MADE: Security Analytics for Enterprise Threat Detection

Alina Oprea
Northeastern University
a.oprea@northeastern.edu

Zhou Li
University of California, Irvine
zhou.li@uci.edu

Robin Norris
EMC/Dell CIRC
robin.norris@emc.com

Kevin Bowers
RSA
kbowers@rsa.com

ABSTRACT

Enterprises are targeted by various malware activities at a staggering rate. To counteract the increased sophistication of cyber attacks, most enterprises deploy within their perimeter a number of security technologies, including firewalls, anti-virus software, and web proxies, as well as specialized teams of security analysts forming Security Operations Centers (SOCs).

In this paper we address the problem of detecting malicious activity in enterprise networks and prioritizing the detected activities according to their risk. We design a system called MADE using machine learning applied to data extracted from security logs. MADE leverages an extensive set of features for enterprise malicious communication and uses supervised learning in a novel way for prioritization, rather than detection, of enterprise malicious activities. MADE has been deployed in a large enterprise and used by SOC analysts. Over one month, MADE successfully prioritizes the most risky domains contacted by enterprise hosts, achieving a precision of 97% in 100 detected domains, at a very small false positive rate. We also demonstrate MADE's ability to identify *new malicious activities* (18 out of 100) overlooked by state-of-the-art security technologies.

ACM Reference Format:
Alina Oprea, Zhou Li, Robin Norris, and Kevin Bowers. 2018. MADE: Security Analytics for Enterprise Threat Detection. In *2018 Annual Computer Security Applications Conference (ACSAC '18), December 3–7, 2018, San Juan, PR, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3274694.3274710

1 INTRODUCTION

Criminal activity on the Internet is expanding at nearly exponential rates. With new monetization capabilities and increased access to sophisticated malware through toolkits, the gap between attackers and defenders continues to widen. As highlighted in a recent Verizon Data Breach Investigations Report (DBIR) [3], the detection deficit (difference between an attacker's time to compromise and a defender's time to detect) is growing. This is compounded by the ever-growing attack surface as new platforms (mobile, cloud, and IoT) are adopted and social engineering gets easier.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ACSAC '18, December 3–7, 2018, San Juan, PR, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6569-7/18/12…$15.00
https://doi.org/10.1145/3274694.3274710

These concerns affect not only individuals, but enterprises as well. The enterprise perimeter is only as strong as its weakest link and that boundary is becoming increasingly fuzzy with the prevalence of remote workers using company-issued computers on networks outside of the enterprise control. Enterprises attempt to combat cyber attacks by deploying firewalls, anti-virus agents, web proxies and other security technologies, but these solutions cannot detect or respond to all malware. Large organizations employ "hunters" or tier 3 analysts (part of the enterprise Security Operations Center – SOC) [44] to search for malicious behavior that has evaded their automated tools. Unfortunately, this solution is not scalable, both due to the lack of qualified people and the rate at which such malware is invading the enterprise.

In this paper we address the problem of detecting new malicious activity in enterprise networks and prioritizing the detected activities for operational use in the enterprise SOC. We focus on a fundamental component of most cyber attacks – malware command-and-control communication (C&C). Command-and-control (also called *beaconing* [30]) is the main communication channel between victim machines and attacker's control center and is usually initiated by victim machines upon their compromise. We design a system MADE (**M**alicious **A**ctivity **D**etection in **E**nterprises) that uses supervised learning techniques applied to a large set of features extracted from web proxy logs to proactively detect network connections resulting from malware communication. Enterprise malware is increasingly relying on HTTP to evade detection by firewalls and other security controls, and thus it is natural for MADE to start from the web proxy logs collected at the enterprise border. However, extracting intelligence from this data is challenging due to well-recognized issues including: large data volumes; inherent lack of ground truth as data is unlabeled; strict limits on the amount of alerts generated (e.g., 50 per week) and their accuracy (false positive rates on the order of 10^{-4}) for systems deployed in operational settings. An important requirement for tools such as MADE is to provide interpretability of their decisions, as their results are validated by SOC through manual analysis. This precludes the use of models known to provide low interpretability of their results, such as deep neural networks.

In collaboration with tier 3 security analysts at a large enterprise, we designed MADE to overcome these challenges and meet the SOC requirements. To address the large data issue, we filter network communications that are most likely not malicious, for instance connections to CDNs and advertisement sites, as well as popular communications to well-established external destinations. To address the ground truth issue, we label the external domains using several threat intelligence services the enterprise subscribed to. Finally, to obtain the accuracy needed in operational settings, MADE

https://t.me/learningnets

leverages interpretable classification models in a novel way, by training only on malicious and unknown domains, and predicting the probability that an unknown domain is malicious. Domains with highest predicted probabilities can then be prioritized in the testing stage for investigation by the SOC. In designing MADE, we defined an extensive set of features to capture various behaviors of malicious HTTP enterprise communication. In addition to generic, well-known malware features, MADE proposes a set of enterprise-specific features with the property of adapting to the traffic patterns of each individual organization. MADE performs careful feature and model selection to determine the best-performing model in this context.

MADE has been used in operational setting in a large enterprise with successful results. In our evaluation, we demonstrate that over one month MADE achieves 97% precision in the set of 100 detected domains of highest risk, at the false positive rate of $6 \cdot 10^{-5}$ (3 in 50,000 domains in testing set). MADE detects well-known malicious domains (similar to those used in training), but also has the ability to identify entirely new malicious activities that were unknown to state-of-the-art security technologies (18 domains in the top 100 are new detections by MADE).

2 BACKGROUND AND OVERVIEW

2.1 Enterprise Perimeter Defenses

Enterprises deploy network-level defenses (e.g., firewalls, web proxies, VPNs) and endpoint technologies to protect their perimeter against a wide range of cyber threats. These security controls generate large amounts of security logs that are typically stored in a centralized security information and event management (SIEM) system. Large enterprises recognize that these protections are necessary, but not sufficient to protect themselves against continuously evolving cyber attacks. To augment their cyber defense capabilities, they employ incident response teams including security analysts tasked to analyze alerts and detect additional suspicious activities. Most of the time, security analysts use the collected security logs for forensic investigation. Once an attack is detected by some external mechanism, they consult the logs to detect the root cause of the attack.

We are fortunate to collaborate with the Security Operations Center (SOC) of a large, geographically-distributed enterprise and obtain access to their security logs. The tier 3 security analysts of the SOC utilize a variety of advanced tools (host scanning, sandboxes for malware analysis, threat intelligence services), but they rely quite extensively on manual analysis and their domain expertise for identifying new malicious activities in the enterprise. In the broadest sense, the goal of our research is to *design intelligent algorithms and tools for the SOC analysts that automatically detect and prioritize most suspicious enterprise activities*.

2.2 Problem definition and adversarial model

More concretely, our goal is to use machine learning (ML) to proactively identify and prioritize external network communications related to a fundamental component of most enterprise cyber attacks. Our focus is on malware command-and-control (C&C) communication over HTTP or HTTPs, also called *beaconing* [30]. As enterprise firewalls and proxies typically block incoming network connections, establishing an outbound malware C&C channel is the main communication mechanism between victims and attackers. This allows malware operators to remotely control the victim machines, but also to manually connect back into the enterprise network by using, for instance Remote Access Tools [21]. C&C is used extensively in fully automated campaigns (e.g., botnets or ransomware such as Wannacry [34]), as well as in APT campaigns (e.g., [42]).

C&C increasingly relies on HTTP/HTTPs channels to maintain communication stealthiness by hiding among large volumes of legitimate web traffic. Thus, it is natural for our purposes to leverage the web proxy logs intercepting all HTTP and HTTPs communication at the border of the enterprise network. This data source is very rich, as each log event includes fields like the connection timestamp, IP addresses of the source and destination, source and destination port, full URL visited, HTTP method, bytes sent and received, status code, user-agent string, web referer, and content type. We design a system MADE (**M**alicious **A**ctivity **D**etection in **E**nterprises) that uses supervised learning techniques applied to a large set of features extracted from web proxy logs to proactively detect external network connections resulting from malware communication.

In terms of adversarial model, we assume that remote attackers have obtained at least one footprint (e,g, victim machine) into the enterprise network. Once it is compromised, the victim initiates HTTP or HTTPS communication from the enterprise network to the remote attacker. The communication from the victim and response from the attacker is logged by the web proxies and stored in the SIEM system. We assume that attackers did not get full control of the SIEM system and cannot manipulate the stored security logs. That will result in a much more serious breach that is outside our scope. If enterprise proxies decrypt HTTPS traffic (a common practice), our system can also handle encrypted web connections.

Designing and deploying in operation a system like MADE is extremely challenging from multiple perspectives. Security logs are large in volume and more importantly, there is an inherent lack of ground truth as data is unlabeled. Existing tools (such as VirusTotal [2] and Alexa [6]) can be used to partially label a small fraction of data, while the large majority of connections are to unknown domains (they are not flagged as malicious, but cannot be considered benign either). Our goal is to prioritize among the unknown domains the most suspicious ones and provide meaningful context to SOC analysts for investigation. Finally, MADE is intended for use in production by tier 3 SOC analysts. This imposes choice of interpretable ML models, as well as strict limits on the amount of alerts generated (at most 50 per week). Achieving high accuracy and low false positive rates when most of the data has unknown labels is inherently difficult in machine learning applications to cyber security [57].

