Hawk-Eye: Holistic Detection of APT Command and Control Domains

Almuthanna Alageel
Department of Computing, Imperial College London
National Center for Cybersecurity, KACST, SA
a.alageel18@imperial.ac.uk

Sergio Maffeis
Department of Computing, Imperial College London
maffeis@imperial.ac.uk

ABSTRACT

The high complexity and low volume of APT attacks has lead to limited insight into their behavior and to a scarcity of data, hindering research on effective detection techniques. In this paper we present a comprehensive study of the usage of domains in the context of the Command and Control (C&C) infrastructure of APTs, covering 63 APT campaigns spanning the last 13 years. We discuss the APT threat model, focusing in particular on evasion techniques, and collect an extensive dataset for studying APT C&C domains.

Based on the gained insight, we propose a number of novel features to detect APTs, leveraging both semantic properties of the domains themselves and structural properties of their DNS infrastructure. We build Hawk-Eye, a system to classify domain names extracted from PCAP files, and use it to evaluate the performance of the various features we studied, and compare them to malicious domain detection features from the literature. We find that a holistic approach combining selected orthogonal features achieves the best performance, with an F1-score of 98.53% and a FPR of 0.35%.

CCS CONCEPTS

• **Computer Security** → **Network System Security**; *Advanced Persistent Threats*; Command and Control Domains; Evasion Techniques; • **Machine Learning** → Random Forest; Cybersecurity Feature engineering;

KEYWORDS

Advanced Persistence Threats, Command and Control, Malicious Domains, Intrusion Detection

ACM Reference Format:
Almuthanna Alageel and Sergio Maffeis. 2021. Hawk-Eye: Holistic Detection of APT Command and Control Domains. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3412841.3442040

1 INTRODUCTION

Advanced Persistent Threats (APTs) are malicious actors which target specific organizations in order to carry out complex activities such as espionage, data exfiltration or damaging critical infrastructure. Published estimates put the cost of APT activity worldwide at USD 1 trillion [35].

Detecting APT campaigns is more challenging than detecting regular malware campaigns. APT operators tend to be well-trained in cybersecurity, coming from military and governmental organizations, academia and R&D entities [20]. APTs can invest significant resources per victim, and launch targeted and low-volume attacks which can extend over several years. Evasion techniques which would be unfeasible or prohibitively expensive for high-volume or less stealthy attacks, such as botnets or phishing campaigns, are within the arsenal of APTs. Detecting such attacks using traffic analysis may exhaust storage and memory resources.

Data on APT campaigns is sparse, and the security industry is the primary source of information [22]. Targeted organizations request forensics services from security firms, who sometimes publish (part of) their confidential findings after obtaining the client's permission. In this paper we present the most comprehensive study to date of the usage of domains in the context of APT Command and Control (C&C) infrastructure. We cover 63 APT campaigns spanning the last 13 years until August 2020, by reviewing 125 public reports and 146 threat intelligence pulses by 35 leading security organizations. Based on that, we propose a detailed threat model for APT C&C usage, which focuses in particular on evasion techniques (Section 2).

We leverage the gained insight to study the effectiveness of existing and novel features of domain names, and their DNS infrastructure, for detecting APT C&C domains. Malicious domains has been studied extensively in the case of botnets or phishing campaigns. Botnets often use Domain Generation Algorithms (DGAs) to protect their C&C channels from disruption. DGAs generate a large number of pseudo-random variations of domain names using algorithms based on arithmetic, hash functions, wordlists, and permutations [26]. These make it hard to predict what domain names will be used, and even when the algorithm is successfully reverse-engineered, take-downs are expensive as only a small fraction of the large number of generated domains is effectively registered by the botnet operator. Character level features [38] and Non-Existent Domain (NXDomain) responses [10, 28, 33] were shown to be effective in detecting DGA domains. Some APTs use domains that look similar to DGA-generated domains, so character level features are relevant to their detection as well. On the other hand, APTs prioritize stealth and are unlikely to generate large number of NXDomain responses, making the corresponding detection technique not relevant. Overall we have not found typical DGA domains in our analysis of APT campaigns in Section 2. Phishing campaigns can be detected by their choice of domain names, or by analyzing the look, structure and behavior of phishing web pages themselves

SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8104-8/21/03…$15.00
https://doi.org/10.1145/3412841.3442040

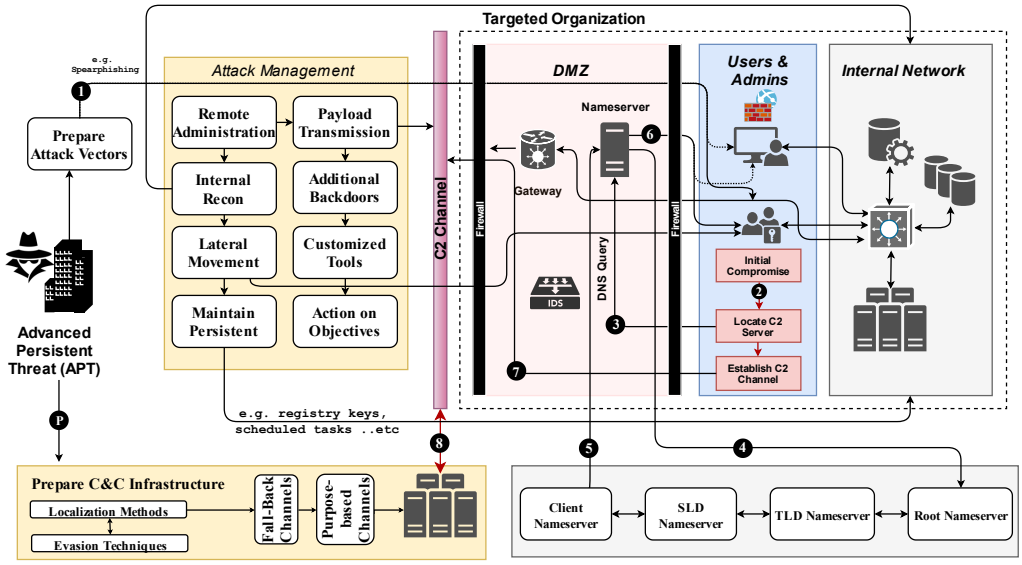

Figure 1: APT Threat Model. Prior to infection, APTs prepare the C&C infrastructure and the initial compromise vectors. Once exploitation begins, APTs move to the attack management phase, which includes C&C localization by infected hosts.

[36, 37]. We note in Section 2 that APT domains may serve innocent pages by default, preventing the latter kind of analysis. Phishers who choose deceptive domain names do so with the goal of confusing an end-user into thinking that the deceptive domain is in fact the legitimate target. APTs who adopt deceptive domains for their C&C communications instead do so in order to avoid automated and manual detection. Yet, phishers and APTs adopt some similar techniques, and we shall leverage lexical-based phishing domain detection techniques to build APT-specific detectors. Some of the features we select, such as the length of a domain, or its decomposition into meaningful parts are *semantic*, in that they reflect the choice of a particular domain name. Other features are *contextual*, in that they express the organization of the DNS infrastructure that supports the C&C operation for the chosen domain. We also collect *hybrid* features, which extract semantic properties from the DNS infrastructure (Sections 4 and 5).

In order to evaluate the performance of the various feature sets we studied, we built Hawk-Eye, a system to classify domain names requested in PCAP files. Hawk-Eye includes modules for robustly parsing and retrieving DNS data, which is sometimes technically challenging, and for extracting features and classifying domains. As a basis for objective evaluation, we collect and make publicly available [7] an extensive dataset of APT C&C domains, based on the campaigns we reviewed (Section 3).

