our design choice, we tested SVM [12] using `libsvm` with parameters tuned to our application. We found that in the low-false-positive region of operation, CPM produced a 10% higher true-positive rate, and trained faster than an SVM.

6.2 Building Detection Models

The first step of building the model is to normalize the continuous and ordinal features. We transform real values into the $[0, 1]$ interval to ensure that they do not overly dominate categorical features. We compute the ranges for each of the continuous and ordinal features to obtain max/min values, and normalize feature v to $(v - v_{min})/(v_{max} - v_{min})$. Since the categorical features are in binary, we do not need additional normalization process.

We adapt a sliding window mechanism for re-training models and evaluating the detection accuracy close to the real-deployment scenario. We define three windows: training $\triangle T_{train}$, cooling $\triangle T_{cool}$, and testing $\triangle T_{test}$. As shown in Figure 3, suppose at round N the training window starts at time T_N. The model will

be constructed at time $\widehat{T}_N = T_N + \triangle T_{train} + \triangle T_{cool}$, with the domains registered during $[T_N, T_N + \triangle T_{train}]$. Since this approach requires time to corroborate that a domain is indeed involved with spam-related activity, especially based on observations from blacklists, we use the ground truth collected during the period $[T_N, \widehat{T}_N]$ to label domains to build the model (in the training mode). In the testing period (corresponding to the operation mode), PREDATOR makes real-time prediction on those domains registered during $[\widehat{T}_N, \widehat{T}_N + \triangle T_{test}]$. The ground truth that we use in the testing period to evaluate the detection accuracy is composed of the domains showing on blacklists from time $\widehat{T}_N$ up to our last collection date of the blacklists. In the next round, $N + 1$, we move the time window forward by $\triangle T_{test}$, which makes the new model build at time $\widehat{T}_N + \triangle T_{test}$. The period $\triangle T_{test}$ indicates how frequently we re-train the model. Operators can customize the three window lengths according to different requirements and settings (see Section 7.4).

6.3 Assessing Feature Importance

Given a subset $S \subset \{1, \ldots, d\}$ of features, a derived CPM model $\{\mathbf{w}^1, \ldots, \mathbf{w}^K\}$, and a dataset of points $\{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$, we derive a measure to evaluate the importance of the set of features in our classifier. If the model weights have large magnitudes while at the same time the associated features have low variance, *i.e.*, they are essentially constant, these dimensions are not particularly informative and should receive a low importance score. We design a scoring method to measure the total amount of variation on the score $f(\mathbf{x})$ over the dataset induced by the features S. In the case of a single linear classifier ($K = 1$), we measure this quantity as:

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

To generalize this measure to the case $K \geq 2$, for each sub-classifier k, we compute the score I_S^k based on its subset of assigned instances A_k, and combine the scores.

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

where A_k is composed of $\mathbf{x}$, satisfying $k = \arg\max_{k'} \langle \mathbf{w}^{k'}, \mathbf{x} \rangle$. Higher values of I_S indicate that the feature group S contributes more on the decision-making. We demonstrate the feature importance in Section 7.4.

7. EVALUATION

In this section, we report our evaluation results, compare the performance of PREDATOR to existing blacklists, and analyze the evasion scenarios.

7.1 Data Set and Labeling

Our primary dataset consists of changes made to the .com zone, the largest TLD [60], for a five-month period, March–July 2012. We obtain the DNZA files from Verisign (which have five-minute granularity), find the registrations of new domains, and extract the updates of authoritative nameservers and IP addresses. During March–July 2012, we have 12,824,401 newly registered second-level .com domains. To label the registered domains as legitimate or malicious, we collected public blacklisting information from March–October 2012 (eight months), including Spamhaus [49] (updated every 30 minutes), URIBL [56] (updated every 60 minutes), and a spam trap that we operate (real time). If a domain appeared on blacklists after registration, we label the domain as being involved in spam-related activities and registered by miscreants. To obtain benign labels, we queried McAfee SiteAdvisor [48] in June 2013 to find the domains that are reported as definitely benign. Eventually we have about 2% of .com domains with malicious labels and 4% with benign domain labels. We discuss the prediction results on labeled and unlabeled domains respectively in Section 7.2.

We also obtain the DNZA data of .net zone for five months, from October 2014 to February 2015, which contain 1,284,664 new domains. We used similar blacklists (URIBL and our spam trap) from October 2014 to May 2015 (eight months) to label malicious domains and queried McAfee in November 2015 to find benign labels. However, the information for .net domains is not complete, which just allows limited analysis on .net domains. We only have Spamhaus snapshot on December 7th 2015 (instead of a continuous feed), and the previous registrar information is not available.

