Monitoring Enterprise DNS Queries for Detecting Data Exfiltration from Internal Hosts

Jawad Ahmed*,†, Hassan Habibi Gharakheili*, Qasim Raza*, Craig Russell†, and Vijay Sivaraman*

***Abstract*—Enterprise networks constantly face the threat of valuable and sensitive data being stolen by cyber-attackers. Sophisticated attackers are increasingly exploiting the Domain Name System (DNS) service for exfiltrating data as well as maintaining tunneled command and control communications for malware. This is because DNS traffic is usually allowed to pass through enterprise firewalls without deep inspection or state maintenance, thereby providing a covert channel for attackers to encode low volumes of data without fear of detection. This paper develops and evaluates a real-time mechanism for detecting exfiltration and tunneling of data over DNS. Unlike prior solutions that operate off-line or in the network core, ours works in real-time at the enterprise edge. Our first contribution is to collect and analyze real DNS traffic from two organizations (a large University and a mid-sized Government Research Institute) over several days and extract numerous stateless attributes of DNS messages that can distinguish malicious from legitimate queries. Our second contribution is to develop, tune, and train a machine-learning algorithm to detect anomalies in DNS queries using a benign dataset of top rank primary domains. To achieve this, we have used 14 days-worth of DNS traffic from each organization. For our third contribution, we implement our scheme on live 10 Gbps traffic streams from the network borders of the two organizations, inject more than three million malicious DNS queries generated by two exfiltration tools, and show that our solution can identify them with high accuracy. We compare our solution with the two-class classifier used in prior work. We draw insights into anomalous DNS queries of two enterprise networks by their anomaly scores, the trace of query count over time, enterprise hosts querying them, and TTL and Type fields of their corresponding responses. Our tools and datasets are made available to the public for validation and further research.**

***Index Terms*—DNS, Enterprise, Exfiltration, Anomaly Detection.**

I. Introduction

The Domain Name System (DNS) is used for converting domain names (*e.g.,* `google.com`) into IP addresses and as such constitutes a mission-critical service. However, DNS communication is relatively poorly policed by organizations (compared to services like email, FTP, and HTTP) and has been exploited by cyber-criminals to maintain covert communication channels with compromised hosts. The resulting damages can be huge, amounting to several million dollars in a single attack [2]. Based on a recent DNS security survey of Infoblox [3], 46 percent of the businesses of North America and Europe have faced the DNS exfiltration and about 45 percent are affected by DNS tunneling. Several high-profile data exfiltration breaches have been reported recently: the Sally Beauty breach (a theft of 25K credit cards) [4] and FrameworkPOS malware (a theft of 56M credit cards from Home Depot) [5] in 2014, BernhardPOS malware [6] in 2015, MULTIGRAIN malware [7] in 2016, Win32.Backdoor.Denis [8] in 2017, and UDPoS Malware [9] in 2018. In addition, there have been a number of DNS tunneling incidents in which malware actors used their DNS servers to send and receive the command and control commands to and from compromised hosts; examples include Feederbot [10] and botmaster [11], Morto worm [12], and Wekby pisloader [13].

One way for the attacker to exploit DNS is to register a domain (*e.g.,* `foo.com`) so that the attacker's malware in a host victim can then encode valuable private information (such as credit card numbers, login passwords or intellectual property) into a DNS request of the form `arbitrary-string.foo.com`. This DNS request gets forwarded by resolvers in the global domain name system to the authoritative server for the `foo.com` domain (under the attacker's control), which in turn sends a response to the host victim. This provides the attacker with a low-rate but covert two-way communication channel between a host victim and their command-and-control center.

Interestingly, enterprise firewalls are typically configured to allow all packets on UDP port 53 (used by DNS) since DNS is such a crucial service for virtually all applications. Some firewalls do offer enhanced DNS protection but these require deep packet inspection of DNS messages to identify the covert channel and then isolate domains that contain encoded data. The significant resources required for this capability [14], and the resulting impact on firewall forwarding performance, usually results in enterprise network operators disabling such features. This ability to transit firewalls gives attackers a covert channel, albeit a low-rate one, by which to exfiltrate private data and to maintain communication with malware by tunneling other protocols (*e.g.,* SSH, FTP) to command-and-control centers. As one example, the remote access trojan DNSMessenger [15] discovered in 2017 used DNS queries and responses to execute malicious powerShell commands on compromised hosts.