Finally, we compare and discuss the detection performance of different sets of features. We find that a holistic feature set including largely orthogonal features performs the best, achieving accuracy of 98.51% and F1-score of 98.53%. Although semantic features have a strong intuitive appeal and contribute to detection, contextual and hybrid features are in fact the most effective, hinting that the DNS infrastructure may be the weakest link for APT campaigns (Section 6). In summary, our main contributions are:

- A detailed, evidence-based analysis of the use of domains in the context of APT C&C infrastructure.
- The implementation of Hawk-Eye, a classifier to detect APT C&C domain requests from PCAP files, crawling live DNS and WHOIS data where necessary.
- The first publicly available, curated dataset of APT domains used for C&C infrastructure.
- New features for malicious domain detection, targeted for APTs.
- A thorough evaluation of the classification performance of new and existing features under different scenarios.

2 APT CAMPAIGN ANALYSIS

Understanding how APTs behave, from the first day of a campaign until mission accomplished, is crucial in order to identify the attack surface at each stage and helps selecting appropriate detective- and protective-controls. The Lockheed Martin Cyber Kill Chain® could be considered as the first model that was able to describe Advanced Persistent Threats (APT) proposed in 2011 [18]. That was followed by the Mandiant [20] and MITRE ATT&CK [32] frameworks, both increasingly accepted among the security community. In Figure 1 we propose a more specific APT model, focusing on the localization and establishment of the C&C channel, which we use to inform the design of Hawk-Eye. The threat model describes the typical lifecycle for most APTs from the network perspective, and in particular how an APT smoothly penetrates these defenses and manages to hide their activities over time through stealthy C&C channels. The attacker activities are divided into three main phases.

2.1 Prepare C&C Infrastructure

APTs tend to prepare their infrastructure over multiple years. They normally establish a sophisticated network for fall-back channels

support, carefully selecting the locations of their C&C infrastructure and the localization method (Figure 1, ❾). These decisions are based on the properties of their targets. For example, if the target is a government entity in a specific country, C&C servers are likely to be located in the same country, and to use domain names that mimic government-like domains. Using domain names that blend in with the business of the target, or that could plausibly belong to its technical infrastructure is a common technique to evade IDS detection (which needs to avoid false positives), and to mislead SOC engineers into overlooking innocent-looking alerts. For instance, DarkHydrus [14] used `symanteclive.download` (with nameserver `ns102.kaspersky.host`) to localize its C&C server. It also registered `owa365.bid` to imitate Outlook Web Access (OWA), `kaspersky.science`, `fortiweb.download` to masquerade as Kaspersky and Fortinet products, `data-microsoft.services`, `Akamai.agency` and `windowsdefender.win` to blend in with network and cloud infrastructure. Other APT campaigns also use this technique, with CobaltGroup using `api.outlook.kz`, APT 41 using `macfee.ga` and `kasparsky.net`, and APT39 using `win7-update.com`. As a further measure to avoid C&C detection, APTs tend to reserve C&C domain names exclusively for that purpose, and set them up to display clean pages as their default content.

2.2 Prepare Attack Vectors

This stage (Figure 1, ❶) includes selecting the targeted organization, information gathering, customizing malware and tools, identifying vulnerabilities and delivering a malicious payload to the targeted host, using techniques such as spear phishing, whaling or strategic web compromise (SWC). We consider domains used exclusively as part of this phase as out of scope of our investigation.

2.3 Attack Management

At this stage APTs have partial control of the target and can perform malicious activities, such as *remote administration* using SSH or another tunneled protocol. Meanwhile, they can also *transfer more payloads*, over different time windows, binary files, PowerShell scripts, RATs and post-exploitation tools. At the same time, the campaign is sneaking deeper into the network through *internal reconnaissance* and *lateral movement*, potentially using superuser accounts which appear legitimate to the targeted network.

These actions are controlled via the C&C channel. Typically, once the target has been compromised (Figure 1, ❷), malware locates the C&C through the domain that is hard-coded in the configuration block of its main binary (❸). The victim is made to issue a DNS query for that domain to the local nameserver. This request is initiated by operating system processes, without the need of evading a web application firewall. If the requested record is not cached, (❹) the local nameserver communicates with root, TLD, SLD and enterprise-level nameservers to obtain them (❺, ❻). The contacted domain may display legitimate web pages, or not even display any content. In the context of APTs, the `A` record normally points to a C&C server, and a typical APT campaign changes the `A` record periodically to avoid frequent connections to the same IP (❼). For example, the APT C&C domains `windowsdefender.win` and `micrrosoft.net` changed their `A` records every 24 and 48 hours during June - July 2018 and August - September 2019, using nameservers `ns102.kaspersky.host` and `ns11025.ztomy.com`, providing each time a fresh IP. Strider APT aggressively changed the `A` records for `bikessport.com` to 203 distinct IPs from February until December in 2018, despite the nameserver providing only a single `A` record at a given time. We also notice that, at the opposite end of the spectrum, some APTs make substantial reuse of IP addresses. For example, the Donot APT C&C subdomains `jasper`, `qwe`, `alter`, `genwar`, `param`, `car` and `bike` of `.drivethrough.top` shared the same IP address from April 2019 until July 2019.

While both phishing or botnet campaign infrastructure tend to exhibit large numbers of `A` or `NS` records (compared to RFC recommendations) as a defense against take-downs [34], we notice that APT C&C domains tend to deploy even fewer such records than popular legitimate domains do. Also DNS `TXT` records can be abused by APT tools. For example, RoyalDNS of APT 15 (Ke3chang) used `TXT` records to send malicious payloads [17], HTTPBrowser and Pisloader of APT 18 (Wekby) used them to exfiltrate data [29], and also FIN7 (Carbanak) embedded data in them [12].

To *maintain persistence* over the network, APTs adopt many techniques including altering registry keys, scheduling tasks and using additional malicious domains for fall-back channels, in case one of the earlier ones had been detected and taken down by defenders (❽). We even noticed taken-down domains being later released back to the market and used again for the same campaign. For example, although `maccaffe.com` is a typosquatting of `mcafee.com` registered on October 2019, its historical records reveal that the domain has been resolved to 99 distinct IP addresses from August 2013 until December 2019. Similarly, `0ffice.com` and `hotmai1.com` have been active since 2008 and were still active at the time of writing this paper, despite violating common registrar regulations.

3 HAWK-EYE

In order to detect if a domain name being accessed by a monitored host belongs to APT C&C infrastructure, we have implemented Hawk-Eye, a system that processes PCAP files to extract and classify relevant domains. Our classification is based on a Random Forest classifier, which uses the *semantic*, *contextual* and *hybrid* features described in Sections 4 and 5. In order to evaluate our classifier, and facilitate further research in the area, we built the Hawk-Eye Dataset, [7] which collects a large number of features for the C&C domains of the APT campaigns of Section 2.

3.1 Architecture

Hawk-Eye is composed of the *Parser, Crawler, Preprocessor* and *Classifier* modules. The Parser inputs a PCAP file and extracts from DNS queries the Fully Qualified Domain Names (FQDNs) to be classified. Next, these FQDNs are segmented into host, Entity Level Domain (ELD) and public suffix, as explained in Section 4.1. The output of the Parser is used directly to extract semantic features, and is passed to the Crawler, to retrieve contextual and hybrid features. For active campaigns, live WHOIS and DNS information suffices, whereas for older campaigns, including those with expired registration or run by a sinkhole operator, we query the historical information from SecurityTrail [5] as discussed in Section 3.2.

