July, 2014

Metadata-driven Threat Classification of Network Endpoints Appearing in Malware

Andrew G. West
Aziz Mohaisen

Metadata-driven Threat Classification of Network Endpoints Appearing in Malware

Andrew G. West and Aziz Mohaisen

Verisign Labs – Reston, Virginia, USA
{awest, amohaisen}@verisign.com

Abstract. Networked machines serving as binary distribution points, C&C channels, or drop sites are a ubiquitous aspect of malware infrastructure. By sandboxing malcode one can extract the network endpoints (*i.e.,* domains and URL paths) contacted during execution. Some endpoints are benign, *e.g.,* connectivity tests. Exclusively malicious destinations, however, can serve as signatures enabling network alarms. Often these behavioral distinctions are drawn by expert analysts, resulting in considerable cost and labeling latency.

Leveraging 28,000 expert-labeled endpoints derived from ≈100k malware binaries this paper characterizes those domains/URLs towards prioritizing manual efforts and automatic signature generation. Our analysis focuses on endpoints' static metadata properties and not network payloads or routing dynamics. Performance validates this straightforward approach, achieving 99.4% accuracy at binary threat classification and 93% accuracy on the more granular task of severity prediction. This performance is driven by features capturing a domain's behavioral history and registration properties. More qualitatively we discover the prominent role that dynamic DNS providers and "shared-use" public services play as perpetrators seek agile and cost-effective hosting infrastructure.

1 Introduction

Malware, whether in the form of adware, banking trojans, or corporate espionage, is an issue that needs little introduction. With malware now resulting in over $100 billion in damages per year in the U.S. alone [10] there is an obvious incentive to mitigate its ill effects. Signature-based detection of existing malware installations has proven a popular and effective paradigm. By monitoring the network, filesystem, and/or registry interfaces one can trigger alerts when behaviors match *threat indicators* (TIs) or *indicators of compromise* (IOCs) published by anti-malware vendors. These indicators are produced by profiling known malware. For example, hashcodes of malware binaries are basic indicators which are now skirted through the frequent repacking and obfuscation of malcode.

In this work we concentrate our efforts on network activity and in particular the *endpoints* (*i.e.,* domains/URLs) of connections initiated by malcode. This is based on the observations that: (1) Outbound network connections are ubiquitous in malware as exploits obtain more complete program code, C&C

instructions, or transfer stolen data at drop sites. (2) Network endpoints are persistent identifiers; we identify several malicious domains that appear in 1000+ unique malware binaries. (3) Identifying an endpoint as malicious should force the malactor to migrate that destination, presumably with cost implications that disrupt attack economics. (4) Once threat endpoints have been identified, monitoring for infections can be centrally administrated at router/switch/firewall granularity in a lightweight fashion.

Given a set of known malware binaries their execution can be sandboxed to produce endpoints (see Section 3). Using Verisign's proprietary malware collection, roughly 93k samples produced 203k unique endpoints. Verisign's analysts have labeled ≈28k of these using their domain expertise to: (1) Classify endpoints as threats/non-threats. (2) Assign threats a low/medium/high severity. (3) Determine the granularity best encapsulating the threat (*i.e.,* the exact URL path or broadening that to a domain/subdomain). The process analysts use to arrive at these determinations is described further in Section 3.2.

Ascertaining the client-side performance of TIs/IOCs is difficult. Multiple anti-malware vendors publish such indicator feeds, hinting at their commercial viability. Regardless, it is clear the application of machine-assisted classification can improve the generation and coverage of such feeds. A scoring model for endpoints could lower latency by intelligently routing analysts to the most acute cases or eliminating their intervention altogether.

The 28k labeled endpoints act as a corpus for mining patterns that distinguish malware infrastructure from benign artifacts. Our feature categories include:

- URL STRUCTURE: TLD, subdomain depth, *etc.*
- WHOIS DATA: domain age, registrar, *etc.*
- BAYESIAN n-GRAMS: character patterns in names
- REPUTATIONS: historical behavioral evidence

Our measurements reveal a need for malicious entities to be cost-effective and agile. Dynamic DNS is extremely prevalent among threats, as are cheap TLDs, certain registrars, and Sybil attacks via public "shared-use" services. Reputation features in particular drive model performance, as parent domains tend to show consistent behavior at the subdomain level. The result is a scalable classifier that predicts binary threat status with 99.4% accuracy and severity at 93% accuracy. Performance is currently being evaluated in a production system, as is the feasibility of using the model/reputations to *proactively* grey-list endpoints.

Existing literature has explored URL structure, domain reputation, and registration patterns in in multiple security contexts including email spam [14, 26, 37], collaborative abuses [34], and phishing [8, 27]. As we detail in the next section, endpoints discovered in the context of malware execution are fundamentally different in structure and purpose than those in related fields. Relative to more complex sandbox analysis we show that a simplistic set of features is sufficient for strong performance without requiring a specialized perspective. Moreover, our use of expert human taggers enables confident supervised learning and the more nuanced ability to predict malware severity.

2 Related Work

To the best of our knowledge there is no single work that has analyzed and classified network endpoints contacted during malware execution. However, our pursuits are closely related to several research veins: feature development over URLs/domains in various security contexts, dynamic analysis of malware's network behavior, *etc.*. Here we elaborate on that related literature.

Endpoint Analysis in Security: The notion of using URL structure to predict malice is well established. Fields such as email spam [14, 26, 37], collaborative abuse [34], and phishing [8, 27] commonly leverage surface properties of a URL. While our proposal implements many of those features in this work it reveals the respective sets of URLs to be very different. For example, [26] shows token patterns are critical to learning spam/phishing URLs. Our proposal uses Bayesian language learning in a similar fashion and finds it be one of the most ineffective features (Section 4.3; Table 4). Consider that spam/phishing URLs often need to incentivize human click-throughs while the endpoints of our malware corpus tend to be buried deep in code/infrastructure.