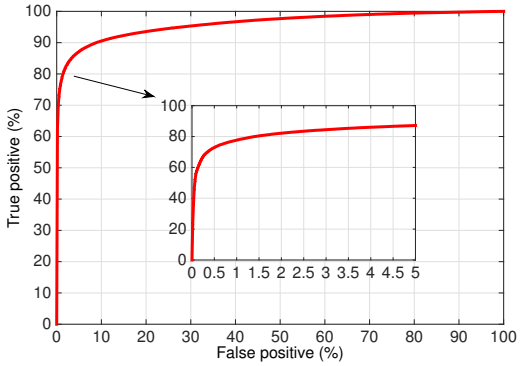

Figure 4: ROC of PREDATOR on .com domains. The inlay figure shows the ROC curve under the range of 0–5% false positives.

Page content of domain	*Percentage*
Advertisement links	38%
Lottery, survey, and coupon	7%
Adult content	7%
Merchandise	7%
Pharmaceutical	6%
Download of Software and files	5%
Online gambling	4%
No obvious spam-related content	26%

Table 4: Breakdown of manually checking 100 random samples of unlabeled domains that PREDATOR classifies as malicious. (Note that pages with no obvious spam-related content might still host other malicious activities such as drive-by downloads.)

7.2 Detection Accuracy

We demonstrate the accuracy of PREDATOR in terms of false positive rate, which is the ratio of benign domains misclassified as malicious to all benign instances; and detection rate, which accounts for the ratio of correctly predicted spammer domains to all spammer domain samples. By setting different thresholds, we make tradeoffs between false positive rates and detection rates.

For .com domains, we use data from March 2012 to extract the known-bad domain set and derive probability models for registration batches, and take April–July 2012 for our experiments. We used the sliding window method (introduced in Section 6.2) and tested different window lengths, where better results resulted from longer training windows (*i.e.*, more domains for training) and shorter testing windows

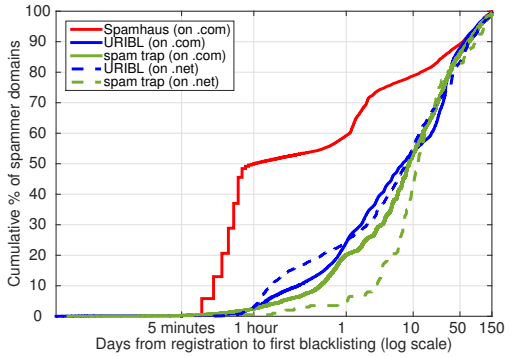

Figure 5: Distribution of days between domain registration and appearance in either our spam trap, URIBL, or Spamhaus, which indicates how early PREDATOR **can make detections compared to the existing blacklists (no Spamhaus timing information on .net).**

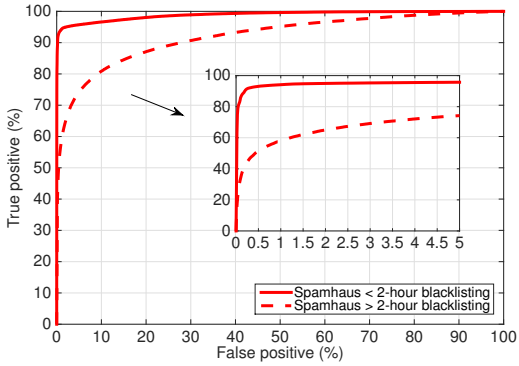

Figure 6: ROC of PREDATOR **using domains that Spamhaus blacklisted within the first 2 hours of registration and after the first 2 hours of registration for labels.**

Training window \ *Testing window*	7 days	35 days	56 days
35 days	**70.00%**	68.29%	66.81%
21 days	67.10%	64.96%	60.56%
14 days	64.13%	60.51%	58.22%

Table 5: Detection rates (under a 0.35% false positive rate) with different window settings. With shorter training windows and longer testing windows (*i.e.*, less frequent re-training), the prediction will become more inaccurate. Our experiments show that the performance is not overly sensitive to the window settings.

(*i.e.*, more frequent re-training). We demonstrate the performance results of PREDATOR with the setting of the training window to 35 days, the cooling window to 1 day, and the testing window to 7 days, which produces good detection accuracy and allows realistic operation (see Section 7.4 for detailed discussion on window selection).

Figure 4 shows the ROC curve of PREDATOR. The x-axis shows the false positive rate, and the y-axis shows the detection rate. The inlay figure shows the ROC curve for the range of 0–5% false positives. PREDATOR achieves good detection rates under low false positives. For example, with a 70% detection rate, the false positive is 0.35%. We emphasize that these results only rely on features constructed from the limited information available at registration time. Thus, as an early-warning mechanism, PREDATOR can effectively detect many domains registered for malicious activities.