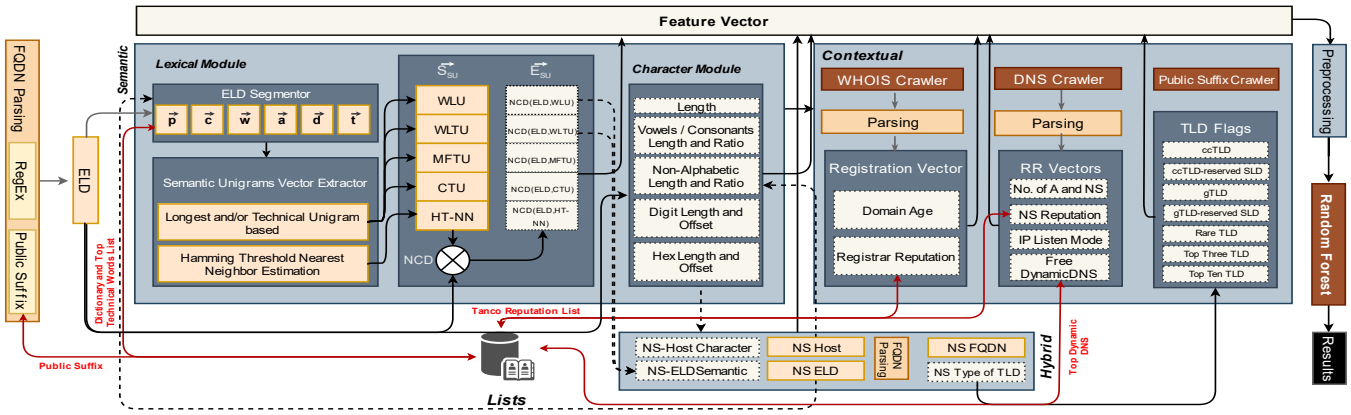

Figure 2: Hawk-Eye data flow. $\vec{p}$, $\vec{c}$, $\vec{w}$,$\vec{a}$ $\vec{d}$ and $\vec{t}$ are the results of segmenting an ELD with respect to popular, common, worldwide, ASCII, English and technical vocabularies (see Section 4.3). Light-shaded boxes denote elements of the feature vector.

The WHOIS collector sub-module requests registration information from WHOIS servers, and is able to parse 23 different WHOIS formats, covering the vast majority of the Alexa top 500K domains. The DNS collector sub-module handles the Resource Record (RR) responses we need to process. RRs are individually obtained with independent requests, as several nameservers disable `ANY` queries for security reasons and to comply with RFC8482 [19]. The RData payload of different RRs have different formats: Hawk-Eye implements parsers for all the RData and zone file formats relevant to any domain in our datasets. The data obtained by the Crawler is passed to the Preprocessor, which normalizes data, fills in missing values and encodes categorical and ordinal features in order to meet the requirement of different machine learning models. Finally, the Classifier labels each FQDN, according to the chosen classifier. For the etherogeneous features considered here, we found Random Forest to be the most appropriate model. Figure 2 shows the flow of data in Hawk-Eye, from observed FQDN to classification result. Further details on each stage are provided in Sections 3.2-6.

3.2 Datasets

As part of our analysis of APT campaigns from Section 2 we collected 5687 FQDNs and 2984 unique domains which were used as part of APT C&C infrastructure. After the quality-control described in Section 3.2.2, we kept respectively 3894 FQDNs and 1894 domains. For each sample we collected metadata recording the specific APT campaign ID, the period of activity, the report source, and other relevant data such as the IP addresses of C&C servers. We omitted domains that are not explicitly attributed to C&C, such as the ones used as part of an initial compromise phase via spear phishing. Some APT reports provide ranges of malicious domains: in such cases we collected only one concrete example to limit redundancy.

As a primary source of comparison against APT domains, in November 2019 we collected 47 thousand legitimate domains, randomly selected from the Alexa 1M list following a distribution heavily skewed towards the highest ranking sites. We collected one order of magnitude more legitimate domains than APTs in order to support unbalanced training of classifiers. Although it is easy to collect legitimate domains, we did not attempt to reflect the actual frequency of APTs in typical traffic, which is unknown and likely to be vanishingly small. Instead of a campaign ID, we assign to each legitimate domain a label representing its ranking bracket, such as `Alexa 101-1000`. To further compare APT domains against different malicious domains which may use related evasion techniques, in July 2020 we collected 6804 phishing FQDNs with 2523 unique domains labeled as valid and active from PhishTank [2].

In our experiments, we use three combinations of these datasets: *Hawk-Eye APT and Legitimate* (**HEAL**), *Hawk-Eye APT and Phishing* (**HEAP**), and *Hawk-Eye APT, Legitimate and Phishing* (**HEALP**).

3.2.1 Data Collection Challenges. For malicious domains, we tried to obtain the features relative to the time when their respective campaigns were active, by querying the SecurityTrail API [5] for historical records, as current live data may not reflect the actual domain ownership and configuration whilst malicious. Several factors contribute to complicate the analysis of historical data, which is crucial to obtain reliable features. Some malicious domains were kept online by a sinkhole operator for more than 3000 days [8]. Others were taken-down and subsequently released, and repurchased for legitimate or malicious purposes [8]. While inspecting historical DNS data, we found that most malicious domains have tens of `A`, `NS`, `MX`, `SOA` and `TXT` records. For instance, the domain `ceofanb18.mipropia.com` used by Machete [13] has more than 600 nameserver configurations since December 2017, whereas `corporatefaxsolutions.com` used by Carbank resolved to 433 different IP addresses since July 2015.

In order to zoom in on the most likely attack configuration, we select historical data in a given APT campaign time window, and we prioritize matches for `NS` reported in threat intelligence feeds for the same campaign or an overlapping one. For example, that led to choosing `ns1621.ztomy.com`, `ns2621.ztomy.com` and `brit.ns.cloudfronts.services` for DarkHydrus APT. This is a manual and time consuming process, due to the insufficient analysis of NS records in the published reports. Once we identify the validity window of the nameserver used during the campaign, we are able to filter relevant `A` records within that time interval. If we identify several records with different dates during a interval,

ID	Feature	Type	New?	ID	Feature	Type	New?
	I. Semantic Features						
1	Apex consonants len.	N	✓	14	ELD max digit len.	N	[31]
2	Apex consonants ratio	N	✓	15	ELD max digit offset	N	[31]
3	Apex len.	N	[8, 10, 14, 23, 34]	16	ELD Hex len.	N	[31]
4	Apex vowels len.	N	[28]	17	ELD max Hex offset	N	[31]
5	Apex vowels ratio	N	[28]	18	Longest meaningful string	N	[11]
6	Apex start with vowels	B	[28]	19	NCD_e(ELD, CTU)	N	✓
7	ELD's WLU len.	N	✓	20	NCD_e(ELD, HT-NN)	N	✓
8	ELD's WLTU len.	N	✓	21	NCD_e(ELD, MFTU)	N	✓
9	ELD's WLU pct	N	✓	22	NCD_e(ELD, WLTU)	N	✓
10	ELD's WLTU pct	N	✓	23	NCD_e(ELD, WLU)	N	✓
11	ELD non-alphabetic len.	N	[28, 31]	24	Technical words pct	N	✓
12	ELD non-alphabetic ratio	N	[28]	25	Top 1k pct	N	✓
13	ELD digit len.	N	[28]	26	Top 10k pct	N	✓
	II. Contextual Features						
27	Domain age	N	[25, 30, 31]	34	FQDN's Rare TLD	B	✓
28	Free Dynamic DNS-NS	B	✓	35	IP listen mode	N	✓
29	FQDN's ccTLD	B	[25, 34]	36	No. of IP	N	[30]
30	FQDN's gTLD	B	[25, 34]	37	No. of NS	N	[31, 34]
31	FQDN's reserved SLD	B	✓	38	NS reputation	N	[31]
32	FQDN's Top 3 TLD	B	✓	39	Registrar reputation	N	[31]
33	FQDN's Top 10 TLD	B	✓				
	III. Hybrid Features						
40	NCD_e(NS-ELD, NS-WLU)	N	✓	57	NS-Host max Hex offset	N	✓
41	NCD_e(NS-ELD, NS-WLTU)	N	✓	58	NS-Host consonants len.	N	✓
42	NS-ELD WLU len.	N	✓	59	NS-Host consonants ratio	N	✓
43	NS-ELD WLU pct.	N	✓	60	NS-Host non-alphabetic len.	N	✓
44	NS-ELD WLTU len.	N	✓	61	NS-Host non-alphabetic ratio	N	✓
45	NS-ELD WLTU pct.	N	✓	62	NS-Host startwith vowels	B	✓
46	NS-ELD digit len.	N	✓	63	NS-Host vowels num	N	✓
47	NS-ELD max digit len.	N	✓	64	NS-Host vowels ratio	N	✓
48	NS-ELD max digit offset	N	✓	65	NS len	N	[34]
49	NS-ELD max Hex len.	N	✓	66	NS-rareTLD	B	✓
50	NS-ELD max Hex offset	N	✓	67	NS-ccTLD	B	✓
51	NS-ELD non-alphabetic len.	N	✓	68	NS-gTLD	B	✓
52	NS-ELD non-alphabetic ratio	N	✓	69	NS Top 3 TLD NS	B	✓
53	NS-Host digit len	N	✓	70	NS Top 10 TLD	B	✓
54	NS-Host max digit len.	N	✓				
55	NS-Host max digit offset	N	✓				
56	NS-Host max Hex len.	N	✓				