Spam email defense in particular has sought to analyze the content residing at endpoints. The structural and language patterns of HTML pages have been generically mined [29, 33] and parsed for signs of commercial intention [12]. Our approach opts not to consider endpoint content. Although we have made preliminary progress in analyzing content acquisition towards the detection of drive-by-downloads [23], textual content and drive-by exploits form only a small portion of those URLs contacted by malware (Figure 4).

One set of spam-inspired features we successfully apply to malware endpoint classification are those speaking to domain registration behaviors [19, 26].

Dynamic Analysis of Malware: Dynamic malware analysis and sandboxed execution of (potential) malware is also an established approach as surveyed in [11, 13]. Bailey *et al.* [4] and the more scalable [5] have focused on behavior-based event counts (*e.g.,* processes created). Feature development has since advanced such that malware families can now be reliably identified [22, 36] and dynamic analysis can be deployed on end hosts [21].

Network Signatures of Malware: At the intersection of sandboxed execution and network signature generation lies [30, 31]. In that work, Internet-scale crawling is the first step in a scalable hierarchy of drive-by-download detection. Similar to our proposal, that system's output is effectively a blacklist of network endpoints; the Google Safe Browsing project. Though able to proactively identify threats on the public web, [30, 31] will not identify non-indexed exploits nor endpoints that are passively involved in malware infrastructure. By operating reactively over known malware binaries our approach has this broader scope.

Rather than sandboxing, [37] mines enterprise-scale network logs towards discovering malware presence and "suspicious" activity (corporate policy violations). That approach uses massive aggregation over deep network properties such as user agent strings, domain contact patterns, and traffic bursts. The

cluster-based approach is promising even in the absence of malware ground-truth, although it takes on the order of hours to process a single day's log. Other works rely on more specialized network perspectives. Bilge *et al.* proposed Exposure [7], a system to detect malware domains based on DNS query patterns on a local recursive server. Antonakakis *et al.* [2] functions similarly but analyzes global DNS resolution patterns and subsequently creates a reputation system for DNS atop this logic [1]. Others have focused on using network flows as the basis for discovering botnet command-and-control (C&C) traffic. This includes Bilge *et al.* [6] and a series of related systems from Gu *et al.* [15–17]. While those systems detect infections in an online fashion our work concentrates on the offline production of signatures (for online application) given known malware binaries. Our approach and its lightweight deployment footprint could sparsely deploy these more complex traffic monitoring techniques to find the malware binaries needed for analysis.

The aforementioned works all provide perspective on the malware ecosystem. Adding to this is the work of Stringhini*et al.* [32] which crowd-sources the discovery of suspicious redirections. Similarly, Levchenko *et al.* [25] studied malware ecosystems by analyzing click fraud and spam value chains. Our feature development and evaluation contributes to understanding this landscape.

Expert Produced Labels: Many academic works attempting malware analysis do so using corpora with machine-assisted labeling. Recent work shows such labels to be alarmingly inconsistent and poor in coverage [28]. This work is fortunate to use expert annotators which also reliably label the severity of threats.

3 Data Collection

Focus now shifts to the data used in analysis and model building. We describe how malware samples are obtained and sandboxed to produce network traces from which potential indicators are extracted (Section 3.1). These endpoints are given to analysts who determine threat legitimacy and severity (Section 3.2). The expert-produced labels are the primary dataset analyzed in subsequent sections, so the basic properties of that corpus are summarized (Section 3.3).

3.1 Obtaining & Sandboxing Malware

Binaries obtained from Verisign's researchers, customers, and industry partners form the malware set used in this research.[1] We utilized 92,776 binaries representing roughly two years of collection prior to our mid-2013 analysis. These were sandboxed in a proprietary execution environment named *AutoMal*.[2] AutoMal is a typical sandbox environment and we expect that alternative dynamic

[1] http://www.verisigninc.com/en_US/cyber-security/index.xhtml

[2] A small quantity of domains/URLs enter the corpus without sandboxing, *e.g.,* lists of botnet C&C servers provided by industry partners.

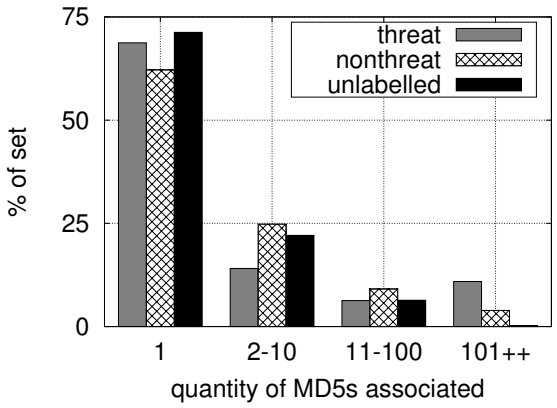

Fig. 1. Quantity of malware MD5s mapping to corpus endpoints, *i.e.,* 25% of non-threat endpoints were contacted by 2-10 unique malware binaries.

analysis tools such as Anubis[3], ChakraVyuha[4], and those described in [13] could fulfill a similar role. During execution AutoMal collects artifacts regarding the malware sample's interaction with the file system, memory, registry settings, and network. Though a more complete analysis suite is brought to bear over these outputs, this work is concerned primarily with the PCAP (packet capture) files that log activity over the network interface.

That PCAP file is post-processed with a parser pulling: (1) DNS lookups being performed on *(sub)domains* and (2) HTTP requests for full *URLs*. These endpoints are stored along with metadata as *potential threat indicators*. Note that a typical URL request will usually result in multiple potential indicators: the full URL (HTTP), the domain (DNS), and any subdomains (DNS).

3.2 Labeling Endpoints

Expert analysts are next brought to bear on the potential indicators with four main tasks: (1) choose a potential indicator, (2) evaluate if the potential indicator is a threat/non-threat in binary terms, (3) determine the broadest appropriate granularity for the aforementioned assessment, and (4) if a threat is present, annotate the severity of that threat.

Indicator choice: Analysts are free to choose the indicators they label as there is no forced queuing workflow. As of this writing, roughly $^1/_8$ of potential indicators have been labeled. The finite workforce desires their work to be impactful so analysts are likely to choose indicators that ...