Results on the entire .com zone. We project the 0.35% false positive to the entire .com zone. Since there are around 80,000 new domains everyday, the daily false positives are about 280 domains (as an upper estimate, assuming all domains are benign). Given that even the known spammer domains totalled more than 1,700 every day, PREDATOR can greatly help to narrow down the set of suspect domains. We ran additional tests to examine how many unlabeled domains are classified as spam-related by using the constructed detection model on the entire zone dataset, about seven million .com new domains registered over three months. With a threshold under a 0.35% false positive rate (in Figure 4, obtaining a 70% detection rate), PREDATOR reports about 1,000 unlabeled domains per day as spam-related, the same magnitude as the labeled spammer domains (1,700 per day). Overall, PREDATOR predicts 3% of all newly registered .com domains as malicious, which capture 70% of the malicious domains showing up later on blacklists. As a first line of defense, PREDATOR can effectively reduce and prioritize suspect domains for further inspection (*e.g.*, URL crawling or manual investigation) and find more malicious pages given a fixed amount of resources. In Section 7.3, we investigate to what extent the unlabeled domains that PREDATOR classifies as malicious indeed connect to illicit online activities while missed by current blacklists.

Detection accuracy on .net domains. We performed a similar experiment to report the detection accuracy on .net zone (five months in 2014–2015). Due to data limitation, two features, the previous registrar and re-registration from the same registrar, are unavailable, and in the blacklists Spamhaus only has a single snapshot. With the same sliding window setting, the detection rate on .net domains is 61% (close to the 70% on .com domains) under a 0.35% false positive rate. The result shows that PREDATOR can successfully make prediction at different zones. In the rest experiments, we focus on .com domains (.net domains either yield similar results or cannot conduct the analysis due to data defect).

7.3 Comparison to Existing Blacklists

We investigate and compare different blacklists and find that PREDATOR can help to mitigate the shortcomings of current blacklisting methods and detect malicious domains earlier.

Detection of more spammer domains. The first property we examine is *completeness*, which explores how many spammer domains PREDATOR detects compared to other blacklists. We find that during May–July 2012, the exclusive blacklisted .com domains (*i.e.*, not reported by other feeds) on Spamhaus, URIBL, and our spam trap number 24,015, 4,524, and 442 respectively. Each blacklist has many domains not identified by other sources, which indicates the existing blacklists are not perfect to detect all malicious domains. Having incomplete blacklists makes it quite challenging to develop more accurate registration-time detection, and also shows how PREDATOR can complement current detection methods by leveraging the central observation of domain registrations.

To investigate the potential of PREDATOR to detect spammer domains that existing blacklists miss, we sample the unlabeled .com domains that PREDATOR predicts as bad, and manually check whether they host any spam-related content. When we performed the analysis, most of the domains had expired, and we could not directly crawl and assess their content. Alternatively, we used historical snapshots from DomainTools [10] and the Internet Archive [28]. Table 4 shows the page categories of 100 randomly sampled domains. 74% of the unlabelled domains that PREDATOR predict as bad refer to content

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

Table 6: Ranking of feature importance in PREDATOR **(D for domain profile category, R for registration history category, and B for batch correlation category).**

that is often hosted on spam-related sites (*e.g.*, pharmaceutical content, adult content), and 26% of these pages have no obvious spam-related content (though might have other malicious activities that we cannot measure, such as drive-by downloads). This result demonstrates PREDATOR's ability to augment existing blacklists by exposing malicious domains that they fail to report.

Early detection. Another important blacklisting characteristic concerns *delay*: how long after a spammer domain registration the blacklists identify it. Detection delays leave users unprotected in the interim, allowing attackers to reap greater benefits from their domains. Figure 5 shows the distribution of the time between a .com/.net domain's registration and its first appearance on blacklist (no Spamhaus timing information on .net). We observe that both URIBL and our spam trap take significant time to identify spammer domains, *e.g.*, around 50% of blacklisted domains manifest after 7 days. Clearly, PREDATOR can make detection early, even weeks before appearance on blacklists, which provides more time to respond or prevent attacks. On the other hand, Spamhaus has a mode of time-of-registration blacklisting, where a certain amount of blacklisting occurs shortly after domain registrations. We use a two-hour threshold to estimate conservatively, since the Spamhaus feed that we use updates every half hour.

To assess the degree to which Spamhaus uses time-of-registration features to blacklist domains, and to explore how the features that Spamhaus uses compare to our features, we evaluate the accuracy of PREDATOR using (for both training and testing) only the domains that Spamhaus blacklists in the first two hours of registration to label malicious domains. We then repeat the analysis for domains that Spamhaus blacklists more than two hours of registration. Figure 6 shows the prediction accuracy of PREDATOR using these two sets of labels. PREDATOR achieves a detection rate above 93% with a 0.35% false positive when using as labels the domains that were blacklisted within two hours. The high accuracy result suggests PREDATOR features already contain most of those used by Spamhaus (anecdotes indicate that Spamhaus involves only simple features, and in Section 7.4 we further infer what features Spamhaus relies on). PREDATOR also achieves decent accuracy using the domains that