Table 1: Hawk-Eye features (N: numerical, B: boolean).

we select the ones closest to the date the domain was reported. We proceed in a similar way for historical WHOIS records.

3.2.2 Missing Values. We have adopted a rigorous methodology to handle missing data in our datasets. In order to reduce the bias introduced by imputation, we omit features for which at least 20% of the values are missing. For example, WHOIS features such as registrant email, country, etc adopted in [31] are missing from up to 40% of the data, due to the adoption of new privacy rules on WHOIS files for gTLD domains, and because of ccTLD WHOIS servers that choose not to report that information to the public. We also removed features related to `CNAME`, `MX`, `SOA` and `TXT` which were missing from the majority of legitimate and malicious domains. For some FQDNs we could not find any nameserver although the domains were not taken down. In such cases we assumed the APT campaign used dynamic DNS services during the attack for domain resolution, and then deleted the records for anonymity. Recent domains tend to lack historical data to capture such behavior, yet we noticed several domains using this technique. In a similar way, we excluded from our dataset entire campaigns (FIN 8, APT 38 and others) where every single domain missed at least one feature.

4 SEMANTIC FEATURES

Semantic features, such as the length of a domain, or its decomposition into meaningful parts, reflect the choice of a particular domain name. These aim to capture DGA-wordlist- and phishing-like techniques used by APTs, including typosquatting (`telagram.net`), TLD squatting (`microsoft.store`), and the use of technical words (`accounts-google.com`). A selection of the semantic features described below are summarized in Table 1.

4.1 ELD Identification

As a preliminary step, we split each FQDN into an *apex* domain (the registrable part) and a possibly empty *host* prefix. Next we split the apex into its largest *public suffix* [4], and its *ELD*. The ELD represents the part of the domain name chosen by the entity owning and controlling it. For example, the Strider APT domain `ping.sideways.ru` has host `ping`, apex `sideways.ru`, and ELD `sideways`, whereas `mynetwork` is the ELD of the APT33 domain `mynetwork.ddns.net`. Identifying the ELD precisely, and in particular avoiding the pitfall of including dynamic DNS, public hosting providers or a reserved SLD, is important. Several APTs use dynamic DNS providers, such as `ddns.net` above, to bypass domain detection techniques which overlook this distinction. Moreover, one may want to extract character and lexical features such as the ones described below, or those used in related work (length, number of vowels, string similarity, etc.) only from the ELD, and not for example from the public suffix of a domain.

4.2 Character Features

The length [24, 28, 31, 34] and the number of vowels [28] of a FQDN have proven to be useful features for malicious domain detection. Besides these, we also collect the number of consonants, the ratio of vowels and consonants to length, and a boolean feature for whether or not a word starts with a vowel. We apply these 6 features to the Apex and ELD. From the ELD we also extract the total number of digits, the longest sequence of digits and the maximum digit offset, and similarly for hexadecimal strings of even length [31]. We also record the percentage of digits in the ELD string [11]. Finally we collect new features counting the number and ratio of non-alphabetical characters in an ELD.

4.3 Lexical Features

Before extracting lexical features we try to split a domain name (specifically, an ELD) into its constituent words, if possible.

4.3.1 Vocabularies. In NLP, typical approaches to segment a sentence into tokens include dedicated APIs such as `nltk.tokenize` [3], or regular expressions with explicit word separators, such as whitespace, comma, colon, and period. However, many of the domains we consider concatenate multiple words together without any separator, and such techniques cannot be applied directly. To the best of our knowledge there is no specific technique to segment domain names semantically.

Although the English language is commonly used for conveying meaning through the choice of a domain, also brands, technical words, and the ASCII representations of words in other languages are pervasive. It is trivial to identify `apple.com` as the word "apple", matching an English word, a brand name and a popular domain [21]. The case for `howstuffworks.com` is more subtle: it can be segmented as the popular domain "howstuffworks", or as the English words "how", "'stuff" and "works". Similarly `yandexmail.ru` should be segmented as "yandex", an ASCII sequence denoting another popular domain, and the English word "mail".

Campaign	ELD.SUFFIX	ELD Segmentation Vectors			Unigram Extractions			ELD Neighbor	
		$\vec{p}$	$\vec{a}$	$\vec{t}$	WLU	MFTU	CTU	HT-NN	h
DarkHydrus	hotmai1.com	None	None	None	hot*	None	None	hotmail	1
Pegasus	foxlove.life	None	[love]	None	love	None	None	foxlive	1
APT28	updatecenter.name	None	[update, center]	[update, enter]	updatecenter	update	updateenter	irdatacenter	3
APT35	gmail-com.xyz	[mail]	[gmail]	[gmail]	gmail	mail	gmail	email-hog	3
APT15	englishedu-online.com	None	[english, u-on, line]	[online]	english	None	online	englishhelponline	4
APT37	securytingmail.com	[mail]	[securyti, gmail]	[gmail]	securyti	mail	gmail	security4arabs	7
APT39	update-microsoft.space	[microsoft]	[update, microsoft]	[update, microsoft]	microsoft	updatemicrosoft	updatemicrosoft	updatesmartphone	8
APT32	googleuserscontent.org	[google]	[google, user, cont]	[google, user, content]	content	googleuser	googleusercontent	localguidesconnect	9
FIN7	windowsupdatemicrosoft.com	[microsoft]	[windows, update, microsoft]	[windows, update, microsoft]	microsoft	windowsupdatemicrosoft	windowsupdatemicrosoft	windowsactivatorloader	14

* In this example, **WLU** selects only the unigram "hot" from the English dictionary ELD segmentation vector $\vec{d}$, not reported in the table due to space limitations.

Table 2: APT domain segmentation: examples from the Hawk-Eye Dataset.

Based on the intuition above we collect a number of vocabularies as a basis for domain name segmentation: note that we do not use these directly to determine the maliciousness of a domain.