In this paper, we develop and validate a mechanism for real-time detection of DNS exfiltration and tunneling in two operational networks – a large University and a mid-sized Government Research Institute. Our **first** contribution is to collect and conduct a thorough analysis of real DNS traffic from the two organizations over several days

* J. Ahmed, H. Habibi Gharakheili, Q. Raza and V. Sivaraman are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, 2052 Australia. E-mails: {j.ahmed@, h.habibi@, q.raza@student.,vijay@}unsw.edu.au

† C. Russell is with CSIRO Data61, Sydney, NSW 2015, Australia. E-mail: craig.russell@data61.csiro.au

This submission is an extended and improved version of our paper accepted to the IFIP/IEEE IM 2019 conference [1]

and extract stateless attributes of DNS messages, such as length, entropy, dots, numerics, uppercase characters, and the number of labels, that can distinguish malicious from legitimate queries. Our **second** contribution is to develop, tune, and train a machine-learning algorithm to detect anomalous DNS queries based on the above attributes using a known dataset of benign domains as ground truth based on 14 days worth of DNS data from the two organizations. For our **third** contribution we implement our scheme on live 10 Gbps traffic streams from the network borders of the two organizations, inject more than three million malicious DNS queries generated using two exfiltration tools (our customized tool and an open-source tool) and show that our scheme is able to identify such malicious activity with high accuracy. We also show our one-class classifier outperforms an existing two-class classifier in detecting unknown DNS exfiltration attacks. We draw insights into anomalous DNS queries detected by our models, looking into their anomaly scores, tracking query counts in real-time, the number of enterprise hosts querying them, and investigating the TTL/Type fields of their corresponding responses. We make our tools and datasets available to the public to facilitate further research into this area.

II. Related Work

Malicious DNS Traffic: DNS traffic has been analyzed to identify malicious network activities [1], [16]. Studies in [17], [18] survey the available research literature on the misuse of DNS protocol for various attacks. Common malicious activities that utilize DNS include command and control (C&C) traffic tunneled over DNS channel, circulating spam messages, transferring credit card numbers (or other sensitive information), and hosting scams and phishing websites [19], [20]. Therefore, it is important to profile and detect these malicious activities. Over the last decade, there has been an increasing amount works [21]–[25] on identifying these malicious activities mostly related to C&C communications [11], [26] and phishing [27]. The primary focus of our work is to detect the queries that are involved in the exfiltration of the sensitive information to the attacker from the compromised host within the network or facilitate C&C communications from an enterprise network.

DNS Exfiltration and Tunneling: Researchers have used three categories of methods for detecting DNS exfiltration and tunneling, namely statistical-based techniques [28]–[32], supervised multi-class classification [33]–[36], and unsupervised one-class classification [37], [38].

Work in [28] proposed a method to find maximum information that can be encoded in a sub-domain portion of a DNS query name to detect whether the query contains encoded data or not. Authors used an information-theoretic approach, namely the use of Kolmogorov complexity. The authors established an upper bound on the volume of surreptitious communication by investigating inter-query time and query record type. In [29], authors employed mutual information and principle component analysis for dimensionality reduction based on consecutive DNS request and response sizes. In [30]–[32], authors have proposed DNS tunnel detection using character frequency analysis. However, the detection criteria is based on the threshold value for which attackers can go undetected easily.

In [33], [34] the authors employed a supervised learning-based model with logistic regression to classify queries as either normal or exfiltration. Buczak et al. [35] used the Random Forest algorithm for the two-class classification of benign and malicious DNS queries. Similarly, Samuel et al. [36] proposed a model to detect malicious DNS query names (generated by malware-infected machines) using Random Forest. However, attributes used in prior works to train the model are either stateful (*e.g.,* tracking the inter-arrival time of DNS packets or the frequency of query type) or require both DNS query and response messages (such as response length) [33], [35]. Also, this body of work essentially trains a model with both benign and malicious instances (*i.e.,* a two-class classifier) and the accuracy of detecting malicious queries dropped when a new family of attack is introduced (*e.g.,* model accuracy varied from 27% to 75% depending upon model parameters in [35]).