[3] http://anubis.seclab.tuwien.ac.at/
[4] http://ibm.co/OFJyOA

– ... appear in many binaries. Per Figure 1, virtually no indicators mapping to 100+ binaries remain unlabeled.
– ... have recently been discovered, as the goal is to produce indicators useful in flagging active malware.
– ... look threat-like on the surface. Non-threats are useless to customers (although they aid research), so investigations on benign cases are wasteful.
– ... correspond to customer submitted binaries or acute exploits.

Thus the labeled portions likely over report the prevalence of threat endpoints. Fortunately, this bias does not affect our model construction. All the labels are fundamentally correct, only the class imbalance is slightly skewed. When the final trained model is run over unlabeled endpoints it predicts a 63% threat density (compared to 75% in labeled portions).

Binary label: When assessing a potential indicator, an analyst seeks to answer: *Is there a benign reason someone would access this resource?* Given that published threats are often installed on the client-side as alarms or blacklists the labeling process must be conservative to avoid false positives.

A number of utilities and datapoints (some subsequently captured in our features) are brought to bear. For example, reverse WHOIS lookups will be used to find web properties associated with those currently under inspection. The age of the domain will be considered, the host may be geo-located, *etc.*. Most critical is the content that resides at the endpoint. Endpoints hosting human readable/viewable content and APIs/services (*e.g.,* bandwidth tests, IP information services) usually are labeled as "non-threats". Regardless of how the malware might be using them these are resources which might be arrived at innocently. While it is easy to imagine edge cases, our characterization in Section 3.3 reveals a quite narrow spectrum of endpoints in practice, considerably simplifying the work of analysts and eliminating noise from our corpus.

Label granularity: An analyst is likely to first inspect the resource at the full URL path, *e.g.,* `sub.ex.com/file1.bin`. If that is found to be a "threat" then `sub.ex.com` or `ex.com` might also be ripe threat indicators. It is not difficult to imagine a malicious actor configuring their webserver so that for all `n`, the URL `sub.ex.com/file[n].bin` will redirect to the same binary. Then, this URL can be randomized at each repacking to evade naïve URL blacklists.

Often times corroborating evidence is a factor in making broader threat classifications. Past threat domains with a matching reverse WHOIS or a collection of URL granularity threats accumulating beneath a single (sub)domain are both strong evidence for a broader label. Observe that there are roughly 4× as many (sub)domain threats as URL ones in our corpus (Table 1). While broad labels often provide great utility, analysts must be sensitive to shared resources. For example, if `domain.com` is a popular public service that assigns subdomains to all of its customers, labeling the entire SLD as threatening could cause many false-positives. Indeed, malicious individuals often make use of such services to create Sybil-like identities at no/minimal cost (Section 3.3).

TOTAL			**28077**
domains		**21077**	**75.1%**
high-threat	5744	27.3%	
med-threat	107	0.5%	
low-threat	11139	52.8%	
non-threat	4087	19.4%	
urls		**7000**	**24.9%**
high-threat	318	4.5%	
med-threat	1299	18.6%	
low-threat	2005	28.6%	
non-threat	3378	48.3%	

Table 1. Corpus composition by type and severity

Severity label: If a potential indicator is labeled as a "threat" the analyst also annotates the *severity* of that threat. Note that this does not refer to the URL/domain resource but the underlying malware that contacted that resource. This determination is made using the full-fledged AutoMal output and other heuristics. The severity labels and their characteristic members include:

- LOW-THREAT: "nuisance" malware; ad-ware.
- MEDIUM-THREAT: untargeted data theft; spyware; banking trojans.
- HIGH-THREAT: targeted data theft; corporate and international espionage.

3.3 Corpus Composition

Analyst labeled data forms the basis of our future measurements and model-building. Therefore we now describe some basic properties of that set:

By the numbers: Table 1 best summarizes the 28,077 labeled endpoints that form our corpus, breaking them down by type and severity. There are 4× as many domain threat indicators as URL ones. This suggests that few malicious URL endpoints reside within (sub)domains that also serve benign purposes. Besides the fact URL file paths enable some structural features that domains do not, this type distinction is not significant.

Threats form 73.4% of all indicators, an extremely rich density relative to other classification tasks involving malicious URLs (*e.g.,* Internet-scale crawling). Figure 1 plots how endpoints distribute over the binaries which contact them. Although most indicators appear in just one binary, realize that this may be a response to the existence of indicator feeds. If malactors are aware the endpoints appearing in their malware will be effectively blacklisted then they are forced to frequently migrate domains. When an indicator does map to multiple MD5s it is evidence that URL/domain endpoints are a more persistent malware signature than MD5s. In the most dramatic case the now defunct subdomain

THREAT SLD	#	NONTHREAT	#
3322.ORG	2172	YTIMG.COM	1532
NO-IP.BIZ	1688	PSMPT.COM	1277
NO-IP.ORG	1060	BAIDU.COM	920
ZAPTO.ORG	719	GOOGLE.COM	646
NO-IP.INFO	612	AKAMAI.NET	350
PENTEST[...].TK	430	YOUTUBE.COM	285
SURAS-IP.COM	238	3322.ORG	243
FIREFOX[...].COM	221	AMAZONAWS.COM	191

Table 2. SLDs parent to the most number of endpoints, by class. These are/were all likely shared-use providers where broader SLD tagging would be ambiguous.

`os.solvefile.com` appeared in 1901 malware binaries. Classed as "low" severity the associated binaries were advertised as Firefox video codecs which were packaged with browser toolbars and modified Windows firewall settings.

Common SLDs: As a result of fine granularity threat labeling some higher-level entities appear multiple times in our corpus, *i.e.,* `a.ex.com` and `b.ex.com` might be two threat endpoints that reside beneath the `ex.com` second-level domain[5] (SLD). Table 2 enumerates those SLDs serving as parent to the greatest quantity of indicators. The fact these SLDs can not be assigned a blanket label makes them inherently interesting, a fact we will explore shortly.