2.3 System Overview

The MADE system architecture is in Figure 1 and consists of the following components:

Training (Section 3). For training MADE, historical web proxy logs over three months are collected from the enterprise SIEM. (1) In the *Data Filtering and Labeling* phase, connections that are unlikely C&C traffic (e.g., CDN, adware, popular domains) are excluded from the dataset and the malicious domains in the collected data are labeled using Threat Intelligence services such as VirusTotal. (2) In *Feature Extraction*, a large number of features (89) are extracted using the domain expertise of SOC, measurement on our

https://t.me/learningnets

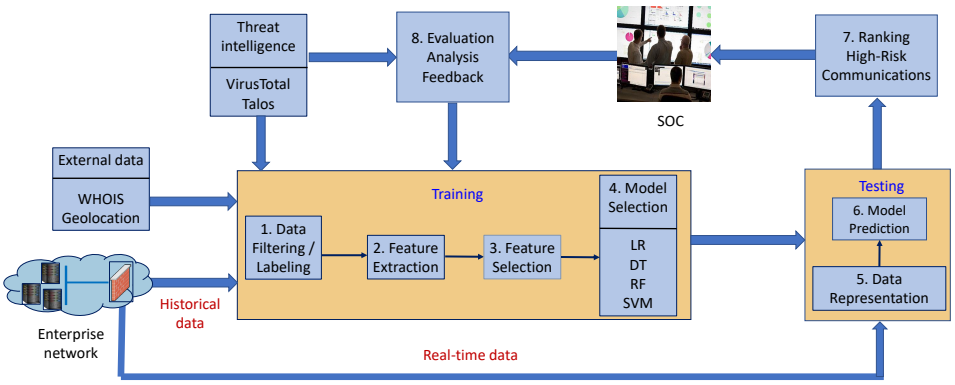

Figure 1: System architecture.

System	Malware Type	Features	Method	Dataset size	Accuracy	Detect new malware
ExecScent [45]	Known malware communication	URL, UA, Header values	Clustering	158 million events per day	66 TP / 13 FP (Dataset 1) 32 TP / 26 FP (Dataset 2) 2 TP / 23 FP (Dataset 3)	No
Oprea et al. [47]	Periodic malware communication Malware delivery	Inter-arrival time Communication, UA WHOIS	Belief propagation Prioritization (top 375)	660GB per day Two months	289 TP / 86 FP	Yes
Bartos et al. [11]	Known malicious	Inter-arrival time URL, Communication, Lexical Scaling and shifting feature transformation	Classification on sequences of flows	15 million flows	90% precision 67% recall	No
BAYWATCH [30]	Periodic malware communication	Inter-arrival time Lexical	Time-series auto-correlation Classification Prioritization (top 50)	30 billion events	48 TP / 2 FP	No
MADE Our approach	**Generic malware communication**	**Communication, Domain URL, UA, Result code Referer, Content type WHOIS, Geolocation**	**Classification Prioritization (top 100)**	**300 million events per day 15 billion events total**	**97 TP / 3 FP**	**Yes**

Table 1: Comparison with previous systems for enterprise malware detection using web proxy logs. Legend: TP (True Positives), FP (False Positives).

dataset, public reports on malware operations, and previous research on malware analysis in the academic literature. We complement features extracted from HTTP logs with additional attributes available from external data sources, e.g., domain registration information from WHOIS data and ASN information from MaxMind [1]. (3) In *Feature Selection*, we rank the set of features and select a subset with highest information gain. (4) Finally, *Model Selection* analyzes various metrics of interest for four classes of supervised learning models (Logistic Regression, Decision Trees, Random Forest, and SVM) and selects the best performing ML model for our system.

Testing (Section 4). In testing, new real-time data collected during another month is used into the ML model learned in the training phase. In (5) *Data Representation* the selected features are extracted from the new testing data, while (6) *Model Prediction* outputs domain risk scores using the trained ML model. (7) *Ranking High-Risk Communications* prioritizes the most suspicious connections according to the SOC budget (10 per business day or 50 per week).

Evaluation (Section 4). In (8) *Evaluation, Analysis, and Feedback* the list of most suspicious domains is manually investigated by tier 3 SOC analysts and feedback is provided on MADE's detections.

2.4 Comparison with previous work

As malware communication has been one of the most extensively studied topic in cyber security for many years, the reader might wonder what MADE contributes new to this area. We would like to mention upfront the new features of MADE and how it compares to previous work. MADE is designed to detect **enterprise malware communication** and is the result of close collaboration over several years with the enterprise SOC in all aspects of the project from problem definition, to algorithm design, evaluation, and integration into operational settings. A number of relatively recent papers are also specifically designed to detect malicious communication in enterprise settings from web proxy log analysis. These are the closest systems related to MADE and they are surveyed in Table 1.

One of the first systems in this space is ExecScent [45], which executes malware samples in a sandbox and constructs communication templates. ExecScent has the benefit of detecting new malware variants by similarity with existing samples, but it is not designed to detect new malware. Oprea et al. [47] apply belief propagation to detect malware periodic communication and malware delivery in multi-stage campaigns. Bartos et al. [11] design a system for generic malware detection that classifies legitimate and malicious web flows using a number of lexical, URL, and inter-arrival timing features computed for a sequence of flows. They propose a new feature representation invariant to malware behavior changes, but unfortunately the method does not retain feature interpretability. BAYWATCH [30] uses supervised learning based on inter-arrival timing and lexical features to detect periodic C&C or beaconing communication. As

https://t.me/learningnets

we can see from the table, MADE has several interesting characteristics: (1) MADE uses the most extensive set of features to date for identifying HTTP malware communication, carefully crafted based on the SOC domain expertise; (2) MADE can identify various malware classes since it does not rely on timing or lexical features; (3) MADE achieves best precision from all prioritization-based systems (at similar false positive rates); (4) MADE identifies new malware not available during training and unknown to the community at the time of detection; (5) MADE achieves interpretability of ML results and can be used successfully by SOC domain experts.

MADE can detect general malware communication in enterprise network. As discussed, previous systems are crafted for specific types of malware communication protocols (either with periodic timing [30, 47], or those that are similar to malware available in training [11, 45]). Therefore, there is no existing system with which we can meaningfully compare the results achieved by MADE. Our hope is that the community will create benchmarks including datasets and algorithms publicly available in this space. But at the moment we are limited in deploying existing systems in our environment and comparing them explicitly with MADE.

2.5 Ethical considerations

The enterprise SOC provided us access to four months of data from their SIEM system. Employees consent to their web traffic being monitored while working within the enterprise perimeter. Our dataset did not include any personally identifiable information (PII). Our analysis was done on enterprise servers and we only exported aggregated web traffic features (as described in Table 3).

3 MADE TRAINING

We obtained access to the enterprise SIEM for a four month period in February-March and July-August 2015. The number of events in the raw logs is on average 300 million per day, resulting in about 24TB of data per month. We use one month of data (July) for training as we believe it is sufficient to learn the characteristics of the legitimate traffic and one month (August) for testing. To augment the set of malicious domains, we used the February and March data to extract additional malicious connections, which we include in the training set. We extract a set of external destinations (domain names or FQDN) contacted by enterprise machines during that interval. After data filtering, we label each destination domain as *malicious* (M), *benign* (B), or *unknown* (U) using several public services (Section 3.1). The large majority of domains (more than 90%) are unknown at this stage. Each domain is represented by a set of numerical and categorical features extracted from different categories (Sections 3.2 and 3.3).

We express our problem in terms of ML terminology. Let $\mathcal{D}_{tr} = \{(x_1, L_1), \ldots, (x_n, L_n)\}$ be the training set with domain i having feature set x_i and label $L_i \in \{\text{M,B,U}\}$. Our aim is to train an ML model $f \in \mathcal{H}$, where $\mathcal{H}$ is the hypothesis space defined as the set of all functions from domain representations X to predictions Y. f is selected to minimize a certain loss function on the training set:

$$\min_{f \in \mathcal{H}} \ell(f(x_i, L_i, \theta)) + \lambda\omega(\theta)$$

where θ is the model parameter, and $\lambda\omega(\theta)$ a regularization term.

This is a general supervised learning setting and can be instantiated in different ways in our system. To be concrete, let us start by a simple example. Most previous work on detecting malicious domains (e.g., [8, 10, 12, 13, 51]) employ classification models to distinguish malicious and benign domains. In the above framework, we could define the training set $\mathcal{D}_{tr}$ as the domains with labels M and B, $\mathcal{H}$ the set of SVM classifiers $h : X \rightarrow \{\text{M,B}\}$, and the loss function ℓ the hinge loss. In the testing phase, a new set of domains $\mathcal{D}_{test} = \{x'_1, \ldots, x'_m\}$ is observed and our goal is to generate predictions on these domains using the optimal SVM classifier f learned during training. We compute the prediction that the domain is malicious or benign $L'_i = f(x'_i)$.

This approach of distinguishing malicious and benign domains works well in general, since these domains have different characteristics. Unfortunately, this approach is not sufficient in our setting to generate meaningful alerts for the SOC. If we simply train a model on a small percentage of the domains (including only the malicious and benign classes), our results on the unknown domains would be unreliable (since such domains are not even part of the training set). Instead, MADE's goal is to *identify and prioritize the most suspicious domains among the unknown class* and we discuss how we adapt this general framework to our setting in Section 3.4.