Inspired by [9], who collect two lists with the 8 and 100 most popular domains, we build non-overlapping vocabularies with the ELDs of the **top 100 (popular), 1k (common),10K (worldwide) and 500k (ASCII)** Alexa domains. Our goal is to increase the coverage of brands, abbreviations and non-English-words which are common in domains which may be targeted by phishing-like, typosquatting or similar impersonation attempts. We also build a vocabulary of 350 **technical terms**, which could be useful to an APT for impersonating a process or piece of infrastructure (as discussed in Section 2), including terms such as: *update, DNS , mail, support, account, CDN, API, cloud*. Finally, we build an **English vocabulary** of the 10k most common English words from the Cambridge dictionary.

4.3.2 Segmentation Features. In order to segment a target ELD with respect to a vocabulary, we collect all the ordered, non-overlapping matches against a regex which contains the vocabulary words in decreasing length order (to favor longest matches). Once we have segmented an ELD, we join the list of matches, and we save the length of the resulting unigram, as well as its ratio to the ELD length as features. These features attempt to quantify how much of the ELD is semantically related to an entity of interest. For example, a fake ELD `yandexm4il-cdn` would be segmented by the union of common and technical vocabularies into the unigram `yandexcdn`, yielding a 65% match.

For completeness we also mimic a lexical feature proposed in [11] by saving the length and ratio to the ELD of the longest substring of the ELD that has a match in the English vocabulary.

4.3.3 Unigrams and NCD_e Features. Besides direct segmentation as described above, we propose five more flexible methods to select a target unigram for comparison with a given ELD. We then compute the *Entropy Normalized Compression Distance (NCD_e)* between the ELD and each unigram as proxies for similarity, where a small NCD_e implies a high similarity. Also these features help identifying domain spoofing attempts.

After segmenting the ELD with respect to the popular, common, worldwide, ASCII, English and technical vocabularies , the *Weighted Longest Unigram* (**WLU**) collects the longest unigrams, concatenating multiple unigrams if they have the same length. The WLU preserves the order of match from the ELD and does not include overlapping matches. Concrete examples of this unigram and the ones below are provided in Table 2. The *Weighted Longest Technical Unigram* (**WLTU**) instead is a restricted form of WLU using only the technical vocabulary. The *Most Frequent Technical Unigram* (**MFTU**) also only considers technical terms, but the segmentation is based on a re-sorting of the vocabulary according to the frequency of each term in the training set, cutting the vocabulary at a threshold τ. The *Concatenated Technical Unigram* (**CTU**) is a variant of the MFTU that does not consider frequency. The last unigram we consider is the *Hamming Threshold Nearest Neighbor* (**HT-NN**), which consists of the ELD from the ASCII vocabulary which is closest in Hamming distance to the ELD of the FQDN being classified. Besides collecting NCD_e between the ELD and the HT-NN, we also collect the Hamming distance itself as a feature. This captures the intuition that APT operators may choose domain names that are subtle alterations of existing ones, such as replacing one character with a similar looking one. For example, a Hamming distance of 1 captures the APT domain `go0gle.com` which impersonated `google.com`.

5 CONTEXTUAL AND HYBRID FEATURES

Contextual features are orthogonal to semantic features, and express the organization of the DNS infrastructure that supports the C&C operation for the chosen domain. We expect the low-and-slow nature of APTs to differentiate their infrastructure from that of generic malware, or regular domains. *Hybrid* features collect semantic features from the DNS infrastructure. Selected contextual and hybrid features are reported in Table 1.

5.1 Contextual Features

We observed in Section 2 that APT C&C behavior involves other aspects of the DNS infrastructure besides the choice of an ELD. Hence, we proceed to consider orthogonal features, based on the domain infrastructure used by APTs.

5.1.1 Domain Suffix. As discussed in Section 4.1, we hypothesize that the type of TLD used by a domain, and in particular the presence of a public SLD denoting virtual hosting or dynamic DNS may be relevant to detecting C&C domains. Hence we propose 4 boolean features recording the kind of public suffix of a domain: ccTLD, gTLD, ccTLD, and gTLD with a reserved SLD. In addition, we propose 3 features, recording if a public suffix belongs to the top three TLDs (`.com`, `.net` , `.org`), top ten TLDs or bottom 100 TLDs based on the frequency observed in the training fraction of our dataset.

5.1.2 Domain Age. WHOIS servers are responsible to provide registration information for a domain for public use, including the owner name, primary nameserver, admin email, registrar, creation and expiry dates. WHOIS information has been used to detect malicious domains in the past, for example in [15, 24, 34]. However, recent increases in privacy regulation and concerns from the users have led WHOIS servers to severely restrict the amount of information divulged, therefore several features that have been used in the past are no longer widely available. The domain age is computed as the difference between expiration and creation date for a given domain, expressed in months [24]. This feature is present in historical datasets and is also currently available when querying up to date WHOIS information.

5.1.3 Registrar Reputation. This feature approximates the reputation of the Registrar URL parsed from a WHOIS file by its ranking in the Tranco List [21]. We used Tranco instead of Alexa as the latter has been found vulnerable to poisoning [21].

5.1.4 DNS Resource Records. In Section 2.3 we discussed some examples of DNS abuse by APTs. Here we consider related DNS features relevant to APT C&C domain classification. The `A` resource record is used to communicate the IPv4 addresses a domain resolves to. The `NS` record, for communicating the nameservers storing the zone file for the domain. We use, as features, the count of `A` and `NS` entries found in the respective responses for a candidate domain being classified.

5.1.5 Nameserver Reputation. This feature reports the Tranco-based reputation of the apex of the `NS` of a domain. The idea is that some APTs will not be able to meet the stringent anti-fraud requirements of highly ranked DNS providers, and resort to less popular ones. However, several APT campaigns are able to comply, and have used for example `domaincontrol.com` by GoDaddy.

5.1.6 Use of Free Dynamic DNS. Dynamic DNS (DDNS) is an approach to update the mapping of a domain to different IPs quickly and automatically. A known technique for domain flux [23, 25] is to use *Free DDNS Hosting* for malicious purposes, where the ELD is controlled by the attacker but the reserved SLD refers to the provider itself [9, 23, 25, 27]. Several APT campaigns, such as APT32, APT 37, APT 41, FIN7 and SilverTerrier [6], instead abuse DDNS in a different way, by directly delegating the nameserver of a C&C domain to a free DDNS provider. Therefore, we collect a list of top 40 free DDNS providers and extract a boolean feature to record if the `NS` of a domain is in the list.

5.1.7 IP Listen Mode. Some APT campaigns configure their C&C domain to be resolved to the *loopback address* (`127.0.0.1`) or to a *non-routable meta-address* (`0.0.0.0`) [1] in order to let the victim connect to an attacker-controlled processes listening on the same machine, on a specific open port. We extract a feature recording if the `A` of a domain is internal or non routable.

5.2 Hybrid Features

The Hybrid feature set consists of semantic features extracted from entities retrieved as part of the contextual analysis of a domain. For the ELD of a `NS`, we add lexical features similar to the ones described in Section 4, although we only use WLU and WLTU to identify the longest (technical) unigrams for computing the NCD_e. Finally we add character features, including consonant length, non-alphabetic length, max digit and hex length, digit and hex offset for both the ELD and the Host of the `NS`.

The Semantic, Contextual and Hybrid features reported in Table 1 are combined as the *Holistic* feature set.