This multiplicity also complicates our measurements and their presentation. While it is intuitive to develop features regarding an endpoint's SLD, when the same SLD appears hundreds or thousands of times in the corpus it lends tremendous statistical weight to a single feature value. Consider that `3322.org` is parent to ≈2400 labeled endpoints. Towards this we are careful to encode features that make apparent and leverage prior evidence about related entities. These prove critical to overall performance when considered in a multi-dimensional fashion. However, the flatter presentation of individual features to readers is sometimes less intuitive. For example, a registrar might host 2000+ malicious endpoints and all could be subdomains of a single malicious customer (Figure 6); saying very little about the actual reputation of that registrar. Ultimately our goal is to characterize and measure the workload of analysts, not necessarily make representative statements about the broader threat topology (as others have previously done [19, 25, 32]).

Content and acquisition trends: Since our feature extraction explicitly avoids endpoint content and its network acquisition (as others have researched; Section 2) it may be useful to casually address these topics. This perspective was

[5] We define a *second-level domain* to be the granularity just beneath the TLD (inclusive of the TLD). We treat all entries in the public suffix list (http://publicsuffix.org/list/) as TLDs, *i.e.,* `sld.com` and `sld.co.uk` are both SLDs.

FEATURE	TYPE	DESCRIPTION
TYPE	bool	Whether indicator is of "URL" or "DOMAIN" format
DOM_TLD	enum	Top-level domain (TLD) in which the domain resides
DOM_LENGTH	num	Length in chars. of the second-level domain (SLD)
DOM_ALPHA	num	Percentage of alphabetical domain chars. (vs. numeric)
DOM_DEPTH	num	Quantity of subdomains (*e.g.,* # of dots in full domain)
URL_LENGTH	num	Length of the URL in characters
URL_DEPTH	num	Number of subdirectories in the URL path
URL_EXTENSION	enum	File extension, if URL path concludes at a specific file
DOM_AGE	num	Time since the domain was registered
DOM_TTL_RENEW	num	Duration of domain registration (*e.g.,* years until renewal)
DOM_AUTORENEW	bool	Whether auto-renewal is enabled for the domain
DOM_REGISTRAR	enum	Registrar through which the domain was registered
DOM_BAYESIAN	num	Lower-order classifier over character n-grams in SLDs
DOM_REPUTATION	num	Quantity derived from past behavioral history of SLD

Table 3. Comprehensive feature listing; organization mirrors presentation order.

gleaned from Verisign's malware analysts who spend considerable time labeling endpoints and reverse engineering the malware they appear in.

We begin with *what actually resides at threat endpoints* and bin the results into three classes:

1. Malicious binaries: Initial exploits (*e.g.,* drive-by-downloads) tend to be small files, with larger payloads obtained after confirmation of compromise. Malware often obtains other binaries with orthogonal objectives as part of pay-per-install schemes [9].
2. Botnet C&C: Instructions coordinating botnet members in DDOS and spam attacks are common. Obfuscation, encryption, and unusual techniques are common. In one example, a threat endpoint was a HTML file whose source comments contained an encrypted instruction set. In another, a well-formed (*i.e.,* w/proper headers) JPG file was a wrapper for malicious data.
3. Drop sites: Though most network activity is DNS and HTTP GET requests, we observe some data theft operations performing HTTP POST actions as a means to return stolen information to the perpetrator.

Knowing that, *what resides at non-threat endpoints?* Malcode often queries web services to learn about the IP, geolocation, and bandwidth of the infected host (*e.g.,* `whatsmyip.org`). However, since these services are public and can be accessed under benign circumstances they cannot be treated as threats. Similarly, advertisement services are seen in click-fraud malware (*e.g.,* mechanizing ad click revenue). Finally, we observe image hotlinking in scare-ware and phishing campaigns as perpetrators try to reduce their own infrastructure footprint.

The inability to label such endpoints as malicious despite their use in malware underscores a weakness in the threat indicator approach. Non-dedicated and shared-use infrastructure is problematic. All entries in Table 2 are there precisely

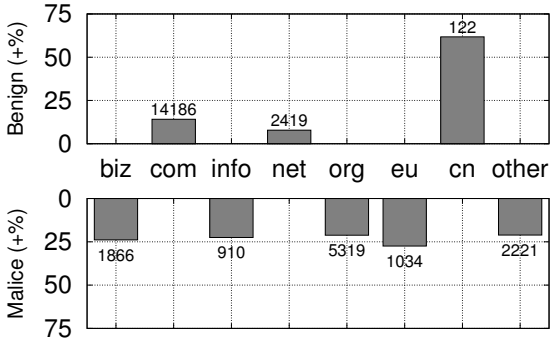

Fig. 2. Class patterns by TLD. Percentages are normalized to account for class imbalance, *i.e.,* the `cn` TLD is 62% more innocent than random expectation. Data labels indicate raw quantity by TLD.

because they are services which make it possible to cheaply serve content along distinct subdomains or URL paths. When a parent domain cannot be blacklisted because it has benign residents, URLs must be handled individually resulting in more analyst labor. Our reputation features are a direct response to such cases.

Finally, we address the *routing of malicious content*. Datapoints like traceroutes or the IP resolution of endpoints might prove helpful. However, these were not retained by our sandboxing mechanism and their dynamic nature make them impossible to recover in hindsight. Our more static perspective does make apparent the prevalent role of dynamic DNS (DDNS) services in serving threat endpoints. Six of the eight most common threat SLDs per Table 2 are DDNS providers. This includes the #1 offender (in terms of malicious children), `3322.org`, a now-defunct Chinese DDNS provider which was part of a botnet takedown [24]. It is intuitive why DDNS is preferred by malactors as it provides hosting agility and mobility.

Joined data: Aside from the indicator corpus, monthly "thin WHOIS" snapshots are also used. These snapshots provide basic registration data for domains while excluding registrant's personal information. Verisign's position as the authoritative registrar for the COM/NET/CC/TV zones permits us direct access to data covering 53% of our endpoints. Public access to bulk WHOIS information (including TLDs outside of Verisign's scope) is available via third-party re-sellers such as www.domaintools.com. Unlike DNS records, the WHOIS fields of interest tend to be quite stable. As such we consider the monthly snapshot immediately following an endpoint's discovery sufficient to glean registration data.