3.1 Data Filtering and Labeling

For each external domain contacted by an enterprise machine, the web proxy logs include: connection timestamp, IP addresses and ports of the source and destination, full URL visited, HTTP method, bytes sent and received, status code, user-agent string, web referer, and content type. The enterprise maintains separately the first date each FQDN domain was observed on the enterprise network. During July and August, a total number of 3.8M distinct FQDNs were included in the proxy logs. We applied a number of filters to restrict our attention to potential domain of interest that could be related to malware communications (see Table 2 for statistics):

- *Recent domains:* We focus on *recent domains*, appearing first time on the enterprise in the last two weeks. Our motivation is three-fold: malicious domains have short lifetime; long-lived malicious domains are more likely to be included in threat intelligence services; and we are interested in new malware trends. There are about 1.1 million distinct recent domains contacted in July and August.
- *Popular domains:* We exclude for analysis popular domains contacted by more than 50 enterprise hosts per day. Most enterprise infections compromise a few hosts and large-scale compromises can be detected by perimeter defenses and other existing tools.
- *CDN domains:* We filter out the domains hosted by reputable CDN services by using several public lists [24, 38]. Based on discussions with the enterprise SOC analysts, most CDN domains are considered safe and are unlikely to be used as malicious communication channels. Reputable CDN service providers employ multi-layer defenses for malware detection and usually take remediation measures quickly. Nevertheless, recent research revealed that adversaries started to deliver potentially unwanted programs (PUPs) using CDN domains [35]. In the future, attackers could host their command-and-control infrastructure on CDN domains, and new defenses need to be designed against this threat.

https://t.me/learningnets

	Category	Training		Testing		Total
	Recent domains	582,657	July	532,706	August	1,115,363
Filter	Popular domains	581,814		531,992		1,113,806
	CDN	576,064		528,614		1,104,678
	Ad traffic	572,186		525,469		1,097,655
	Low connections	252,476	July	218,879	August	471,355
Labeling	Benign (Alexa 10K)	16,052	6.57%	15,779	7.2&	31,831
	Malicious (VirusTotal ≥ 3)	603	0.24%	516	0.23%	1,119
	Unknown	227,555	93.18%	202,584	93.55%	430,139
	VirusTotal 1 or 2	8,266				
	Additional malicious	1,152	Feb - March			

Table 2: Dataset statistics.

- *Legitimate advertisement traffic:* We exclude reputable advertisement domains based on EasyList [4] and EasyPrivacy [5]. Prior research has shown that malicious content can be delivered through online advertisements [66]. However, malicious advertisements usually originate from lower-reputation ad networks, participating in the arbitration process. We filter out the reputable domains in the two advertisement lists, owned by legitimate advertising companies, as we believe that they are unlikely to be associated with malware command-and-control centers. Malicious communication with lower-reputable ad networks are still within the scope of MADE.
- *Domains with few connections:* We are mainly targeting enterprise C&C for MADE and we restrict our attention to domains that have at least 5 connections over a one-month period. We believe this to be a minimal assumption, as most C&C domains see more traffic (at least several connections per day). This has a significant impact on data reduction (more than 50%).

After all filtering steps, there are a total of 471K distinct FQDN contacted in July and August.

Data Labeling. We conservatively label as *benign* the domains with their second-level domain in top 10K Alexa (6.57% of training and 7.2% of testing domains). These are popular sites with strong security protections in place. We queried all other domains to VirusTotal, a cloud-based antivirus engine. We label *malicious* all domains flagged by at least three anti-virus engines in VirusTotal. This resulted in 1,119 domains labeled as malicious in July and August, representing 0.24% of all domains contacted during that time interval (after filtering). The enterprise of our study is at the high-end spectrum in terms of cyber defense, which manifests into low rate of malicious activities. To augment the set of malicious domains, we applied the same procedure to data from February and March and identified a set of 1,152 malicious domains in that period. We consider *unknown* all domains with a score of 0 on VirusTotal that are not already labeled as benign. They represent the large majority of traffic (92.88%). Domains with scores of 1 and 2 are not included in training, as we are not certain if they are indeed malicious (many times low-profile adware or spyware campaigns receive scores of 1 or 2, but are not actually malicious).

3.2 Feature Extraction

In this section, we elaborate on the large set of features (89) we used for representing FQDN domains in our dataset. See Table 3 for a list and description of features. The majority of the features are extracted from the web proxy logs of the enterprise, and we call them *internal features*. We extract additional *external features* related to domain registration and IP geolocation. To the best of our knowledge, our set of features is the most extensive to date for detecting enterprise C&C communication over HTTP. We leveraged feedback from tier 3 SOC analysts, as well as previous academic research, public reports on malware operations, and measurement of our dataset to define these features. The ones that are novel compared to previous work are highlighted in bold in Table 3. In addition to generic malware features applicable for malware detection in any environment, we define a set of *enterprise-specific features* (see column "'Enterprise" in Table 3) that capture traffic characteristics of individual enterprises.

Internal Features. We provide first a description of the internal features extracted from web proxy logs, grouped into seven categories:

Communication structure. These are features about communication structure for domains contacted by enterprise hosts. Specifically, we count the number of enterprise hosts contacting the domain and compute several statistics per host (Avg/Min/Max and Ratio) for bytes sent and received, and POST and GET connections. Malicious domains have different communication structure in number of connections, POST vs GET, and number of bytes sent and received compared to legitimate domains. For instance, the median average connections per host to legitimate domains is 27, while for malicious domains is 13.5. Malicious domains exhibit more POST requests (on average 43) than legitimate ones (only 12.58 on average). Also, malicious domains have higher ratios of bytes sent over received (five times higher than legitimate domains).

Domain structure. Malicious domains are not uniformly distributed across all TLDs (top-level domains). A well-known strategy for attackers is to register domains on inexpensive TLDs to reduce their cost of operation [27]. As such, we extract the TLD from the domain name and consider it as a categorical feature. We also consider the number of levels in the domain, the number of sub-domains on the same SLD (second-level domain), and domain name length.

URL features. URL-derived features are good indicators of web attacks [36]. In enterprise settings, malware is increasingly designed to communicate with external destinations by manipulating URLs [50]. URL path (the substring after domain name), folder, file name, extension, parameters, and fragment fields can all be used to update host status or exfiltrate host configurations. Therefore, we calculate the overall statistics per URL for these attributes. We also count the number of distinct URLs, the fraction of URLs with query string, file name, file extension, as well as domain URLs (those for which path and query string are empty). Another set of features of interest are related to number of parameters and their values in the query string, as malicious domains tend to have more diverse parameter values. Measurement on our dataset confirms that malicious domains differ greatly in these features compared to other domains. For instance,