Rank	*Category*	*Feature*	*Score ratio*
1	**D**	Authoritative nameservers	100.0%
2	**D**	Registrar	47.72%
3	**D**	IP addresses of nameservers	44.26%
4	**D**	Trigrams in domain name	37.91%
5	**D**	ASes of nameserver IP addresses	24.98%
6	**D**	Daily hour of registration	14.23%
7	**R**	Re-registration from same registrar	19.50%
8	**B**	Retread proportion	11.48%
9	**R**	Life cycle	10.93%
10	**B**	Drop-catch proportion	10.70%

Table 7: Top 10 ranked features in PREDATOR **when applying on Spamhaus $<$ 2-hour blacklisting (same categories as in Table 6).**

Spamhaus blacklists more than two hours after registration as labels: a 0.35% false positive for a detection rate of about 47%. This result shows PREDATOR can detect some malicious domains much faster than Spamhaus can.

7.4 Analysis of the Classifier

Contribution of new features. As shown in Section 5 (and Table 3), 16 out of 22 features that we identified and incorporated into PREDATOR are proposed for the first time. To evaluate the contribution of these new features to improving the accuracy, we run an experiment with solely the features that previous work has explored, and the detection rate drops to 58.40% (under a 0.35% false positive rate). On the other hand, PREDATOR with the full feature set achieves the 70% detection rate under the same false positive rate. The result shows that the new features that we introduced can considerably improve the detection accuracy. Our work is the first to develop a reputation system that is able to accurately and automatically predict the maliciousness of a domain at registration time. Note that prior research either just presented preliminary measurement results [20], or was limited to extrapolating from particular properties, such as self-resolving nameservers [13].

Sliding window settings. We have used the sliding window mechanism (introduced in Section 6.2) in the experiments to simulate the practical deployment scenario of PREDATOR. The length of the training window determines how much data to build the classifier, and the length of the testing window indicates how often we re-train the model. In Table 5, we compare the detection rates with different training/testing windows under a 0.35% false positive rate (for our data, different cooling windows yield similar performance and we set it to one day). Shorter training windows (*i.e.*, less training data) and longer testing windows (*i.e.*, less frequent re-training) will produce less accurate predictions. The results show that the accuracy of PREDATOR is not overly sensitive to the window settings. For our data, training on 35-day data and re-training weekly is sufficient to maintain accuracy. We expect that when PREDATOR runs on a different dataset, additional analysis should be performed and the window settings may vary across different datasets.

Feature ranking. We use the scoring method in Section 6.3 to rank our features. The scores represent how much the features can contribute to identify either malicious or benign labels. For easy interpretation, we calculate the score ratio by dividing the score values with the largest one. Table 6 ranks all registration-based features on .com zone (with the most important feature at top). The capitalized letters in the second column indicates the feature categories: **D** for domain profile, **R** for registration history, and **B** for batch correlation. Seven of the top ten features belong to the domain profile category. This result is quite encouraging, since most of these features can be

collected with less overhead and from public sources, such as WHOIS database.

The ranking of features can help us to infer what features Spamhaus appears to rely on for its time-of-registration blacklisting. Table 7 lists the feature importance when we apply our detection algorithm on the domains blacklisted by Spamhaus within two hours of registration. We focus on the top ten features. The difference between the ratio for the first two features in Table 7 appears larger than the ratio difference in Table 6, which indicates that Spamhaus was inclined to use the feature of authoritative nameservers for detection. In fact, when considering only the nameservers that have more than 90% of their hosted domains appearing on Spamhaus time-of-registration blacklisting, those nameservers account for 86% of all domains appearing on Spamhaus time-of-registration blacklisting. The observation suggests that Spamhaus heavily uses nameservers to make time-of-registration blacklisting decisions.

7.5 Evasion

As with any detection system, sophisticated attackers may attempt to evade PREDATOR. We argue that trying to evade PREDATOR will alter the economics for miscreants to acquire domains and consequently impair their attack capability.

We first consider two groups of PREDATOR features, nameserver-related and lexical, which have relatively high ranks in Table 6. Nameserver-related features include nameservers of the second-level domains and the corresponding IP addresses and ASes (rank 1, 3, 5 in Table 6). These features are inherent to the hosting infrastructure, which require effort for attackers to alter. We evaluate PREDATOR's performance under different evasion scenarios by excluding the corresponding features from the system, as shown in Figure 7. The red solid curve corresponds to the ROC curve of PREDATOR incorporating all features on .com zone, and the blue dashed curve indicates the ROC curve if miscreants evade nameserver-related patterns (three features). Though PREDATOR's performance degrades, it still achieves a good level of detection accuracy. This observation suggests that nameserver-related features are important, but in their absence other features can still contribute to retain good detection.