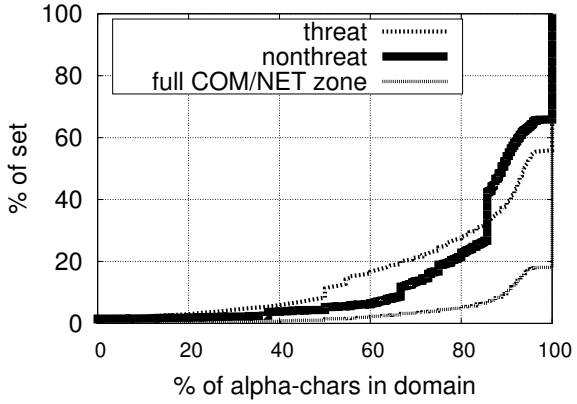

Fig. 3. CDF for the percentage of alphabetical characters in domain names, by class.

4 Feature Selection

The features of our model are enumerated in Table 3. We now describe the intuition behind selections and evaluate their single-dimension effectiveness. Features are organized into four groups: the lexical structure of the endpoint (Section 4.1), WHOIS properties of the domain (Section 4.2), token patterns (Section 4.3), and the aggregation of prior evidence into reputations (Section 4.4).

4.1 Lexical Structure

Surface properties of the indicator are straightforward and we first consider the TLD of the endpoint (`DOM_TLD`; Figure 2). A majority of indicators, regardless of class, reside in COM. Behaviorally speaking we see that traditionally cost-effective TLDs (*e.g.,* BIZ, INFO, and certain ccTLDs) most often lean towards being threats. The malicious inclinations of ORG are somewhat surprising but explained by the fact that TLD hosts several prevalent DDNS providers in our corpus. Although not immediately apparent from the percentage-wise presentation of Figure 2, nearly all non-threat indicators are in COM/NET.

Feature `DOM_LENGTH` counts the characters in the SLD. We suspected that dedicated threat domains might be longer as this could eliminate collisions for algorithmically generated names [3, 35]. Moreover, dedicated malware domains should have little concern for length as it relates to address memorability. As an isolated datapoint, shared-use settings and their differing selection criteria seem to have more statistical influence. While all domains are ≈17 characters at mean, aside from a cluster of threat domains around 128 characters in length, most over 33 characters tend to be non-threats. Because machine-generated names appear to be a small part of the problem space, the ratio of numeric to alphabetical characters is also less indicative than anticipated (`DOM_ALPHA`; Figure 3). See also Section 4.3 which is concerned with specific character choice and ordering.

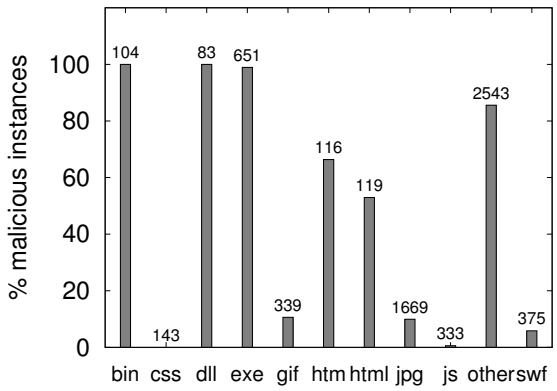

Fig. 4. Behavioral distribution over file extensions (URLs only). Data labels indicate raw quantity of occurences per file extension.

Whether or not a subdomain (*i.e.,* one or more beneath the SLD) is present for an endpoint (`DOM_DEPTH`) is significant in distinguishing shared use settings from dedicated infrastructure.[6] The most common number of subdomains, and that with the greatest density of malice, is one (*i.e.,* `sub.domain.com`). We observe subdomain quantities as high as 25, but beyond one subdomain it is non-threats which are most common.

Some features can only be calculated for URLs as they quantify properties along the file path. Both URL length in characters (`URL_LENGTH`) and the folder depth of the file path (`URL_DEPTH`) function similarly to their domain equivalents. More interesting is the endpoint's file extension, when present (`URL_EXTENSION`; Figure 4). We assume that these file extensions are indicative of file content although these relationships are not checked. Executable file types (*e.g.,* `bin`, `dll`, and `exe`) are almost always threats. Meanwhile, plain-text web documents (*e.g.,* `htm` and `html`) are behaviorally diverse, with image formats tending to be the most benign. Readers should note the large quantity of "other" extensions in Figure 4. While the most prevalent extensions are plotted, there is a great diversity of extensions observed, many of which are unfamiliar to the authors and may be "invented" for obfuscation purposes.

4.2 Domain WHOIS

The WHOIS information of endpoint domains produces some of the most indicative features. The age of a domain, *i.e.,* the time since the initial registration (`DOM_AGE`; Figure 5) is one such data point. Some 40% of threat domains are less than one year old. At median, threat domains are 2.5 years old compared to 12.5 years for non-threat ones. When older domains are threats it is characteristic of shared-use services or isolated compromises of established websites. It is

[6] In this analysis `www` is not considered a subdomain.

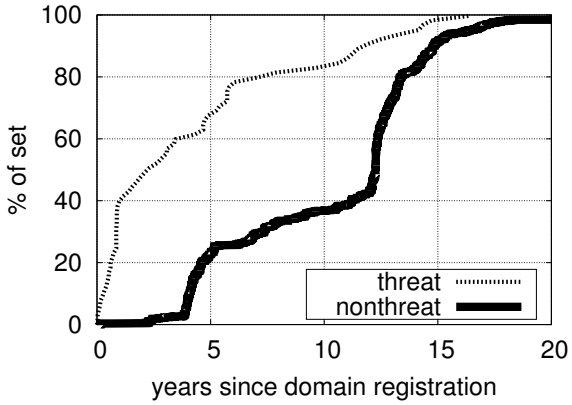

Fig. 5. CDF for domain age (time between registration and first observation in malware), by class; only calculated for COM/NET/CC/TV zones.

non-intuitive for purely malicious domains to pay multiple renewal fees before being put into active use. The lease period for a domain name (`DOM_TTL_RENEW`), while fixed by some registrars, is a variable that others expose to customers. If one is registering a domain only to serve malware he/she should presume it will quickly become blacklisted. Accordingly we see relatively few threat domains registered for more than a 5 year interval. Feature `DOM_AUTORENEW` is an option whereby a registrar will automatically extend a lease for a customer assuming payment information is stored. It performs quite poorly in practice perhaps due to inconsistent usage among registrars.