https://t.me/learningnets

Category	Features	Description	Type	Enterprise	Novel
		Internal features			
Communication	Num_Hosts	Total number of hosts contacting the domain	Numeric	No	Used in [47]
	Num_Conn	Total number of connections to the domain	Numeric	No	Similar to [52]
	Avg_Conn, Min_Conn, Max_Conn	Avg/Max/Min number of connections per host	Numeric	Yes	Novel
	Total_sent_bytes, Total_recv_bytes	Total number of bytes sent and received	Numeric	No	Used in [11, 12, 63]
	Avg_ratio_rbytes, Min_ratio_rbytes, Max_ratio_rbytes	Avg/Max/Min ratio bytes recv. over sent per host	Numeric	Yes	Novel
	Total_GET, Total_POST	Total number of GET/POST	Numeric	No	Novel
	Avg_ratio_PG, Min_ratio_PG, Max_ratio_PG	Avg/Max/Min ratio of POST over GET per host	Numeric	Yes	Novel
Domain	Dom_Length	Domain name length	Numeric	No	Used in [10, 40]
	Dom_Level	Number of levels	Numeric	No	Used in [10, 40]
	Dom_Sub	Number of sub-domains on SLD	Numeric	No	Novel
	Dom_TLD	Top-level domain	Categorical	No	Used in [10, 40, 61]
URL	Num_URLs	Distinct URLs	Numeric	No	Used in [33]
	Avg_URL_length, Min_URL_length, Max_URL_length	Avg/Max/Min URL path length	Numeric	No	Similar to [11]
	Avg_URL_depth, Min_URL_depth, Max_URL_depth	Avg/Max/Min URL path depth	Numeric	No	Similar to [11]
	Num_params	Total number of parameters across all URLs	Numeric	No	Used in [11]
	Avg_params, Min_params, Max_params	Avg/Max/Min parameters per URL	Numeric	No	Similar to [11]
	Avg_vals, Min_vals, Max_vals	Avg/Max/Min values per parameter	Numeric	No	Similar to [11]
	Frac_URL_filename	Fraction of URLs with file name	Numeric	No	Novel
	Num_filename, Num_exts	Total number of file names and extensions	Numeric	No	Novel
	Frac_query	Fraction of URLs with query string	Numeric	No	Novel
	Frac_frag, Num_frag	Fraction and number of URLs with fragments	Numeric	No	Novel
	Frac_bare	Fraction of domain URLs	Numeric	No	Used in [61]
UA	**Distinct_UAs**	Distinct UAs in all connections to domain	Numeric	No	Novel
	Ratio_UAs	Ratio of distinct UAs over hosts	Numeric	Yes	Novel
	Avg_UAs, Min_UAs, Max_UAs	Avg/Max/Min number of UAs per host	Numeric	Yes	Novel
	Frac_no_UA	Fraction connections with empty UA	Numeric	Yes	Novel
	Frac_UA_1, Frac_UA_10	Fraction unpopular UAs used by 1 and $\leq$ 10 hosts	Numeric	Yes	Used in [47]
	UA_Popularity	Inverse average UA popularity	Numeric	Yes	Novel
	Browser	Dominant browser	Categorical	No	Novel
	Avg_Browsers	Avg number of browsers per host	Numeric	Yes	Similar to [61]
	OS	Dominant OS	Categorical	No	Novel
	Avg_OS	Avg number of OSes per host	Numeric	Yes	Similar to [61]
Result code	**Frac_200, Frac_300, Frac_400, Frac_500**	Fraction 2xx/3xx/4xx/5xx	Numeric	No	Novel
	Num_200, Num_300, Num_400, Num_500	Connections 2xx/3xx/4xx/5xx	Numeric	No	Novel
	Ratio_fail	Ratio failing connections	Numeric	No	Novel
Referer	Frac_no_ref	Fraction connections without referer	Numeric	No	Used in [46, 47]
	Num_ref_doms	Number of distinct referer domains	Numeric	No	Similar to [61]
	Ratio_ref	Ratio of distinct domains over hosts	Numeric	Yes	Novel
	Avg_ref, Min_ref, Max_ref	Avg/Max/Min number of domains per host	Numeric	Yes	Novel
	Has_Referer	Has referer different than itself	Boolean	No	Novel
Content-type	**Distinct_ct**	Number of distinct content-types	Numeric	No	Novel
	Frac_ct_empty	Fraction of empty content-type	Numeric	No	Novel
	Frac_ct_js, Frac_ct_image, Frac_ct_text,	Fraction of content-types per category	Numeric	No	Novel
	Frac_ct_video, Frac_ct_app, Frac_ct_html	(Java script, image, text, video, application)	Numeric	No	Novel
		External features			
WHOIS	Reg_Age	Registration age	Numeric	No	Used in [40, 47]
	Update_Age	Update age	Numeric	No	Used in [40]
	Reg_Validity	Registration and update validity	Numeric	No	Used in [40, 47]
	Update_Validity	Update validity	Numeric	No	Used in [40]
	Reg_Email	Registration email category	Categorical	No	Novel
Hosting Type	**Free_Host, Dynamic_DNS, URL_Shortner**	Free hosting, dynamic DNS, URL shorteners	Binary	No	Novel
Geolocation	Set_ASNs	ASNs of resolved IPs	Categorical	No	Similar to [8]
	Num_ASNs	Number of distinct ASNs	Numeric	No	Similar to [8]
	Set_Countries	Set of countries for resolved IPs	Categorical	No	Used in [40]
	Num_countries	Number of distinct countries	Numeric	No	Similar to [13, 40, 61]

Table 3: List of Internal and External Features. New features not used in previous work are in bold.

malicious domains have on average twice as many parameter values than legitimate ones. Malicious domains are more likely to use domain URLs: 16.15% of all URLs to malicious domains, while only 3.79% URLs to legitimate domains, are domain URLs.

User-agent string features. User-agent (UA) string might be exploited as a channel for command and data transmission from enterprise victim machines [45]. First, we count the distinct UAs seen under the given domain and statistics per host. In addition, we count the fraction of HTTP requests with empty UA. Third, enterprise machines install similar software and we expect most UAs to be seen across a large set of hosts. We build a history of UAs observed over a month-long period and consider a UA as *popular for that enterprise* if it has been used by at least 10 hosts. Then, we compute several enterprise-specific features: the fraction of *unpopular UAs* (used by 1 and less than 10 hosts) and *inverse average of UA popularity* [1]. In our dataset, 15.2% of malicious domains and only 1.5% legitimate domains are contacted solely through unpopular UAs. The UAs also encode the OS and browser, which can be obtained by parsing the UA. It turns out that Windows XP is 5 times more vulnerable than other OSes (Windows 8, Android, and MAC OS), while IE is twice as vulnerable as other browsers (Firefox, Chrome, Opera, and Safari).

Result code features. We divide the result code from HTTP response into four categories (2xx/3xx/4xx/5xx) and compute the fraction and total number of such connections for each category. Since malicious domains tend to be taken down quickly by hosting providers/registrars and have short lifetimes, more failed connections (4xx/5xx) are

[1] Assume a domain has n UAs (UA_1, UA_2, …, UA_n) and the popularity (number of hosts) of UA_i is X_i, the inverse average is computed as $\sum_{i=1}^{n} \frac{1}{X_i}$.

https://t.me/learningnets

usually observed. In addition, the enterprise proxies tend to block access to unknown domains (responding with code 404). Repeated visits to such domains with high number of failed connections are likely attributed to automated software rather than humans. In fact, the percentage of domains with 100% failed connections is 34.1% for malicious domains and only 4.9% for other domains.

Web referer features. User-initiated browser requests have a referer field, while automated processes might use empty referer. We confirmed that in our training set, 24.5% malicious domains were visited without referer, comparing to only 4.3% for other domains. A large number of referer domains or high ratio of referer domains to hosts suggest that the domain might be exploited by malware [48] or used as central point to coordinate compromised sites [37]. Previous work has analyzed characteristics of re-direction chains extracted from the referer field [61], but in MADE we only look at the referer URL for each visit.

Content-type features. Small number of content-types on a domain suggests the domain might not be used to deliver regular web content. Moreover, certain content-types (e.g., exe and jar) have higher associations with malware and exploits. To capture this, we consider the number and fraction of URLs within each category (html, java script, application, image, video, text). We confirmed that legitimate domains have about twice as many content types than malicious ones. Additionally 41.35% malicious domains, and only 8.98% legitimate domains have empty content types.

External Features. We leverage public external resources to enrich our dataset. The derived set of external features are elaborated below:

WHOIS information. WHOIS information is very relevant in predicting malicious activities (e.g., [22, 28, 40]). We issue WHOIS lookups for all the monitored domains and extract registration/update/expiration dates and registrant email for detection. We compute the number of days since registration as *registration age* and the number of days till expiration as *registration validity*. Similarly, we compute the number of days since the latest update as *update age* and the number of days from update till expiration as *update validity*. The age and validity of malicious domains are usually much shorter than those of legitimate ones, and this is confirmed in our data. Figure 2 (left) shows that the mean registration age and validity for malicious domains are 191 and 366 days, comparing to 2404 and 2927 days for legitimate domains. We also examine the registrant email and classify its hosting services into 5 categories: *personal* (if the service is mainly for personal use, e.g., `gmail.com`), *private* if the domain is registered privately, e.g., `domainsbyproxy.com`, *domain* (if the server name equals to domain name), *empty* (if there is no email available), and *other*. Personal and private emails have higher associations with malicious domains.

Hosting type. We retrieve public lists of known free-hosting providers, dynamic DNS and URL shorteners from `malware domains.com` [19] and match them against the monitored domain names. Attackers abuse free-hosting providers, dynamic DNS for domain fluxing [37] and URL shorteners as redirectors [16], and domains using these services are more suspicious.

IP address geolocation. Malware is not uniformly distributed across geographies, with some countries and ASNs hosting more malicious infrastructures [60]. In our dataset, Figure 2 (center) shows the ratio of malicious domains for different countries, demonstrating its heavy-tailed distribution. We resolve the IP addresses associated with monitored domains and map them into ASNs and countries according to Maxmind [1]. We include the ASN and country as categorical features, and also the number of ASNs and countries as numerical features. Intuitively, higher diversity of ASNs and countries might indicate IP fluxing, a popular attack technique [29].

3.3 Feature Selection

Feature	Binary indicators	Selected
TLD	132	37
ASN	3272	207
Country	99	19
OS	15	11
Browser	9	4
Reg email	5	4
Total	3532	282

Table 4: Statistics on categorical features.

Our goal here is to select the most relevant features for our problem. One of the challenges we face is that 6 of the features are categorical, while the majority (83) are numeric. Among the 6 categorical features, ASN has 3,272 values, while country has 99 distinct values. Representing each distinct value with a binary indicator variable results in 3,532 binary features. Inspired by existing methods, we propose a two-step feature ranking procedure:

1. *Ranking categorical features:* We apply logistic regression (LR) with LASSO regularization on the set of binary features created for all categorical features. Regularization encourages sparse solutions in which many coefficients are set to zero. Table 4 shows the number of binary features for our six categorical features, and the number of features selected by LR.

2. *Ranking numerical and binary features:* We selected the 83 numerical features and 282 relevant binary features provided by LR, in total 365 features. For ranking the numerical features, we use the information gain metric. Among 365 features, 106 had an information gain greater than 0 and 42 features had a gain above 0.01. We show the ranking of the top 20 features based on information gain in the right graph of Figure 2. Interestingly, we observe that representative features from most categories (communication structure, UA, URL, content type, result code as well as external features) are ranked in the top 20 features. Domain age is the highest ranked feature and three WHOS features are also highly ranked (this is consistent with previous work [40]).