We next consider lexical features (rank 2, 7, 12, 17–19, 21 in Table 6). Generating a large number of names is not a trivial task, as the plausibility of the names could influence an attack's efficacy. Changing naming patterns to use irrelevant words or random strings could reduce click-through rates for spam or phishing. Attackers may attempt to exploit HTML emails or pages to manipulate the displayed domains/URLs. However, the mismatch between the hyperlink text and the underlying domains would make it easier to be detected by previous work [15]. If miscreants try to evade name-similarity by inserting numerical or hyphen characters, PREDATOR's features of names containing digits and "–" can capture this. The green dashed curve in Figure 7 indicates the ROC with nameserver-related and lexical features excluded (six more features). We observe that detection accuracy further weakens, suggesting that lexical features help reduce false positives.

The batch features (rank 14, 15, 16, 20 in Table 6), despite their relatively low ranks, can contribute to the detection accuracy. In Figure 7, the purple dashed curve shows the ROC curve with batch features further excluded (four more features), where the performance decreases, especially in the low-false-positive area. As we will discuss later, if miscreants attempt to evade PREDATOR by mimicking legitimate behavior (including changing the patterns in registration batches), this would impair their attack capability, from financial and volume perspectives.

Registrars represent an essential feature in our system (rank 4 in Table 6) to capture miscreants' tactics. Miscreants tend to use the registrars that are cheaper and more tolerant of their activities [37]. Evading this feature forces attackers to change to less "scam-friendly" registrars. Even if miscreants switch to different sets of registrars, PREDATOR can over time automatically learn the shifts and detect new malicious domains. The black dotted curve in Figure 7 illustrates the performance of using only the registrar feature. The relatively large decrease in accuracy suggests that single features have limited detection power, and combinations of other features can significantly improve effectiveness.

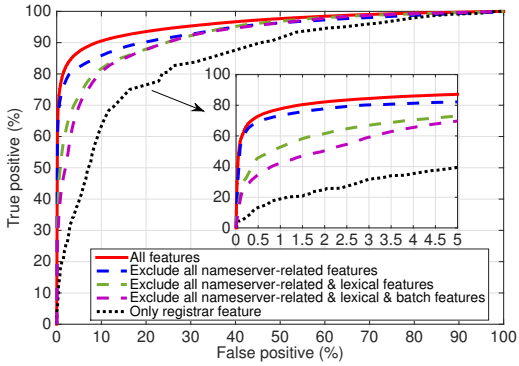

Figure 7: ROC of PREDATOR under different features (simulated evasion scenarios).

Financial cost of evasion. Evading some PREDATOR features forces miscreants to spend more to acquire domains. As discussed, to evade registrar feature (rank 4), attackers have to use some registrars with higher prices that are not their first choice. Evading the bulk-registration feature (rank 20) forces attackers to spend more by foregoing bulk discounts. Some miscreants also pay with stolen credit cards (to reduce cost and avoid tracing their real identities) [11, 14], which requires bulk registration, since fraud detection disables cards after several purchases. Evading the registration-period-length feature (rank 22) likewise requires greater expense due to paying in advance for multi-year terms. Verisign charges a $7.85 annual fee to registrars for each .com domain registration [58]. Miscreants can get low prices close to this amount from registrars affiliated with scam activities (*e.g.*, ABSystems [50]) or registrars offering cheap prices/discounts. If miscreants switch to the largest registrar GoDaddy, with an annual price of $12.99 [17]; register in small batches without taking any bulk discount; and commit to a 2-year registration term then the price per domain rises $\frac{\$12.99}{\$7.85} \times 2 \approx 3.3$ times of the price that miscreants originally pay.

Evading by decreasing volume. Evading some of the PREDATOR features constrains the volume of names that miscreants can easily register. In an attempt to evade the life-cycle proportion features (rank 14–16), miscreants may mix different life-cycle types of domains in the same registration batch. However, this will require incorporating multiple methods to generate domain names, increasing management effort. Moreover, we observe that spammer domains possess a lower brand-new proportion (66%) than non-spammer domains (77%). Given the same quantity of generated brand-new names, miscreants need to reduce re-registration domains to mimic the life-cycle proportions of general domains. We have $\frac{66\%}{100\%-A\%} = 77\%$. Solving for A, this yields that miscreants must alter 14% of their domain registrations to simulate benign behaviors. Note that this estimate demonstrates the impact to evade a single feature. To change other behaviors, such as the aforementioned lexical or registrar features, miscreants may have to further cut their domain registrations to small volumes. Another feature is the dormancy period for re-registration domains (rank 9). Domains re-registered by miscreants are often

those that expired more recently, presumably because miscreants actively mine expired domains. Evading the dormancy period feature forces miscreants to wait longer to register expired domains, which in turn limits the domains miscreants can use over a given period.

In general terms, evasion attempts considerably increase economic and management costs for attackers; PREDATOR raises the bar for miscreants to acquire and profitably use the domains. The combination of the features is effective to detect malicious domains. Altering one or two features will not significantly aid miscreants to fly under the radar.