Training Sets		Testing Set	
APT1_CommentCrew	APT40_Leviathan	AnimalFarm_APT	Higaisa
APT12_IXESHE	APT41_DoubleDragon	APT_Big_Bang	ICEFOG_APT
APT15_ke3chang	APT_Robotic	APT_C_37	KONNI_APT
APT16_EPS	LazarusGroup	APT_C_39	Microcin_APT
APT17_Deputy_Dog	Calypso_APT	APT_CYBEC_TIA	MuddyWater_APT
APT18_Wekby	CobaltGroup_CobaltSpider	APT_Pirate_Panda	Mustang_Panda_APT
APT19_C0d0so0	DarkHydrus	APT10_menuPass	Operation_Transparent
APT2_PutterPanda	Elderwood_OperationAurora	APT23	Pierogi_APT
APT27_EmissaryPanda	FIN7_Carbanak	APT34_OilReg	Scarlet_Mimic_APT
APT28_SednitSofacy	Machete_El_Machete	APT35_MajicHound	StrongPity3_APT
APT29_CozyDuke	NaikonAPT_MsnMM	BITTER_APT	Trident_APT
APT3_Gothic	Patchwork	Chinese_mAPT	Turla_APT
APT30	Pegasus	Deep_Panda_APT	
APT32_OceanLotus	Poseidon_Group	Donot_APT	
APT33_Elfin_Shamoon	Sowbug	Enfal_APT	
APT37_ScarCruft	Strider_PROJECTSAURON	Etirehni_APT	
APT39_Chafer	Taidoor	Gamaredon	

Table 3: Training/testing split of APT campaigns.

6 EVALUATION

We now evaluate and discuss the detection performance of Hawk-Eye, using the feature sets defined in Sections 4 and 5. Since there is no publicly available tool that detects APT C&C domains, we cannot compare with an accepted detection baseline. Instead, we used established features from the literature (those with a citation in Table 1) to create an additional *Literature Baseline* feature set for comparison.

6.1 Classification Performance

We compare the various feature sets on three detection tasks: APT versus legitimate (HEAL), APT versus phishing (HEAP) and APT versus non-APT, that is both legitimate and phishing (HEALP).

In order to prevent leaks from training to testing sets, and to help our results generalize, we follow the methodology of [16] and split each dataset so that APT campaigns are either entirely contained in the training set or in the testing set, as reported in Table 3. This leads to an approximate split of 70% APT samples for training and 30% for testing, which we mirrored for the other classes.

While the classes of the HEAP dataset are mostly balanced, APT domains are 4% of the samples in the HEAL and HEALP datasets. Hence, the results reported in Table 4 are the *weighted average* across the two classes, approximating the performance on a balanced dataset. We also report the macro F1 score (***mF*1**), which reflects the existing 4%/96% bias. The best overall performance of Hawk-Eye is obtained by the Holistic features set, with respectively 98.53%, 88.66% and 98.39% *F*1 for the HEAL, HEAP and HEALP datasets.

6.2 Feature Importance

In Figure 3 we plot the feature importance for the Holistic classifier and for the literature baseline. Different features turn out to have different importance for different tasks. We focus our analysis on the

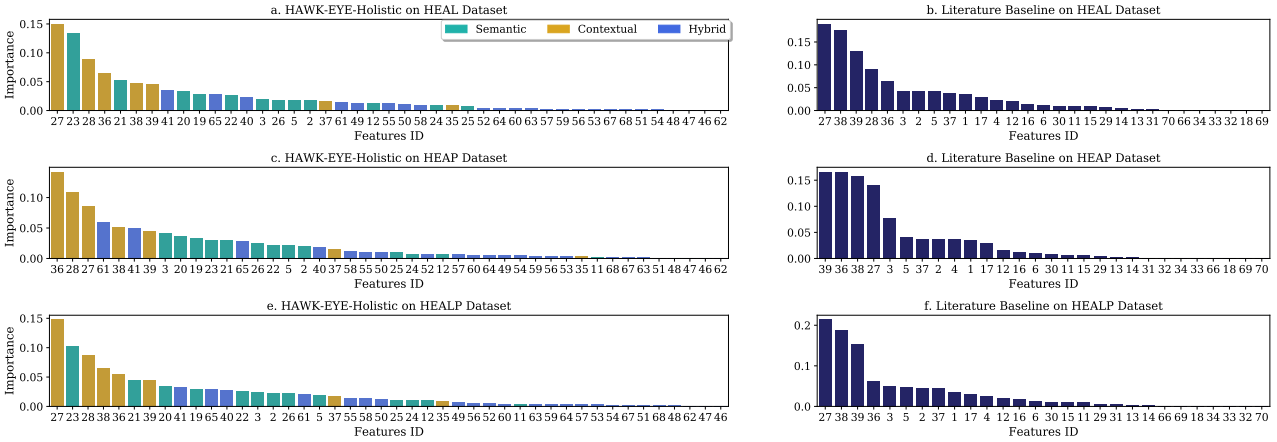

Figure 3: Feature importance: comparison between Hawk-Eye-Holistic and Literature Baseline across all datasets.

Classifier Name	Precision	Recall.	Accuracy	*F*1	*mF*1
I. Dataset: HEAL					
Hawk-Eye-Semantic	96.64	95.66	96.96	96.49	74.99
Hawk-Eye-Contextual	97.53	95.85	97.63	97.40	81.83
Hawk-Eye-Hybrid	95.94	**97.27**	97.45	97.32	81.95
Hawk-Eye-Holistic	**98.52**	96.09	**98.51**	**98.53**	**90.31**
Literature Baseline	95.64	96.55	96.95	96.58	76.81
II. Dataset: HEAP					
Hawk-Eye-Semantic	74.21	74.31	74.36	74.18	73.88
Hawk-Eye-Contextual	81.21	80.85	80.86	80.91	80.91
Hawk-Eye-Hybrid	85.11	85.01	85.11	84.92	84.92
Hawk-Eye-Holistic	**88.72**	**88.64**	**88.65**	**88.66**	**88.56**
Literature Baseline	79.37	79.40	79.40	79.34	79.06
III. Dataset: HEALP					
Hawk-Eye-Semantic	95.89	96.56	96.56	95.74	67.55
Hawk-Eye-Contextual	97.43	97.63	97.63	97.32	80.83
Hawk-Eye-Hybrid	97.52	97.73	97.73	97.52	82.81
Hawk-Eye-Holistic	**98.40**	**98.48**	**98.48**	**98.39**	**89.15**
Literature Baseline	96.22	96.76	96.77	96.28	73.05

Table 4: Classification performance.

HEALP dataset which, including both APT, legitimate and phishing domains, is the most relevant to detecting APT domains the wild.

For convenience, we divide features intro three groups based on importance thresholds of 0.10, 0.05 and 0.025. The first group only contains two features. The first is Domain age (#27), which is known to be an effective feature to detect malicious domains [25, 30, 31], and remains relevant for APTs. The second is $\text{NCD}_e(ELD, WLU)$ (#23), one of the new features we propose, with mean and standard deviation for APT 0.136 (±0.087) , phishing 0.121 (±0.01), and legitimate 0.047 (±0.077). It captures the well-known fact that some APT and phishing domains attempt to resemble popular ones, without exactly matching them. In the next group, 34.46 %, 1.99% and 2.05% of APT, phishing and legitimate domains use free DDNS-NS (#28), respectively. We also observed that some of these nameservers have no reputation (#38) based on appearance in the Tranco list: 23.93 % for APTs, 25% for phishing and 14% for legitimate. Finally the average number of IPs (#36) used by APTs is 1.07 (±0.40), phishing 1.78 (±1.63) and legitimate 2.25 (±2.67). The importance of these novel features confirms our hypotheses from Sections 4 and 5.

The third group contains a number of features that still contribute to classification but that may be effective only for a smaller subset of domains. We verified that performance degrades omitting each of these features. They include registrar reputation (#39) and variants of NCD_e using *MFTU, HT-NN* and *NS-WLTU*.