The top 20 predictors include several enterprise-specific features that depend on the enterprise's traffic profiles (e.g., Avg_Conn, Avg_ratio_rbytes, Max_ratio_rbytes, UA_Popularity, Ratio_UA_hosts). Among the top 20 predictors the ones that have positive correlation with the malicious class (in decreasing order of their correlation coefficients) are: Frac_ct_empty, Frac_400, Avg_URL_length, Num_400, Ratio_fail, and Max_URL_length. This confirms that malicious domains have higher ratio of connections with empty

https://t.me/learningnets

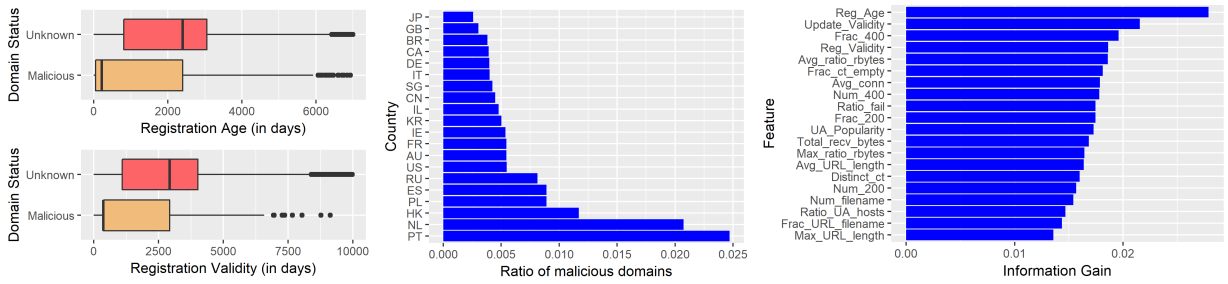

Figure 2: Distribution of WHOIS features (left). Malicious activity by country (center). Ranking of features (right).

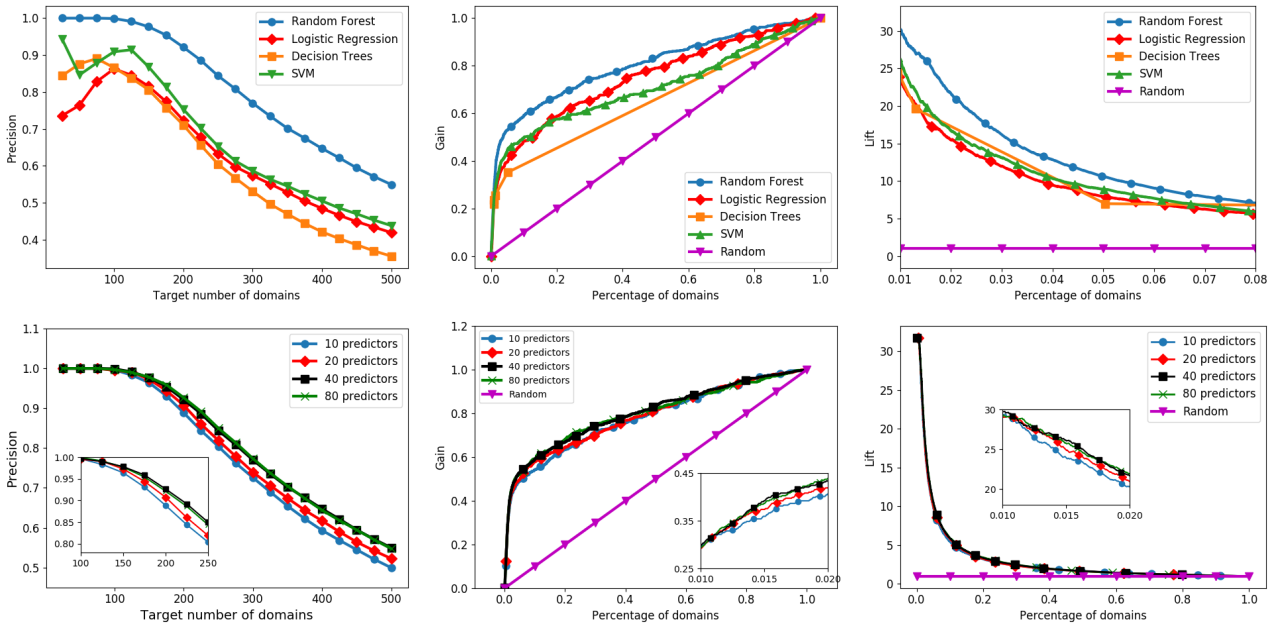

Figure 3: Precision, gain, and lift charts for several supervised learning models (top) and random forest with different number of features (bottom).

content types, more failed connections, and longer URLs than legitimate domains. The predictors correlated with the legitimate class, again in order of their correlation coefficients, are: Frac_200, Frac_URL_filename, Distinct_ct, Reg_Age, Reg_Validity, and Update_Validity. Therefore, legitimate domains have higher number of successful connections, more URLs with file names, serve more content types, and have higher registration age, registration validity, and update validity compared to malicious domains.

3.4 Model Selection

Methodology. The main challenge we encountered is that most domains observed in the enterprise traffic are *unknown*. In our dataset, benign domains represent 6.87% of traffic, and malicious domains about 0.24%, while unknown domains are 92.88%. Unless most previous work that uses classification to distinguish malicious and benign domains, our operational enterprise setting is quite different: we aim to prioritize the most suspicious domains among a large set of unknown domains. A model trained on a small number of benign and malicious domains will not be successful in prioritizing the suspicious domains in the unknown class.

With these insights, we propose here a different approach not yet explored (to the best of our knowledge) in the security community. We first whitelist the benign domains (Alexa top 10K) and then focus on prioritizing the malicious domains among the large set of unknowns. For this task, we adapt the ML framework introduced at the beginning of this section as follows: We create a training dataset $\mathcal{D}_{tr} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ including only malicious and unknown domains, and set numerical labels $y_i = 1$ for malicious domains and $y_i = 0$ for unknown domains. We define $\mathcal{H}$ the hypothesis space of all functions from domain representations X (the set of selected features) to predictions $Y \in [0, 1]$. Intuitively, the higher the prediction value, the more confident we are in the domain's malicious status. We aim to learn a supervised model $f \in \mathcal{H}$ that minimizes a specific loss function. On the testing set $\mathcal{D}_{test} = \{x'_1, \ldots, x'_m\}$, we predict the probability that x'_i is malicious as: $\Pr[x'_i = M] = f(x'_i)$. We leverage several *interpretable supervised models*: logistic regression (LR), decision trees (DT), random forest (RF), and SVM. Our

https://t.me/learningnets

main insight is that by predicting the probabilities that a domain is malicious (rather than simply the label), we can also prioritize the riskier domains based on the predicted probabilities.

We sample 50,000 unknown domains from July and all 1,755 malicious domains observed in February, March, and July to train our models. We believe that this is a large representative set of legitimate connections and under-sampling the majority class is a standard ML technique to handle imbalanced datasets. To perform model selection, we use standard 10-fold cross validation, selecting one third of the training set for validation in each fold. All results are averaged over the 10 folds.

Which metrics? The first question we address is which metrics to use for evaluating different predictive models. Standard metrics for classification (AUC, F1 score) are not sufficient since our main goal is again to prioritize the most suspicious domains. Therefore, we maximize our prediction quality for a small subset of the targeted population. Similar techniques are used in targeted advertising, in which predictions are maximized on the subset of responding population. Inspired from that setting, we use the gain and lift metrics. More specifically, we rank the domains with highest predicted probabilities by model f and define D_p as the fraction p of domains in the validation set with highest predictions $D_p = Top_p[\{x, \Pr[x = M] = f(x)\}]$ (where Top_p is a function that outputs the pv domains with highest probabilities, v being the size of validation set). *Lift* is defined as the ratio between the precision of the model compared to random guessing over the target population in D_p, while *gain* is defined as the recall in the target population D_p (the fraction of all malicious domains included in D_p). We also use *precision in the target population* defined as the true positives rate (malicious domains) in set D_p, and *false positive rate* (FPR) defined as the false positives in D_p divided by the entire set of testing domains. According to the SOC constraints, we set $|D_p|$ to at most 200 domains per month.

Which probabilistic model? Motivated by interpretability considerations, we experiment with four supervised models: LR, DT, RF, and SVM. The top graphs in Figure 3 show the precision, gain, and lift for the four models. Notably, the random forest classifier significantly outperforms other models for the metrics of interest. The random forest precision is 92.15% in the top 200 domains (our budget for one month), but the precision of logistic regression, decision tree, and SVM is only 72.35%, 71.1%, and 75.35% respectively for the same number of domains. The gain of random forest is at 59.2% for 10% of the population, but only 48.34%, 38.52%, and 49.81% respectively, for logistic regression, decision trees, and SVM. The lift metric is also higher for random forest (at 29.7) in 1% of the population compared to 23.4 for logistic regression, 19.64 for decision trees, and 26.1 for SVM. We also experimented with different number of trees in the random forest model (from 50 to 1000) and found that 500 trees is optimal.