8. DISCUSSION

In this section, we discuss possible deployment scenarios, as well as some limitations of our work.

Deployment scenarios. Network operators and security practitioners can benefit from PREDATOR in the following ways. (1) Network operators can take appropriate actions to protect their networks and users. For example, email servers can greylist [34] (*i.e.*, temporarily reject) emails that PREDATOR predicts as suspicious and request the originating servers to try again after a period. Legitimate senders are expected to resend the emails, while spammers usually do not properly handle retries [36]. Meanwhile, network operators can collect more evidence before retry attempts to make final decisions, such as examining the Web content on the domain. (2) Registries or registrars can require more strict documentation or verification (*e.g.*, validation of payment instruments or inquiring the domain purpose), before they approve registrations of domains with low reputation scores. Mitigating domain abuse is consistent with the registrars' responsibilities under the ICANN's Registrar Accreditation Agreement [26] (some registrars have taken important roles [8, 21]), and it can also help to identify and deter the illegal registrations with stolen credit cards [11, 14] (which cause loss from registrars, including refunds and chargeback fees). (3) Law enforcement and cyber-security professionals can prioritize their investigations (for time/resource-consuming analysis, *e.g.*, crawling the page content or repeated manual investigation by an analyst) and proactively monitor low-reputation domains, since the domains selected by PREDATOR are more likely to prove malicious. (4) Operators could also incorporate PREDATOR into other detection systems (*e.g.*, spam filters, botnet detection systems) by using it to provide an additional "confidence score" of registration to help them determine whether a particular domain appears malicious.

Limitations and future work. Given that domain registrations under a single zone provide centralized observation opportunities, miscreants may register domains across different TLDs, especially with the expansion of large numbers of new TLDs [27]. Thus, one direction for future work is incorporating cross-zone features into the classification model. Although PREDATOR is somewhat resistant to evasion, designing a more robust system against the attackers' continual attempts to mislead or evade the classifier is a promising area for future work. While our approach achieves good accuracy, for higher detection rates the false positive rates increase as well. Another area for improvement is combining post-registration detection techniques, such as DNS monitoring or Web crawling, to develop hierarchical decision-making mechanisms for higher accuracy.

9. RELATED WORK

We compare previous work on analyzing and detecting domains and URLs that are used in illicit online activities.

DNS-based detection. Most previous DNS-based detection studies focused on analyzing lookup traffic. Notos [3] and EXPOSURE [5] leverage traffic from local recursive DNS servers to establish domain reputations. Gao et al. used temporal correlation in DNS queries to detect malicious domain groups [16]. Various previous work has analyzed DNS traffic to detect fast-flux domains [24, 44], or malware domains exploiting resource records (*e.g.*, TXT records) as the communication channels [9, 31, 66]. Other work inspects DNS traffic close to top-level domain servers to detect abnormal activity [4, 19]. In contrast, PREDATOR derives domain reputation using registration features to enable early detection, without monitoring DNS traffic.

Registration and domain market. Recent research has paid attention to the registrars, registries, and the domain market, including domain-name speculation, typosquatting, and domain parking [1, 2, 7, 54, 61]. Liu et al. found that registry policy changes and registrar-level takedown had at least temporary effects in deterring spam-advertised domains [37]. Felegyhazi et al. investigated registration information to extrapolate malicious domains from specific instances of known-bad domains, primarily relying on the properties of DNS servers [13]. Hao et al. measured and modeled domain registrations of spammer domains [20]. While their work only presents preliminary measurement results and does not examine how to leverage the findings for detection, it hints at the potential of building a registration-based reputation system. We designed PREDATOR just for that purpose, to accurately detect spammer domains at time-of-registration, and our work studied significantly more features.

Website and URL detection. A conventional technique to detect malicious Web pages is through automatic URL crawling tools. The detection can be based on the page content [43, 57], the presence of cloaking and redirection [35, 62], or the link structure leading to the pages [64]. Thomas et al. built a large-scale system to crawl URLs in email and Twitter feeds to detect malicious messages [55]. Our study is orthogonal to Web crawling methods, and does not require page visits. The output of PREDATOR can help with prioritizing what suspect sites to crawl and inspect. A related approach is to use various lexical and host-based features of the URL for detection, but exclude Web page content [38, 39, 65]. These detection methods require waiting until miscreants use the URLs for attacks. Our work, on the other hand, provides proactive detection of domains before malicious URLs propagate on the Internet.