Motivated by prior work into the registration behavior of spammers [19] we also investigate domain registrars (`DOM_REGISTRAR`; Figure 6). Registrar MarkMonitor[7] has the most endpoints that appear in our corpus and nearly all are non-threats. This is logical: MarkMonitor serves some of the most popular web properties, providing enterprise-scale brand protection, managed DNS, and value added services that come at considerable cost relative to base registration fees. As [19] explains, factors like low cost, weak enforcement, or support for bulk registrations make certain registrars more attractive to malactors.

4.3 Bayesian n-gram

We speculated that certain keywords and character patterns might be indicative of class membership. For example, the character 3-gram "dns" could be common among DDNS providers. Moreover, n-grams may be able to distinguish human readable domains from machine generated ones based on character co-occurrence [3, 35]. Feature `DOM_BAYESIAN` is the output of a lower-order classifier

[7] http://www.markmonitor.com/

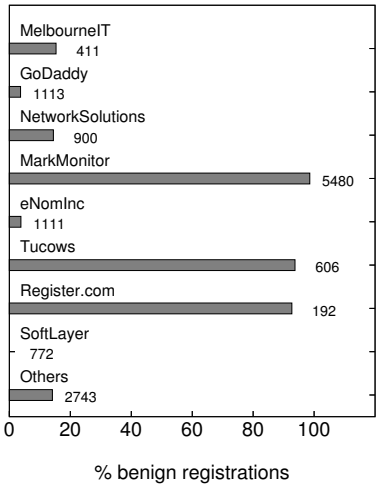

Fig. 6. Behavioral distribution over popular registrars. Data labels indicate quantity of registrations; analysis is limited to COM/NET/CC/TV domains.

FEATURE	GN-RTIO	GAIN↓
DOM_REPUTATION	0.509	0.749
DOM_REGISTRAR	0.073	0.211
DOM_TLD	0.087	0.198
DOM_AGE	0.051	0.193
DOM_LENGTH	0.049	0.192
DOM_DEPTH	0.126	0.186
URL_EXTENSION	0.134	0.184
DOM_TTL_RENEW	0.051	0.178
DOM_ALPHA	0.038	0.133
URL_LENGTH	0.048	0.028
URL_DEPTH	0.011	0.025
DOM_BAYESIAN	0.003	0.001
DOM_AUTORENEW	0.000	0.000

Table 4. Features sorted by info-gain (*i.e.,* KL divergence). Gain ratio is also provided, a metric sensitive to the quantity of unique values for enumerated features.

using established Bayesian document classification techniques using character n-grams for all $n \in [2,8]$. Only unique SLDs are used to train these models.

To gain insight into what this model captures we examine those n-grams that are common (having 25+ instances among unique SLDs) and indicative (having a strong leaning towards one class). We find very few character patterns are common among non-threat domains, with Table 5 presenting dictionary tokens from threat endpoints that meet these criteria.

4.4 Domain reputation

While our n-gram technique operates over unique SLDs we embrace SLD multiplicity by assigning each a reputation value calculated over prior evidence (DOM_REPUTATION; Figure 7). This feature is the single best performing with an information gain nearly 4× that of its closest competitor per Table 4. Reputations are calculated using a binary feedback model based on the Beta probability distribution [20]. Feedback are the expert labels assigned to previously labeled endpoints of the same SLD. Reputations are initialized at 0.5 and bounded on [0,1]. Though we calculate reputations only for SLDs, one could imagine doing similarly for subdomains and partial URL path granularity.

Since reputations are built atop the work of analysts, there would certainly be ramifications if we were to eliminate those analysts via an autonomous threat classifier. Though machine-produced labels could be used as feedback, fears of cascading errors suggest some degree of human supervision should be in place.

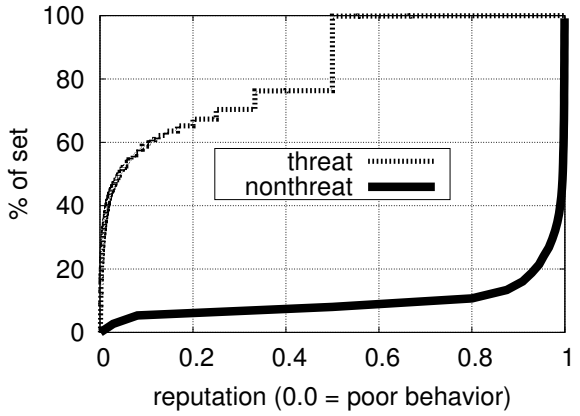

Fig. 7. CDF for domain reputation. Reputation is bound to [0,1] and initialized at 0.5. The reputation progression for an SLD might be 0.5 (initial) → 0.25 → 0.125 → 0.1 → 0.05 → 0.02. The first four of these values would be plotted in this CDF as reputation can only leverage prior evidence.

mail	soft
news	micro
apis	line
free	online
easy	wins
korea	update
date	port
yahoo	winsoft

Table 5. Dictionary tokens most indicative of threat domains per Bayesian document classification.

5 Training & Performance

Having enumerated its features we now train our classifier model (Section 5.1) and evaluate its performance at both the binary and severity tasks (Section 5.2).