6.3 Discussion

The Hawk-Eye-Holistic (HH) feature set consistently outperforms the Literature Baseline (LB). For the HEALP dataset, the FPR, macro recall and precision of the LB, standing respectively at 0.69%, 36.47% and 69.23% are much worse than the corresponding figures for HH, which stand at 0.35%, 70.83% and 89.55%. The 4% positive rate of HEALP that produces these numbers does not reflect the actual frequency of APT domains in corporate traffic: the performance gap between HH and LB increases as the positive rate declines, as can be seen by the larger gap in *mF*1 as opposed to *F*1. In fact, *mF*1 decreases with the number of positives, and at the limit (no positives: we are monitoring a clean network) only the FPR matters.

Our split of training and test set by campaign prevented us from using cross-validation techniques. Due to the limited number of APT domains available for training and testing, we did not set apart a validation set, and so we did not optimize the RF parameters. Instead, in order to include a sufficient number of legitimate domains and better approximate the legitimate distribution, preventing an artificial separation between the APT and legitimate class, we chose to use an unbalanced data set. In fact, the training error for a class-balanced subset of HEAL is 0, whereas the *mF*1 is 90%, indicating overfitting. With the current class imbalance the training error is much closer to the testing error. Increasing the ratio of negatives much further negatively affects the recall for APTs, as legitimate samples overwhelm the model.

Figure 4 shows the means of the top 6 features of the contextual, semantic and hybrid feature sets for each label class. The highest mean of each feature is normalized to 1.0, and the other means are scaled accordingly. We can see that contextual features play a substantial role in distinguishing APTs from both legitimate and phishing domains, whereas semantic and hybrid features single out

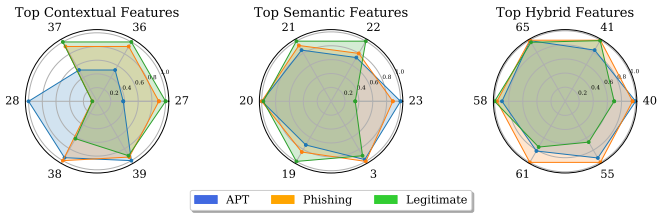

Figure 4: Relative performance of top features.

Domain	Rank	Domain	Rank
googletagmanager.com	24	mp3-youtube.download	5595
youtube-nocookie.com	201	microsoftazuread-sso.com	11132
ssl-images-amazon.com	1011	microsoftonline-p.com	12337
onmicrosoft.com	2808	facebook-danger.fr	34322
firefoxchina.cn	4433	allfacebook.com	27426
symantecliveupdate.com	4407	microsoftemail.com	73976
mcafeemobilesecurity.com	4729	kasperskycontenthub.com	160661

Table 5: Legitimate domains resembling APT domains.

legitimate domains but present fewer differences among malicious domains. The strength of contextual features suggests that domain configuration is a weak spot in the infrastructure of APTs. Character and lexical features tend to have a more mixed performance. Lexical analysis is effective in detecting malicious domains posing as legitimate ones, but we note that it is prone to false positives. In fact, several popular legitimate domains have similar characteristics to malicious ones. A few representative examples are reported in Table 5. Overall our results confirm that in the case of APTs a holistic approach is best suited for malicious domain detection.

6.4 Limitations

Although the FPR of HH at 0.35% can be considered low in the context of the malicious domain detection literature, we think it is still too high for Hawk-Eye to be used as a standalone detector in practice, as the number of false positives for a much higher (realistic) negative rate would be excessive. As it stands, Hawk-Eye can be used as a pre-filtering step to substantially reduce the traffic to be scrutinized for APT presence, or it can be employed in forensic investigations, where a limited window of traffic needs to be analyzed to detect actual APT C&C traffic.

The legitimate domains in our datasets are taken from a website ranking list, and consist mostly of apex domains. Legitimate sessions of end-user traffic would likely include a large number of domains including a host, and would be better suited for a comparison with APT domains, which predominantly include hosts. To avoid introducing a bias in our experiment, we limited ourselves to consider apex features for FQDNs: we leave it to future work to build a dataset including legitimate host features. We experimented with adding host features to the current feature sets, but they flattered APT detection performance, and overwhelmed the other features. For example, the average length of hosts is very close to 0 only for the legitimate class of HEALP, which is not reflective of real world traffic.

Another limitation of our approach is that our segmentation and unigram techniques do not take into account fuzzy matching of words, where characters may be added and removed, thereby missing out on detecting further typosquat variants of popular brands and domains. In fact, the HT-NN unigram based on Hamming-distance is only able to handle character replacement. For example, the APT domain `samrsung.com` is misclassified as legitimate but it could plausibly be caught by a fuzzy segmentation, matching it as a variant of the popular domain `samsung.com`. We leave this extension of Hawk-Eye to future work.

Finally, our legitimate and phishing domains were collected during specific and limited intervals of time, whereas the APT domains span 13 years. This may introduce a form of temporal bias, although we have mitigated that by not using the same APT campaigns for both training and testing. In general, Hawk-Eye should be periodically re-trained in order to adapt to the drift in actual adversarial and legitimate behavior.

7 RELATED WORK

There is very limited previous work on detecting APT domains [24, 39], and it tends to suffer from a lack of details on the features used and on the composition of datasets. As discussed in Sections 4 and 5, several features we considered were inspired by previous work on malicious domain detection, and even more have been considered in the literature. Domain length [25, 28, 31, 34, 37] , % of numerical characters [28, 31], length of meaningful strings [11], vowels and consonant characteristics [28], and TTL [11, 28], NXDomain responses [10, 28, 33] have proven particularly useful to detect DGAs. Notos [9] is a detection system which combines domain ranking with geolocation information based on BGP and AS. In our analysis of APT campaigns we noticed that APTs are able to locate their infrastructure worldwide, and even inside the country hosting the targeted organization, so we did not attempt to geolocate domains. Exposure [11], Pleiades [10] and HinDom [33] detect malicious domains using features such as daily similarity, repeating patterns and access ratio features which are less likely to be effective with the low, slow and stealthy connections used by APTs.

PREMADOMA [31] helps DNS registries prevent malicious domain registrations. The most characteristic features of [31] are related to the selection of name servers, contact emails and phone numbers. APTs often use bulletproof hosting providers, or less reputable ccTLDs which do not have an incentive to deploy measures against malicious domain registration. Although Hawk-Eye focuses on clients rather than registries or registrars, our features may also help identifying suspicious registrations.

YATT [34] is a browser-based framework to prevent users or adware from accidentally accessing typosquat domains. The framework includes WHOIS and DNS features such as total number of typos in nameservers, TXT Google verification and registration date. However, their WHOIS crawler only considers the `.com` format, excluding the substantial number of domains hosted on other TLDs. As described in Section 3.1, Hawk-Eye's crawler is able to handle the WHOIS format for most TLDs' WHOIS servers of the Public Suffix List [4], and may help extend the coverage of YATT.

MADE [25] is a SOC-like enterprise solution to analyze logs received from firewall, antivirus and web proxies, and detect malicious communications. It uses machine learning to assign a risk score

to each connection. However, MADE mainly considers malicious HTTP connections including URL parameters, User Agent features and domain-based features. We included some features from MADE in the literature baseline set, and we show that detection improves when those are combined with our new features.

Finally, PDNS [30] is a host-based malicious DNS detection system to detect encrypted DDNS requests. Since DNS is encrypted, PDNS mainly considers the location of DNS requests, in addition to GUI, UI and Web communication DDLs. This proved to be an effective strategy against botnets, where there is substantial traffic, but does not apply naturally to APTs, where the establishment of C&C is just an initial stage which employs several evasion techniques, as discussed in Section 2.