10. CONCLUSION

Because determining the reputation of DNS domains can significantly aid in defending against many Internet attacks, establishing DNS domain name reputation as quickly as possible provides major benefits. Whereas existing DNS reputation systems establish domain reputation based on features evident after the domain is in use, PREDATOR can accurately establish domain reputation at the time of domain registration, before domains ever see use in attacks. Our results show that PREDATOR can provide more accurate and earlier detection compared to existing blacklists, and significantly reduce the number of suspicious domains requiring more resource-intensive or time-consuming inspection.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. We also thank Christopher Kruegel, Kevin Borgolte, and Jennifer Rexford for many helpful suggestions and discussions to improve the paper. This work was supported in part by the National Science Foundation awards CNS-1237265, CNS-1535796, CNS-1540066, and by a gift from Google. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis. Seven Months' Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse. In *Network and Distributed System Security Symposium (NDSS)*, Feb. 2015.

[2] S. Alrwais, K. Yuan, E. Alowaisheq, Z. Li, and X. Wang. Understanding the Dark Side of Domain Parking. In *23rd USENIX Security Symposium*, Aug. 2014.

[3] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a Dynamic Reputation System for DNS. In *19th USENIX Security Symposium*, Aug. 2010.

[4] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon. Detecting Malware Domains at the Upper DNS Hierarchy. In *20th USENIX Security Symposium*, Aug. 2011.

[5] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *Network and Distributed System Security Symposium (NDSS)*, Feb. 2011.

[6] R. Braden. *Requirements for Internet Hosts – Application and Support*. Internet Engineering Task Force, Oct. 1989. RFC 1123.

[7] S. E. Coull, A. M. White, T.-F. Yen, F. Monrose, and M. K. Reiter. Understanding Domain Registration Abuses. In *25th International Information Security Conference*, Sept. 2010.

[8] Demand Media. eNom and LegitScript LLC Announce Agreement to Identify Customers Operating Illegal Online Pharmacies. http://www.businesswire.com/news/home/20100921005657/en/, 2010.

[9] C. J. Dietrich, C. Rossow, F. C. Freiling, H. Bos, M. van Steen, and N. Pohlmann. On Botnets that use DNS for Command and Control. In *European Conference on Computer Network Defense*, Sept. 2011.

[10] DomainTools. http://www.domaintools.com, 2015.

[11] S. Ellis. Business Email Compromise Scams on the Rise. http://www.markmonitor.com/mmblog/business-email-compromise-scams/, 2015. MarkMonitor Blog.

[12] R. Fan, K. Chang, C. Hsieh, X. Wang, and Lin. LIBLINEAR : A Library for Large Linear Classification. *The Journal of Machine Learning Research*, 9(2008):1871–1874, 2008.

[13] M. Felegyhazi, C. Kreibich, and V. Paxson. On the Potential of Proactive Domain Blacklisting. In *3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, Apr. 2010.

[14] P. Festa. Identity Thieves Strike eBay. http://www.cnet.com/news/identity-thieves-strike-ebay/, 2012. CNET.

[15] I. Fette, N. Sadeh, and A. Tomasic. Learning to Detect Phishing Emails. In *16th International Conference on World Wide Web (WWW)*, May 2007.

[16] H. Gao, V. Yegneswaran, Y. Chen, P. Porras, S. Ghosh, J. Jiang, and H. Duan. An Empirical Reexamination of Global DNS Behavior. In *ACM SIGCOMM*, Aug. 2013.

[17] Godaddy Bulk Registration Prices. http://www.godaddy.com/domains/searchbulk.aspx, 2014.

[18] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The Underground on 140 Characters or Less. In *17th ACM Conference on Computer and Communications Security (CCS)*, Oct. 2010.

[19] S. Hao, N. Feamster, and R. Pandrangi. Monitoring the Initial DNS Behavior of Malicious Domains. In *ACM Internet Measurement Conference (IMC)*, Nov. 2011.

[20] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck. Understanding the Domain Registration Behavior of Spammers. In *ACM Internet Measurement Conference (IMC)*, Oct. 2013.

[21] K. J. Higgins. Google, GoDaddy Help Form Group To Fight Fake Online Pharmacies. http://www.darkreading.com/d/d-id/1134946, 2010. Dark Reading.

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

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

[24] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling. Measureing and Detecting Fast-Flux Service Networks. In *Network and Distributed System Security Symposium (NDSS)*, Feb. 2008.

[25] IANA. Root Zone Database. http://www.iana.org/domains/root/db, 2016.

[26] ICANN. Registrar Accreditation Agreement (Section 3.18). https://www.icann.org/resources/pages/approved-with-specs-2013-09-17-en, 2013.

[27] ICANN. Delegated Strings of New TLDs. http://newgtlds.icann.org/en/program-status/delegated-strings, 2015.

[28] Internet Archive. http://archive.org, 2015.

[29] iPlane. http://iplane.cs.washington.edu/data/data.html, 2015.

[30] A. Kantchelian, M. C. Tschantz, P. L. B. Ling Huang, A. D. Joseph, and J. D. Tygar. Large-Margin Convex Polytope Machine. In *Neural Information Processing Systems (NIPS)*, Dec. 2014.