5.1 Model Training

Our model is built using the Weka implementation of the Random Forest algorithm, an ensemble method over decision trees [18]. This technique was chosen because of its performance, human-readable output, and support for missing features. By examining component decision trees we can learn about which features are used in practice, and therefore which are effective over independent portions of the problem space. Approximately in-order of their influence, `DOM_REPUTATION`, `URL_DEPTH`, `DOM_TTL_RENEW`, and `DOM_LENGTH` features figure most prominently. We also observe that performance is not significantly impacted if WHOIS features (derived from an external dataset) are removed from consideration. It may be possible to exclude these features with minimal performance penalty as a matter of convenience. Table 4 formally ranks feature performance but does so in isolation without considering interdependence.

5.2 Classifier Performance

Performance metrics are produced via 10-fold cross validation over all labeled endpoints, with care taken to ensure that the Bayesian sub-classifier is trained in a consistent fashion. We now discuss results for the binary and severity tasks.

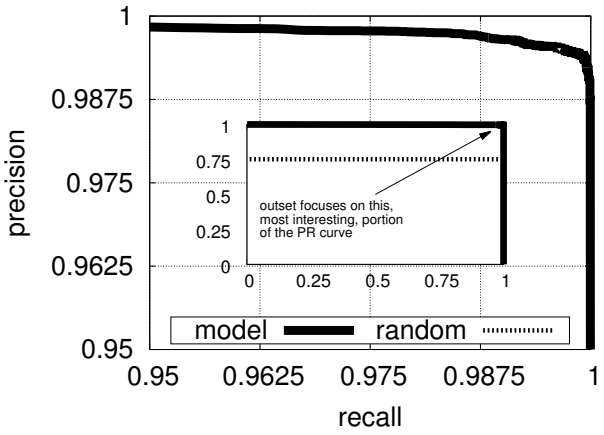

Fig. 8. (inset) Entire precision-recall curve for learned model; (outset) focusing on the interesting region of that precision-recall curve.

Binary task: The task of distinguishing "threat" versus "non-threat" endpoints is straightforward. Our model performs extremely well, making just 148 errors across the 28k element corpus, yielding a 99.47% accuracy. Figure 8 plots the precision-recall curve. The first classification error does not occur until 80% recall. Table 6 presents additional measures which are alternative perspectives confirming the strong performance.

Severity task: Recall that the malware binary associated with an endpoint is given a severity label, per Section 3.2 (a fact under-utilized given presentation difficulties with multi-class data). Our model achieves 93.2% accuracy at this task with the confusion matrix presented in Table 7. This confirms the model's viability as an analyst prioritization tool, bolstered by the other performance measures in Table 6. Such benchmarks are encouraging when considering the fact severity is a property of the malware binary and orthogonal to the endpoint under inspection. The features that drive the severity task closely mirror those of the binary one, though `DOM_REGISTRAR` takes on additional emphasis.

Production version: Due in large part to its excellent offline performance an online implementation of our model is in place and actively scoring new URL/domain threat indicators as they are discovered during malware sandboxing. Preliminary indications are that performance is comparable in both settings; confirming that non-organic parts of the corpus like indicators received from industry partners and bulk labeling play only a minor role. After this trial is complete we plan to expose our model-calculated scores to analysts and use them as a prioritization mechanism. After this we will be better poised to understand the benefits of our technique on analyst efficiency and workflow.

METRIC	BINARY	SEVERITY
accuracy	0.994	0.932
ROC area	0.997	0.987
F-measure	0.995	0.932
RMSE	0.068	0.161

Table 6. Information recall metrics for the binary and severity classification tasks

classified as → / actual label ↓	non	low	med	high
non-threat	7036	308	17	104
low-threat	166	12396	75	507
med-threat	8	89	1256	53
high-threat	36	477	64	5485

Table 7. Confusion matrix for severity classification task

6 Conclusions

Despite strong classifier performance work remains that could further improve its accuracy or extend its scope. Since DDNS is common among threat endpoints it would be helpful to better measure and leverage its use. A monitoring system could measure DNS "A record" stability and TTL values to gain further insight. Given our approach's ability to distinguish threat severity, investigating malware family identification (*e.g.,* Zeus banking trojan, Conficker, *etc.*) is also planned. Although this work has limited itself to network properties we imagine similar malware-driven classifiers operating over registry and filesystem indicators. It is also important to consider attack vectors which can circumvent endpoint blacklisting. For example, a news article's comment functionality might be used to embed C&C instructions on a popular news website which cannot be blacklisted. How to best prevent shared-use, user-generated, and collaborative functionalities from such manipulation deserves future attention.

Though related to efforts in other security contexts, our work herein represents the first known analysis of the network endpoints contacted by malware. Properly vetted, these domains and URLs are a rich source of "indicators" to fingerprint malcode. These indicators are already being effectively used within centralized network monitoring alert services. However, this approach is burdened by the non-trivial expert labor needed to distinguish the benign "non-threat" endpoints that are sometimes contacted by malware.

Using an analyst labeled corpus of 28k+ domains/URLs derived from ≈100k malware binaries, we simultaneously characterized these endpoints while developing features towards an autonomous classifier. Rather than trying to accommodate dynamic network routing and content considerations, we utilize a static metadata approach that leverages endpoint's lexical structure, WHOIS data, and prior behavioral experiences. We observe that malactors commonly leverage dynamic DNS and other cost-sensitive solutions. Shared-use settings prove particularly challenging as perpetrators utilize open infrastructure services that are also host to benign clients. Regardless, we are able to produce a classifier that is 99%+ accurate at predicting binary threat status and 93%+ accurate at predicting threat severity. The resulting model will prioritize manual analyst workload, eliminate some portions of it entirely, and shows promise as a means to grey-list endpoints beyond those explicitly identified as malware signatures.

Acknowledgments

We thank Verisign iDefense team members Ryan Olsen and Trevor Tonn for their assistance in obtaining and interpreting the malware corpus. Verisign Labs director Allison Mankin is also acknowledged for her guidance on this project.