8 CONCLUSIONS

To the best of our knowledge, Hawk-Eye is the first system that attempts to detect C&C domains used by APTs at the network level, and ours is the first dedicated dataset publicly available. By leveraging a number of new and existing features captured at different levels (domain name, WHOIS, DNS records) our best classifier achieves a promising level of performance. A number of novel features introduced by us contribute to achieve the best performance. Hawk-Eye is a prototype built in Python, focusing on robustness, modularity and generality and designed to test different domain detection hypotheses. We envisage that a high-performance tool based on Hawk-Eye could work as a parallel component of a network intrusion detection system, but we leave a study of performance and deployment issues to future work.

Acknowledgments

We thank SecurityTrail for providing access to their historical data. This work was supported by Graduate Studies Scholarship at the National Center for Cybersecurity Technologies, KACST.

REFERENCES

[1] [n. d.]. Calypso APT: new group attacking state institutions. https://www.ptsecurity.com/ww-en/analytics/calypso-apt-2019/
[2] [n. d.]. Join the fight against phishing. https://www.phishtank.com/
[3] [n. d.]. Natural Language Toolkit. https://www.nltk.org/. Accessed: 2019-09-01.
[4] [n. d.]. PublicSuffix. https://publicsuffix.org/list/public_suffix_list.dat. Accessed: 2019-10-18.
[5] [n. d.]. SecurityTrails. https://securitytrails.com/corp/api. Accessed: 2019-01-22.
[6] [n. d.]. Splunk Security Essentials Docs. https://docs.splunksecurityessentials.com/content-detail/sse_dyndns/
[7] Almuthanna Alageel. 2020. Hawk-Eye Dataset. https://doi.org/10.14469/hpc/7675. Release date: 2020-12-17.
[8] Eihal Alowaisheq, Peng Wang, Sumayah A Alrwais, Xiaojing Liao, XiaoFeng Wang, Tasneem Alowaisheq, Xianghang Mi, Siyuan Tang, and Baojun Liu. 2019. Cracking the Wall of Confinement: Understanding and Analyzing Malicious Domain Take-downs.. In *NDSS*.
[9] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. 2010. Building a dynamic reputation system for dns.. In *USENIX Security*. 273–290.
[10] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. 2012. From throw-away traffic to bots: detecting the rise of DGA-based malware. In *USENIX Security*. 491–506.
[11] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis.. In *NDSS*. 1–17.
[12] Nick Carr, Kimberly Goody, Steve Miller, and Barry Vengerik. [n. d.]. On the Hunt for FIN7: Pursuing an Enigmatic and Evasive Global Criminal Operation. https://www.fireeye.com/blog/threat-research/2018/08/fin7-pursuing-an-enigmatic-and-evasive-global-criminal-operation.html. Accessed: 2019-05-01.
[13] ESET. July 2019. MACHETE JUST GOT SHARPER Venezuelan government institutions under attack.
[14] Robert Falcone, Bryan Lee, and Tom Lancaster. July 2018. New Threat Actor Group DarkHydrus Targets Middle East Government. https://unit42.paloaltonetworks.com/unit42-new-threat-actor-group-darkhydrus-targets-middle-east-government/. Accessed: 2019-12-13.
[15] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. 2010. On the Potential of Proactive Domain Blacklisting. *LEET* 10 (2010), 6–6.
[16] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. 2014. An empirical comparison of botnet detection methods. *Computers & Security* 45 (2014), 100–123.
[17] NCC Group. [n. d.]. APT15 is alive and strong: An analysis of RoyalCli and RoyalDNS. https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2018/march/apt15-is-alive-and-strong-an-analysis-of-royalcli-and-royaldns/. Accessed: 2019-04-18.
[18] Eric M Hutchins, Michael J Cloppert, and Rohan M Amin. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research* 1, 1 (2011), 80.
[19] M. Majkowski J. Abley, O. Gudmundsson and E. Hunt. 1987. Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY.
[20] FireEye Mandiant Lab. Feb 2013. APT1: Exposing One of China's Cyber Espionage Units.
[21] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *NDSS*. https://doi.org/10.14722/ndss.2019.23386
[22] Antoine Lemay, Joan Calvet, François Menet, and José M Fernandez. 2018. Survey of publicly available reports on advanced persistent threat actors. *Computers & Security* 72 (2018), 26–59.
[23] Zhou Li, Sumayah Alrwais, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2013. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In *IEEE S&P*. IEEE, 112–126.
[24] Weina Niu, Xiaosong Zhang, GuoWu Yang, Jianan Zhu, and Zhongwei Ren. 2017. Identifying APT malware domain based on mobile DNS logging. *Mathematical Problems in Engineering* 2017 (2017).
[25] Alina Oprea, Zhou Li, Robin Norris, and Kevin Bowers. 2018. Made: Security analytics for enterprise threat detection. In *ACSAC*. 124–136.
[26] Daniel Plohmann, Khaled Yakdan, Michael Klatt, Johannes Bader, and Elmar Gerhards-Padilla. 2016. A comprehensive measurement study of domain generating malware. In *USENIX Security*. 263–278.
[27] RSA. 2020. *Dynamic DNS: Data Exfiltration.* Technical Report. 6 pages. https://www.rsa.com/content/dam/en/solution-brief/asoc-dynamic-dns-data-exfiltration.pdf
[28] Samuel Schüppen, Dominik Teubert, Patrick Herrmann, and Ulrike Meyer. 2018. FANCI: Feature-based automated NXDomain classification and intelligence. In *USENIX Security*. 1165–1181.
[29] Aaron Shelmire. July 2015. Evasive Maneuvers by the Wekby group with custom ROP-packing and DNS covert channels. https://www.anomali.com/blog/evasive-maneuvers-the-wekby-group-attempts-to-evade-analysis-via-custom-rop. Accessed: 2019-04-14.
[30] Suphannee Sivakorn, Kangkook Jee, Yixin Sun, Lauri Korts-Pärn, Zhichun Li, Cristian Lumezanu, Zhenyu Wu, Lu-An Tang, and Ding Li. 2019. Countering Malicious Processes with Process-DNS Association.. In *NDSS*.
[31] Jan Spooren, Thomas Vissers, Peter Janssen, Wouter Joosen, and Lieven Desmet. 2019. Premadoma: an operational solution for DNS registries to prevent malicious domain registrations. In *ACSAC*. 557–567.
[32] Blake E Strom, Joseph A Battaglia, Michael S Kemmerer, William Kupersanin, Douglas P Miller, Craig Wampler, Sean M Whitley, and Ross D Wolf. 2017. *Finding Cyber Threats with ATT&CK™-Based Analytics.* Technical Report. The MITRE Corporation.
[33] Xiaoqing Sun, Mingkai Tong, Jiahai Yang, Liu Xinran, and Liu Heng. 2019. HinDom: A Robust Malicious Domain Detection System based on Heterogeneous Information Network with Transductive Classification. In *RAID*. 399–412.
[34] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. 2014. The long "taile" of typosquatting domain names. In *USENIX Security*. 191–206.
[35] Colin Tankard. 2011. Advanced persistent threats and how to monitor and deter them. *Network security* 2011, 8 (2011), 16–19.
[36] Ke Tian, Steve TK Jan, Hang Hu, Danfeng Yao, and Gang Wang. 2018. Needle in a haystack: Tracking down elite phishing domains in the wild. In *IMC*. 429–442.
[37] Colin Whittaker, Brian Ryner, and Marria Nazif. 2010. Large-scale automatic classification of phishing pages. In *NDSS*.
[38] Bin Yu, Jie Pan, Jiaming Hu, Anderson Nascimento, and Martine De Cock. 2018. Character level based detection of DGA domain names. In *IJCNN*. IEEE, 1–8.
[39] Guodong Zhao, Ke Xu, Lei Xu, and Bo Wu. 2015. Detecting APT malware infections based on malicious DNS and traffic analysis. *IEEE access* 3 (2015), 1132–1142.