[31] A. M. Kara, H. Binsalleeh, M. Mannan, A. Youssef, and M. Debbabi. Detection of Malicious Payload Distribution Channels in DNS. In *Communication and Information Systems Security Symposium*, June 2014.

[32] J. Klensin. *Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework*. Internet Engineering Task Force, Aug. 2010. RFC 5890.

[33] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamcraft: An Inside Look At Spam Campaign Orchestration. In *2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, Apr. 2009.

[34] M. Kucherawy and D. Crocker. *Email Greylisting: An Applicability Statement for SMTP*. Internet Engineering Task Force, June 2012. RFC 6647.

[35] N. Leontiadis, T. Moore, and N. Christin. Measuring and Analyzing Search-Redirection Attacks in the Illicit Online Prescription Drug Trade. In *20th USENIX Security Symposium*, Aug. 2011.

[36] P. Lieven, B. Scheuermann, M. Stini, and M. Mauve. Filtering Spam Email Based on Retry Patterns. In *IEEE International Conference on Communications (ICC)*, June 2007.

[37] H. Liu, K. Levchenko, M. Felegyhazi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage. On the Effects of Registrar-level Intervention. In *4th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, Mar. 2011.

[38] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, June 2009.

[39] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying Suspicious URLs: An Application of Large-Scale Online Learning. In *26th International Conference on Machine Learning (ICML)*, June 2009.

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

[41] Moniker Bulk Registration Discounts. http://www.moniker.com/service/discounts, 2014.

[42] NameJet Domain Name Aftermarket. http://www.namejet.com/pages/downloads.aspx, 2015.

[43] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting Spam Web Pages through Content Analysis. In *15th International Conference on World Wide Web (WWW)*, May 2006.

[44] R. Perdisci, I. Corona, D. Dagon, and W. Lee. Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces. In *25th Annual Computer Security Applications Conference (ACSAC)*, Dec. 2009.

[45] Annual Price $8.49 for .com at INTERNET.bs. http://internetbs.net/, 2015.

[46] Annual Price $24.95 for .com at DomainPeople. http://www.domainpeople.com/domain-names/pricing.html, 2015.

[47] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *24th International Conference on Machine Learning (ICML)*, June 2007.

[48] McAfee SiteAdvisor. https://www.siteadvisor.com/.

[49] Spamhaus. http://www.spamhaus.org/.

[50] Spamhaus. ABSystems Domain Registrar De-accredited. http://www.spamhaus.org/rokso/evidence/ROK10342/, 2013.

[51] Spamhaus. Can Registrars Suspend Domains for Spam and Abuse? https://www.spamhaus.org/faq/section/Generic%20Questions#127, 2015.

[52] B. Stone-Gross, R. Abman, R. Kemmerer, C. Kruegel, D. Steigerwald, and G. Vigna. The Underground Economy of Fake Antivirus Software. In *10th Workshop on Economics of Information Security (WEIS)*, June 2011.

[53] Symantec. Rise in URL Spam. http://www.symantec.com/connect/blogs/rise-url-spam, 2013.

[54] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Felegyhazi, and C. Kanich. The Long "Taile" of Typosquatting Domain Names. In *23rd USENIX Security Symposium*, Aug. 2014.

[55] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and Evaluation of a Real-Time URL Spam Filtering Service. In *32rd IEEE Symposium on Security and Privacy*, May 2011.

[56] URIBL. http://www.uribl.com/.

[57] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne. Tracking Web Spam with HTML Style Similarities. *ACM Transactions on the Web*, 2(1):3:1–3:28, 2008.

[58] Verisign. Verisign Announces Increase in .com/.net Domain Name Fees. https://investor.verisign.com/releasedetail.cfm?releaseid=591560, 2011.

[59] Verisign Domain Countdown. http://domaincountdown.verisignlabs.com, 2011.

[60] Verisign. The Domain Name Industry Brief. http://www.verisign.com/assets/domain-name-brief-dec2012.pdf, 2012.

[61] T. Vissers, W. Joosen, and N. Nikiforakis. Parking Sensors: Analyzing and Detecting Parked Domains. In *Network and Distributed System Security Symposium (NDSS)*, Feb. 2015.

[62] D. Y. Wang, S. Savage, and G. M. Voelker. Cloak and Dagger: Dynamics of Web Search Cloaking. In *18th ACM Conference on Computer and Communications Security (CCS)*, Oct. 2011.

[63] Who.is. http://who.is/domain-history, 2015.

[64] B. Wu and B. D. Davison. Identifying Link Farm Spam Pages. In *14th International Conference on World Wide Web (WWW)*, May 2005.

[65] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipko. Spamming Botnets: Signatures and Characteristics. In *ACM SIGCOMM*, Aug. 2008.

[66] K. Xu, P. Butler, and D. Yao. DNS for Massive-Scale Command and Control. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 10(3):143–153, 2013.