How many features? Finally, we are interested in the minimum number of features for optimizing our metrics. We rank the list of 365 features according to information gain as discussed in Section 3.2 and select the top n features for different values of n. We then train random forest models with n features. The bottom graphs in Figure 3 show the precision, lift and gain chart for $n \in \{10, 20, 40, 80\}$ in a random forest model. Precision for the top 200 domains is improved from 88.85% with 10 features to 90.75% with 20 features and 92.15% with 40 features. Gain in 10% of the population is 59.2% (i.e., 59.2% of all malicious domains are included in the highest ranked 10% of the population) for 40 features compared to 54.33% for 10 features, and 57.76% for 20 features. The lift in 1% population (measuring the improvement in precision over random guessing) is 29.34 for 10 features and 29.7 for 40 features. Interestingly, when using more than 40 features the prediction accuracy with respect to all the metrics starts to degrade. This is explained by the fact that features with rank higher than 40 have low information gain (below 0.01).

Finally, the model with best results in terms of our metrics is a random forest model using the top 40 highest ranked features and 500 trees. We trained such a model (called RF-40) on the entire training set and output it at the end of the training phase.

Is MADE efficient? MADE took 14 hours to process and extract the filtered data from the database for the month of July. MADE took an additional 2 hours to generate internal features by performing aggregation of raw events. The process to query external features took in total 13 hours, split into 9 hours to query WHOIS, 3 hours to query geolocation information, and one hour to extract features. After all the features are extracted, training the RF model takes on average 5 minutes. We thus believe that MADE has reasonable performance.

4 TESTING AND EVALUATION

In this section, we elaborate our approach to evaluate the effectiveness of the RF-40 model on new testing data. The testing process consists of the following steps: *Data Representation*, *Model Prediction*, and *Ranking High-Risk Communications*.

4.1 MADE Testing

Data Representation. For our testing data, we sample a set of 50,000 unknown domains from August, and include all the malicious domains (516). Thus, the testing data is similar in size to our training set. We extract the 40 selected features from Section 3 and create the data representation that can be used in the RF-40 model.

Model Prediction. The random forest model RF-40 is applied to the new testing data. For each domain x visited in the testing interval, the model computes $\Pr[x = M] = f(x)$, the probability that the domain is malicious. We call these predictions *domain risk scores*. We plot the CDF of domain risk scores for malicious and unknown domains in Figure 4 (left), which clearly demonstrates that domains labeled as malicious have higher scores than unknown domains.

Ranking High-Risk Communications. We rank the domains in the testing set according to the predicted risk scores, under the observation that domains with higher risk scores are more likely malicious. We generate a list of 1,000 domains with highest risk scores predicted by the model, and we investigate the top 100 together with the SOC.

4.2 Evaluation, Analysis, and Feedback

Validation process. We submit all 1,000 domains to VirusTotal and use additional threat intelligence services (e.g., Cisco Talos) for high-confidence detections. For the remaining domains ranked in top 100, we perform manual investigation in collaboration with SOC analysts. We issue HTTP requests to crawl the domain home page

https://t.me/learningnets

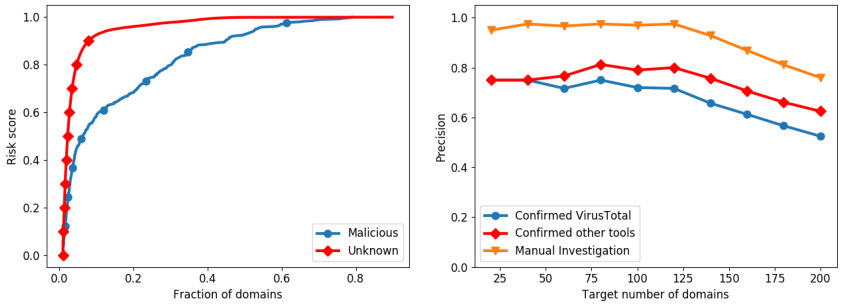

Figure 4: Predicted CDF scores for malicious and unknown domains in the testing set (left). Final precision results (right).

and check if malicious payload is embedded. We also search for public reports on the domain and for other malicious activities on the end host using various SOC tools.

Evaluation results. In Figure 4 (right) we illustrate the precision of RF-40 in the top 200 domains, including 3 lines: (1) *Confirmed VirusTotal* (confirmed by VirusTotal); (2) *Confirmed other tools* (confirmed by VirusTotal and other threat intelligence services); (3) *Manual investigation* (All confirmed by existing services and labeled malicious by manual investigation). Interestingly, in the top 100 domains, 72 were marked as malicious by VirusTotal, 7 other were detected by other tools, and 18 were confirmed by manual investigation. Overall, the MADE precision has reached an impressive 97% in the top 100 domains, with only 3 false positives in 50,000 testing domains (corresponding to FPR of $6 \cdot 10^{-5}$). In the top 200 domains, the MADE precision is 76% with FPR $4.8 \cdot 10^{-4}$.

These results show that MADE is capable of detecting malicious domains with high accuracy. Moreover, the prioritization mechanism in MADE based on the risk score generated by the ML model is quite effective. The precision of MADE in the top 100 domains is 97%, decreasing to 89.86% in the top 150 domains, and 76% in the top 200 domains. Therefore, the domains with highest rank are more likely to be malicious. As another interesting finding, MADE can also detect new malicious domains that remain undetected by VirusTotal and other threat intelligence services (MADE detected a set of 18 such domains in the top 100 prioritized domains). As shown in Table 1, MADE achieves better precision than existing systems at similar false positive rates, while detecting more general classes of enterprise malware communication.

Case study. We describe a malware campaign discovered by manually investigating the domains of highest score. The adversary registered 5 domains (`keybufferbox.com`, `globalnodemax.com`, `maxdevzone.com`, `gencloudex.com` and `bitkeymap.com`) and created 8 or 9 subdomains under each. In total, 645 enterprise hosts visited at least one such domain within one-month period.

We show details about the infrastructure and operations of this campaign in Figure 5. The malware is delivered to the victim's machine when she visits subdomains under prefix `dl.*` and `download.*`. After the extension is installed, it requests additional scripts from subdomains `notif.*`, `js.*` and `app.*`. The profile of the victim is transmitted to `logs.*`, and the victim's status and communication errors are sent to `logs.*` and `errors.*`. The

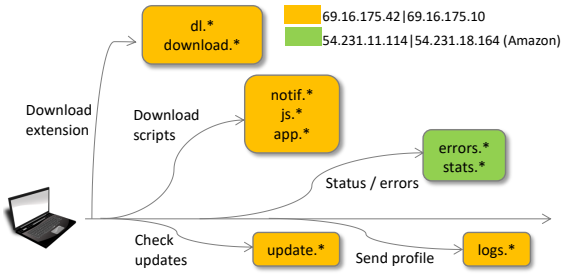

Figure 5: Infrastructure of the malware campaign. $\star$ can be any of the five SLD names.

malware frequently checks `update.*` for updating itself. All of the domains are hosted on two IPs owned by Highwinds network.

The names of these domains consist of combinations of three unrelated words. According to SOC, this is a new trend in DGA malware called "dictionary DGA". Rather than using randomly generated domain names with high entropy, this new DGA algorithm picks unrelated dictionary words. This technique specifically evades lexical features used for DGA detection [10, 40]. Interestingly, MADE detects these domains since it does not rely on lexical features.

Our algorithm detects 26 campaign domains, due to the following distinctive features: (1) Infected machines connect frequently to one of `update.*` control domains; (2) Domains are recent in the enterprise traffic; (3) The referer of all requests is either empty or the domain name itself; (4) A large number of URLs (418) are served under each domain, and a high number of parameters (7) and parameter values (72) are embedded in order to send the status of the infected machines to the control servers. In contrast, legitimate domains have an average of 50 URLs, 2 parameters, and 3 values.

Operational deployment. MADE was used in production at a large organization for more than a year and generated prioritized alerts for the SOC daily. In operation, MADE is re-trained every month with new labeled data to account for evolution of malicious activities. MADE proved extremely useful to the SOC by producing high-quality alerts and detecting new malicious activities. A version of MADE is currently deployed in a security product and has successfully detected malware in other enterprises. The product version operates in streaming mode and detects malware communication close to real-time (as soon as 5 connections to an external destination are made, MADE generates a risk score). We believe that

https://t.me/learningnets

the carefully defined features based on feedback from SOC domain experts, as well as our methodology for ranking and prioritizing most suspicious domains are important factors in MADE's success.

4.3 Discussion and Limitations

MADE is designed to detect malicious communication initiated by enterprise machines using machine learning techniques. Below we discuss several cases in which MADE has limited detection ability, as well as potential adversarial evasion attacks against MADE.

Limitations in detection. MADE is effective in detecting malicious communication occurring with certain minimum frequency. Currently, MADE uses a threshold of 5 connections to the same domain in a two-week interval in order to extract features for that particular domain. For extremely stealthy attacks such as Advanced Persistent Threats (APTs), it is feasible for attackers to carefully craft their malicious connections to remain under the radar.

MADE monitors web connections initiated from within the enterprise network. If malware communicates with C&C domains outside the enterprise network, that activity will not be recorded by the web proxies. MADE applies various enterprise-specific whitelists and considers for analysis only unpopular and recently visited domains. MADE is therefore not designed to detect C&C communication to well-established domains, such as cloud services and CDN domains. At the same time, MADE also filters connections to reputable advertisement networks. Motivated attackers could in theory compromise a well-established domain or a reputable ad network and use it for C&C communication. Designing defenses against these threats remains a challenging tasks because of the difficulty of distinguishing malicious and benign connections directed to highly-reputable web sites.