References

1. M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a dynamic reputation system for DNS. In *Proc. of 19th USENIX Sec. Sym.*, 2010.
2. M. Antonakakis, R. Perdisci, W. Lee, N. V. II, and D. Dagon. Detecting malware domains at the upper DNS hierarchy. In *Proc. of 20th USENIX Sec. Sym.*, 2011.
3. M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From throw-away traffic to bots: Detecting the rise of DGA-based malware. In *Proceedings of the 21st USENIX Security Symposium*, 2012.
4. M. Bailey, J. O. J. Andersen, Z. Mao, F. Jahanian, and J. Nazario. Automated classification and analysis of Internet malware. In *RAID '07: Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection*, 2007.
5. U. Bayer, P. M. Comparetti, C. Hlauschek, C. Krügel, and E. Kirda. Scalable, behavior-based malware clustering. In *NDSS '09: Proceedings of the 16th Network and Distributed System Security Symposium*, 2009.
6. L. Bilge, D. Balzarotti, W. K. Robertson, E. Kirda, and C. Kruegel. Disclosure: Detecting botnet command and control servers through large-scale NetFlow analysis. In *ACSAC' 12: Proc. of the 28th Annual Comp. Security Apps. Conf.*, 2012.
7. L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. EXPOSURE: Finding malicious domains using passive DNS analysis. In *NDSS '11: Proceedings of the 18th Network and Distributed System Security Symposium*, 2011.
8. A. Blum, B. Wardman, T. Solorio, and G. Warner. Lexical feature based phishing URL detection using online learning. In *AISec '10: Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*, 2010.
9. J. Caballero, C. Grieber, C. Kreibich, and V. Paxson. Measuring pay-per-install: The commoditization of malware distribution. In *Proceedings of the 20th USENIX Security Symposium*, 2011.
10. Center for Strategic and International Studies and McAfee. The economic impact of cybercrime and cyber espionage. http://www.mcafee.com/us/resources/reports/rp-economic-impact-cybercrime.pdf, 2013.
11. J. Chang, K. K. Venkatasubramanian, A. G. West, and I. Lee. Analyzing and defending against web-based malware. *ACM Computing Surveys*, 45(4), 2013.
12. K. Dai, L. Zhao, Z. Nie, J.-R. Wen, L. Wang, and Y. Li. Detecting online commercial intention (OCI). In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, 2006.
13. M. Egele, T. Scholte, E. Kirda, and C. Kruegel. A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys*, 44(2), 2008.
14. M. Felegyhazi, C. Kreibich, and V. Paxson. On the potential of proactive domain blacklisting. In *LEET '10: Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats*, 2010.
15. G. Gu, R. Perdisci, J. Zhang, and W. Lee. BotMiner: Clustering analysis of network traffic for protocol and structure independent botnet detection. In *Proceedings of the 17th USENIX Security Symposium*, 2008.

16. G. Gu, P. Porris, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: Detecting malware infection through IDS-driven dialog correlation. In *Proceedings of the 16th USENIX Security Symposium*, 2007.
17. G. Gu, J. Zhang, and W. Lee. BotSniffer: Detecting botnet command and control channels in network traffic. In *NDSS '08: Proceedings of the 15th Network and Distributed System Security Symposium*, 2008.
18. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.
19. S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck. Understanding the domain registration behavior of spammers. In *IMC '13: Proceedings of the 13th ACM Conference on Internet Measurement*, 2013.
20. A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled eCommerce Conference*, 2002.
21. C. Kolbitsch, P. M. Comparetti, C. Kruegel, E. Kirda, X. Zhou, and X. Wang. Effective and efficient malware detection at the end host. In *Proceedings of the 18th USENIX Security Symposium*, 2009.
22. D. Kong and G. Yan. Discriminant malware distance learning on structural information for automated malware classification. In *KDD '13: Proceedings of the 19th SIGKDD Conference on Knowledge Discovery and Data Mining*, 2013.
23. A. E. Kosba, A. Mohaisen, A. G. West, and T. Tonn. ADAM: Automated detection and attribution of malicious webpages (poster). In *CNS '13: Proc. of the 1st IEEE Conference on Communications and Network Security*, 2013.
24. B. Krebs. Malware dragnet snags millions of infected PCs. *Krebs on Security Blog*, September 2012. http://krebsonsecurity.com/2012/09/malware-dragnet-snags-millions-of-infected-pcs/.
25. K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, 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 on Security and Privacy*, 2011.
26. J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. In *KDD '09: Proceedings of the 15th SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009.
27. D. K. McGrath and M. Gupta. Behind phishing: An examination of phisher modi operandi. In *LEET '08: Proceedings of the 1st USENIX Workshop on Large-scale Exploits and Emergent Threats*, 2008.
28. A. Mohaisen, O. Alwari, and M. Larson. A methodical evaluation of antivirus scans and labels. In *WISA '13: Proceedings of the 14th International Workshop on Information Security Applications*, 2013.
29. A. Ntoulas, M. Najor, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW '06: Proceedings of the 15th International World Wide Web Conference*, 2006.
30. N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All your iFRAMEs point to us. In *Proceedings of the 17th USENIX Security Symposium*, 2008.
31. N. Provos, D. McNamee, P. Mavrommatis, K. Wang, N. Modadugu, et al. The ghost in the browser analysis of web-based malware. In *HotBots '07: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets*, 2007.
32. G. Stringhini, C. Kruegel, and G. Vigna. Shady paths: Leveraging surfing crowds to detect malicious web pages. In *CCS '13: Proceedings of the 20th ACM Conference on Computer and Communications Security*, 2013.

33. K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time URL spam filtering service. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2011.
34. A. G. West, A. Agrawal, P. Baker, B. Exline, and I. Lee. Autonomous link spam detection in purely collaborative environments. In *WikiSym '11: Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, 2011.
35. S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan. Detecting algorithmically generated malicious domain names. In *IMC '10: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, 2010.
36. G. Yan, N. Brown, and D. Kong. Exploring discriminatory features for automated malware classification. In *DIMVA '13: Proceedings of the 10th Conference on Detection of Intrusions and Malware & Vulnerability Assessment*, 2013.
37. T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In *ACSAC' 13: Proceedings of the 29th Annual Computer Security Applications Conference*, 2013.