MADE extracts the majority of features from HTTP logs collected from enterprise web proxy servers. In the enterprise of our study, web proxies intercept and act as man-in-the-middle for the majority of HTTPS connections. In general though, MADE will have access to a much smaller set of features for malicious communication over HTTPS (for instance, the URL, user-agent, referer, and content type features will not be available). We have not yet investigated the effectiveness of MADE on detecting HTTPS malicious communication with limited set of features, but this is an interesting topic for future work.

Adversarial evasion. Adversarial attacks against supervised learning models have been highly effective in domains such as image classification [14], face recognition [56], and cyber security [25]. We believe that some of these attack strategies could be adapted to work against the machine learning models employed by MADE. Attackers could manipulate some of the high-importance features used by MADE to modify the results of classification. For instance, the highest-rank feature in MADE is domain age, and therefore attackers could register a domain in advance before using it for C&C communication. However, this will incur some monetary costs to attackers. It is also relatively straightforward for attackers to modify communication profiles to malicious domains (such as bytes sent and received, their ratio, user-agent strings, and URLs). Still, designing an optimal evasion strategy in this setting is currently an open problem, particularly in the *black-box* attack model when attackers do not have access to the training set and details of the ML algorithm. We also conjecture that enterprise-specific features (such as UA popularity) are harder to evade as attackers need additional information about enterprise legitimate communications to design their evasive attack samples. An additional challenge from attacker's perspective is that MADE uses random forests (ensemble of hundreds of trees), currently believed to be more difficult to evade than linear models (e.g., logistic regression or SVM).

5 RELATED WORK

Our work aims to detect suspicious HTTP communications in an enterprise setting through machine learning analysis. There is a large body of related literature in detecting malicious domains related to spam, command-and-control activities or malware delivery, as well as applying machine learning models to security datasets.

Detecting malicious domains. Ma et al. [40] evaluate a large number of features, including WHOIS, geographical, and URL lexical features for detecting spam URLs. Zhao and Hoi [67] propose an active learning framework for URL classification to handle imbalanced training datasets. Kruegel and Vigna [36] identify anomalies in URL structure to detect web attacks. Soska and Cristin [58] design a classifier for predicting the compromise of a website based on web page structure and traffic statistics.

Several systems, e.g., Notos [8] and EXPOSURE [13], build generic domain reputation systems by applying classification algorithms on passive DNS data. Kopis [9] analyzes DNS data collected at the upper level of the DNS hierarchy. Felegyhazi et al. [22] proactively identify malicious domains by mining DNS zone files and WHOIS registration information. Antonakakis et al. [10] build a detector for DGA domains using a combination of lexical, entropy and structural features extracted from DNS traffic collected from an ISP. Segugio [51] propagates reputation scores from benign or compromised machines to visited domains in the DNS query graph. Comparing to HTTP logs, DNS logs include much less information about external destinations visited by enterprise machines, are smaller in size, and have lower risk of revealing user private information. Therefore, DNS detection systems have an advantage when storage and privacy (as mandated by recent European regulations) are major concerns. On the downside, fewer features can be extracted from DNS logs (as for example, the URL in HTTP connections is not available). DNS logs are thus amenable for detecting certain classes of malware (for instance, DGA or fast-flux), but are limited in their ability to detect broader malicious communication.

Other data sources have been used for identifying malicious domains. DISCLOSURE [12] and BotFinder [63] build models to detect C&C traffic using features extracted from NetFlow records. BotMiner [26] applies clustering to features extracted from network flows for botnet detection. Nazca [33] detects malware delivery networks through graph analysis of web requests from ISP networks. PREDATOR [28] designs a system for predicting domain reputation at registration time. Shady Path [61] detects malicious web pages by analyzing how a large set of browsers interact with web pages and extracting characteristics from the redirection graphs generated to these web pages. CAMP [52] leverages features collected from users' browsers during the file downloading process (e.g., the final download URL and IP address) to identify malware hosted by websites.

https://t.me/learningnets

Enterprise log analysis. In addition to the systems surveyed in Table 1, we mention several enterprise log analysis systems. Beehive [65] applies anomaly detection to identify suspicious enterprise hosts. This is orthogonal to detecting malicious communication patterns. Several papers [15, 41] use the idea of propagating trust in the communication graph for detecting malicious domains. WebWitness [46] proposes a forensics method to determine malware download paths after a malicious download event is detected.

Industry solutions. Applying machine learning to detect malicious activities and reduce the workload of SOC has become popular in cyber-security industry in recent years [49, 53]. Machine-learning models have been applied in security applications such as automated endpoint analysis (e.g., detecting malware based endpoint system traces) [17, 23, 62], cloud instance monitoring (e.g., detect anomalous account access) [7, 43], user behavioral analysis (e.g., identifying users with high risk scores) [31, 54, 59], network communication analysis (e.g., detecting malicious domains) [20, 55], security orchestration (e.g., assigning alert tickets to security analysts) [18], and event triaging from data collected by SIEM [32, 39, 64]. MADE focuses on prioritizing alerts related to enterprise malicious web communications and can detect a range of malicious activities.

6 CONCLUSION

We describe the MADE system for detecting malicious HTTP communication in enterprises by web proxy log analysis. MADE is built in collaboration with SOC tier 3 analysts at a large enterprise and leverages an extensive set of enterprise-specific and generic features for capturing malicious behavior. The goal of MADE is to assign risk scores to external destinations contacted by enterprise hosts and prioritize the most suspicious ones. MADE is able to achieve 97% precision in the set of 100 highest-risk domains detected over a month at only $6 \cdot 10^{-5}$ FPR. MADE was successfully used in production and discovered new malicious domains (not identified by several state-of-the-art security technologies). Avenues for future work include adversarial analysis of MADE (which features and ML models are more resilient against advanced attackers), expanding the set of malicious activities, and combining network with host data for more comprehensive view into malicious campaigns.

ACKNOWLEDGEMENTS

We would like to thank the enterprise that gave us access to the web proxy logs for analysis. We are grateful to the entire EMC Critical Incident Response Center (CIRC) for their support over several years when this work was performed. We would like to thank Todd Leetham and Christopher Harrington for their suggestions in designing MADE, insightful discussions on latest attacker trends, and help with evaluating our findings. We also thank the RSA Data Science team and RSA Engineering for their work on transitioning MADE to production. Finally, we thank our shepherd Gianluca Stringhini and the anonymous reviewers for their feedback on the paper.

REFERENCES

[1] MaxMind. http://www.maxmind.com/.
[2] VirusTotal. http://www.virustotal.com/.
[3] Verizon 2018 data breach investigations report. https://www.verizonenterprise.com/verizon-insights-lab/dbir/, 2018.
[4] Adblock Plus. EasyList. https://easylist-downloads.adblockplus.org/easylist.txt, 2015.
[5] Adblock Plus. EasyPrivacy. https://easylist-downloads.adblockplus.org/easyprivacy.txt, 2015.
[6] Alexa. AWS | Alexa Top Sites - Up-to-date lists of the top sites on the web. http://aws.amazon.com/alexa-top-sites/, 2014.
[7] Amazon. GuardDuty Intelligent Threat Detection AWS. https://aws.amazon.com/guardduty/, 2018.
[8] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. Building a dynamic reputation system for DNS. In *Proc. 19th USENIX Security Symposium*, 2010.
[9] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, II, and David Dagon. Detecting malware domains at the upper DNS hierarchy. In *Proc. 20th USENIX Security Symposium*, 2011.
[10] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. From throw-away traffic to bots: Detecting the rise of DGA-based malware. In *Proc. 21st USENIX Security Symposium*, 2012.
[11] Karel Bartos, Michal Sofka, and Vojtech Franc. Optimized invariant representation of network traffic for detecting unseen malware variants. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 807–822. USENIX Association, 2016.
[12] Leyla Bilge, Davide Balzarotti, William Robertson, Engin Kirda, and Christopher Kruegel. DISCLOSURE: Detecting botnet Command-and-Control servers through large-scale NetFlow analysis. In *Proc. 28th Annual Computer Security Applications Conference (ACSAC)*, ACSAC, 2012.
[13] Leyla Bilge, Engin Kirda, Kruegel Christopher, and Marco Balduzzi. EXPOSURE: Finding malicious domains using passive DNS analysis. In *Proc. 18th Symposium on Network and Distributed System Security*, NDSS, 2011.
[14] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE Computer Society, 2017.
[15] Kevin M.. Carter, Nwokedi Idika, and William. W. Streilein. Probabilistic threat propagation for network security. *IEEE Transactions on Information Forensics and Security*, 9, 2014.
[16] Sidharth Chhabra, Anupama Aggarwal, Fabricio Benevenuto, and Ponnurangam Kumaraguru. Phi. sh/$ ocial: the phishing landscape through short urls. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, pages 92–101. ACM, 2011.
[17] CrowdStrike. CrowdStrike Introduces Enhanced Endpoint Machine Learning Capabilities and Advanced Endpoint Protection Modules. https://goo.gl/wVh3s9, 2017.
[18] Demisto. Top Machine Learning Use Cases âĂŞ Part 1. https://blog.demisto.com/demistos-top-machine-learning-use-cases-part-1, 2018.
[19] DNS-BH. Malware Domain Blocklist. http://mirror1.malwaredomains.com/files/, 2015.
[20] Endgame. Using Deep Learning To Detect DGAs. https://www.endgame.com/blog/technical-blog/using-deep-learning-detect-dgas, 2016.
[21] Brown Farinholt, Mohammad Rezaeirad, Paul Pearce, Hitesh Dharmdasani, Haikuo Yin, Stevens Le Blond, Damon McCoy, and Kirill Levchenko. To catch a Ratter: Monitoring the behavior of amateur DarkComet RAT operators in the wild. In *IEEE Symposium on Security and Privacy*, pages 770–787. IEEE Computer Society, 2017.
[22] Mark Felegyhazi, Christian Keibich, and Vern Paxson. On the potential of proactive domain blacklisting. In *Proc. Third USENIX LEET Workshop*, 2010.
[23] FireEye. Reverse Engineering the Analyst: Building Machine Learning Models for the SOC. https://www.fireeye.com/blog/threat-research/2018/06/build-machine-learning-models-for-the-soc.html, 2018.
[24] WPO Foundation. CDN list. https://raw.githubusercontent.com/WPO-Foundation/webpagetest/master/agent/wpthook/cdn.h, 2015.
[25] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michae l Backes, and Patrick D. McDaniel. Adversarial examples for malware detection. In *ESORICS (2)*, volume 10493 of *LNCS*, pages 62–79. Springer, 2017.
[26] Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee. BotMiner: Clustering analysis of network traffic for protocol and structure-independent botnet detection. In *Proc. 17th USENIX Security Symposium*, 2008.
[27] Tristan Halvorson, F. Matthew Der, Ian Foster, Stefan Savage, K. Lawrence Saul, and M.Geoffrey Voelker. From .academy to .zone: An analysis of the new tld land rush. In *15th Internet Measurement Conference (IMC)*, 2015.
[28] Shuang Hao, Alex Kantchelian, Brad Miller, Vern Paxson, and Nick Feamster. PREDATOR: Proactive recognition and elimination of domain abuse at time-of-registration. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1568–1579. ACM, 2016.

https://t.me/learningnets

[29] Thorsten Holz, Christian Gorecki, Konrad Rieck, and Felix C Freiling. Measuring and detecting fast-flux service networks. In *NDSS*, 2008.
[30] Xin Hu, Jiyong Jang, Marc Ph. Stoecklin, Ting Wang, Douglas Lee Schales, Dhilung Kirat, and Josyula R. Rao. BAYWATCH: robust beaconing detection to identify infected hosts in large-scale enterprise networks. In *DSN*, pages 479–490. IEEE Computer Society, 2016.
[31] IBM. Machine Learning Analytics app. https://goo.gl/DCFCBN, 2016.
[32] IBM. Artificial Intelligence for Smarter Cybersecurity. https://www.ibm.com/security/artificial-intelligence, 2018.
[33] Luca Invernizzi, Stanislav Miskovic, Ruben Torres, Sabyaschi Saha, Sung-Ju Lee, Christopher Kruegel, and Giovanni Vigna. Nazca: Detecting malware distribution in large-scale networks. In *Proc. ISOC Network and Distributed System Security Symposium (NDSS '14)*, 2014.
[34] Emi Kalita. *WannaCry Ransomware Attack: Protect Yourself from WannaCry Ransomware Cyber Risk and Cyber War*. Independently published, 2017.
[35] Platon Kotzias, Leyla Bilge, and Juan Caballero. Measuring PUP prevalence and PUP distribution through pay-per-install services. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 739–756, Austin, TX, 2016. USENIX Association.
[36] Christopher Kruegel and Giovanni Vigna. Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, CCS '03, pages 251–261, New York, NY, USA, 2003. ACM.
[37] Zhou Li, Sumayah Alrwais, Yinglian Xie, Fang Yu, and XiaoFeng Wang. Finding the linchpins of the dark web: A study on topologically dedicated hosts on malicious web infrastructures. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 112–126. IEEE Computer Society, 2013.
[38] Jinjin Liang, Jian Jiang, Haixin Duan, Kang Li, Tao Wan, and Jianping Wu. When https meets cdn: A case of authentication in delegated service. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 67–82. IEEE, 2014.
[39] LogRhythm. Big Data Analytics. https://logrhythm.com/solutions/security/security-analytics/, 2018.
[40] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. In *Proc. 15th ACM International Conference on Knowledge Discovery and Data Mining*, KDD, 2009.
[41] Pratyusa K. Manadhata, Sandeep Yadav, Prasad Rao, and William Horne. Detecting malicious domains via graph inference. In *19th European Symposium on Research in Computer Security (ESORICS)*, 2014.
[42] MANDIANT. APT1: Exposing one of China's cyber espionage units. Report available from www.mandiant.com, 2013.
[43] Microsoft. Machine Learning in Azure Security Center. https://azure.microsoft.com/en-us/blog/machine-learning-in-azure-security-center/, 2016.
[44] Shai Morag. Best practices for the SOC team âĂŞ where to automate, where to think. https://www.infosecurity-magazine.com/opinions/best-practices-for-the-soc-team, 2016.
[45] Terry Nelms, Roberto Perdisci, and Mustaque Ahamad. ExecScent: Mining for new C&C domains in live networks with adaptive control protocol templates. In *Proc. 22nd USENIX Security Symposium*, 2013.
[46] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad. WebWitness: Investigating, categorizing, and mitigating malware download paths. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1025–1040, Washington, D.C., 2015. USENIX Association.
[47] Alina Oprea, Zhou Li, Ting-Fang Yen, Sang Chin, and Sumayah Alrwais. Detection of early-stage enterprise infection by mining large-scale log data. In *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2015.
[48] Optimize Smart. Geek guide to removing referrer spam in Google Analytics. http://www.optimizesmart.com/geek-guide-removing-referrer-spam-google-analytics, 2015.
[49] Paul Dwyer. The Security Operations Center Is Evolving Into a Risk Analytics Center. https://securityintelligence.com/the-security-operations-center-is-evolving-into-a-risk-analytics-center/, 2018.
[50] Roberto Perdisci, Wenke Lee, and Nick Feamster. Behavioral clustering of HTTP-based malware and signature generation using malicious network traces. In *Proc. 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, 2010.
[51] Babak Rahbarini, Roberto Perdisci, and Manos Antonakakis. Segugio: Efficient behavior-based tracking of malware-control domains in large isp networks. In *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2015.
[52] Moheeb Abu Rajab, Lucas Ballard, Noe Lutz, Panayiotis Mavrommatis, and Niels Provos. CAMP: content-agnostic malware protection. In *Proc. ISOC Network and Distributed System Security Symposium (NDSS '13)*, 2013.
[53] Robert Lemos. AI is changing SecOps: What security analysts need to know. https://techbeacon.com/ai-changing-secops-what-security-analysts-need-know, 2018.
[54] RSA. NetWitness UEBA. https://www.rsa.com/en-us/products/threat-detection-response/ueba, 2018.
[55] RSA. Threat Detection and Response NetWitness Platform. https://www.rsa.com/en-us/products/threat-detection-response, 2018.
[56] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM, 2016.
[57] Robin Sommer and Vern Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *Proc. IEEE Symposium on Security and Privacy*, SP '10. IEEE Computer Society, 2010.
[58] Kyle Soska and Nicolas Christin. Automatically detecting vulnerable websites before they turn malicious. In *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014.*, pages 625–640, 2014.
[59] Splunk. SIEM - Security Information and Event Management. https://goo.gl/Ljtc6t, 2018.
[60] Brett Stone-Gross, Christopher Kruegel, Kevin Almeroth, Andreas Moser, and Engin Kirda. Fire: Finding rogue networks. In *Computer Security Applications Conference, 2009. ACSAC'09. Annual*, pages 231–240. IEEE, 2009.
[61] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Shady Paths: Leveraging surfing crowds to detect malicious web pages. In *Proc. 20th ACM Conference on Computer and Communications Security*, CCS, 2013.
[62] Symantec. How does Symantec Endpoint Protection use advanced machine learning? https://support.symantec.com/en_US/article.HOWTO125816.html, 2018.
[63] Florian Tegeler, Xiaoming Fu, Giovanni Vigna, and Christopher Kruegel. BotFinder: Finding bots in network traffic without deep packet inspection. In *Proc. 8th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '12, 2012.
[64] VECTRA. Cognito Detect is the most powerful way to find and stop cyberattackers in real time. https://vectra.ai/assets/cognito-detect-overview.pdf, 2018.
[65] Ting-Fang Yen, Alina Oprea, Kaan Onarlioglu, Todd Leetham, William Robertson, Ari Juels, and Engin Kirda. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In *Proc. 29th Annual Computer Security Applications Conference*, ACSAC '13, 2013.
[66] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. The dark alleys of madison avenue: Understanding malicious advertisements. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 373–380, New York, NY, USA, 2014. ACM.
[67] Peilin Zhao and Steven C.H. Hoi. Cost-sensitive online active learning with application to malicious url detection. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 919–927, New York, NY, USA, 2013. ACM.

https://t.me/learningnets