We believe that a two-class classification approach (*i.e.,* signature-based) is not sufficient for addressing new and increasing types of attacks. Also, obtaining "ground truth" on a diverse set of malicious instances in order to train the classifier is difficult [39]. The authors of [37] employed unsupervised machine learning algorithms (*i.e.,* one-class support vector machine and k-means) to detect DNS tunneling. Their primary focus was to identify infected mobile devices by using stateful attributes including the time between a DNS query and its corresponding response as well as the size of the DNS query/response of individual devices. In [40], Homem et al. benchmarked the performance of four algorithms (multi-class decision trees, support vector machine, K-nearest neighbors, and neural networks) in identifying tunneled traffic (*e.g.,* HTTP, HTTPS, and FTP) over DNS. The authors used only three attributes of DNS packets including the size of IP packet, length of query name, and entropy of query name. Similar to our approach, Nadler et al. [38] proposed an anomaly-based solution to detect low throughput data exfiltration over DNS. This work evaluated the performance of isolation forest and support vector machine learning algorithms. However, the authors maintain states of several attributes for each primary domain over the last n hours (*e.g.,* rate of A and AAAA records, the average length of query name). This makes it difficult to detect malicious queries in real-time.

To the best of our knowledge, our work is the first that presents a thorough analysis of attributes for query names from operational enterprise networks. Our focus is on attributes of fully qualified domain names that can be extracted in "real-time", without a need for states (*i.e.,* "stateless") – we assume that DNS traffic in not encrypted over TLS. We believe that our approach is fundamentally different from existing works by enabling detection of new families of DNS exfiltration without training the model by malicious instances. We look for anomalies of query names indicative of deviation from normal behavior as anomaly detection holds promise as a way of detecting new

TABLE I
SUMMARY OF OUR DATASET.

	Research	University
Total DNS packets	249M	589M
IPv4 DNS packets	206M	489M
IPv6 DNS packets	43M	100M
DNS queries	142M	341M
DNS responses	107M	248M
Total Outgoing DNS queries	86.9M	221M
Outgoing DNS queries (IPv4)	69.7M	177M
Outgoing DNS queries (IPv6)	17.2M	44M
Outgoing DNS queries (only qualified)	86M	219.5M
Unique query names (FQDN)	2.2M	6.2M
Unique primary domains	397K	1.1M

and unknown threats pattern. We also provide interesting insights into the practical considerations of such a detection scheme. Our scheme can be extended by collecting states only for those hosts that generate anomalous queries, and ultimately mitigate malicious DNS tunneling/exfiltration – such mitigation is beyond the scope of this paper.

III. DNS Queries of Enterprise Hosts: Data Collection and Attributes Extraction

In this section, we first analyze the characteristics of DNS traffic (with a specific focus on query names) collected from the border of two enterprise networks, a medium-sized research institute, and a large University campus. In both instances, the IT department of the enterprise provisioned a full mirror (both inbound and outbound) of their Internet traffic (each on a 10 Gbps interface) to our data collection system from their border routers (outside of the firewall), and we obtained appropriate ethics clearances for this study (UNSW Human Research Ethics Advisory Panel approval number HC17499, and CSIRO Data61 Ethics approval number 115/17). We extracted DNS packets from each of the enterprise Internet traffic streams in real-time by configuring rules to match incoming/outgoing IPv4 and IPv6 UDP packets on port 53 in an OpenFlow switch. The study here considers data collected over a one-week period from 30-Jul-2018 to 5-Aug-2018.

A. Our Dataset

Table I shows a summary of our dataset from each organization. We captured a total of 249M and 589M DNS packets from the border of the two networks and stored them in daily CSV files – each row in our dataset represents a timestamped DNS packet including headers and payload. The data shows that 17% of total DNS traffic is carried over IPv6 packets in both networks. Also, more than a third of our records correspond to outgoing DNS queries generated by enterprise hosts – *i.e.,* 86.9M and 221M in the Research and University networks respectively. We note that our dataset also contains queries for unqualified domain names (*i.e.,* 900K and 1.5M respectively in the Research and University networks) that are discarded in our analysis – we use the cleaned dataset.

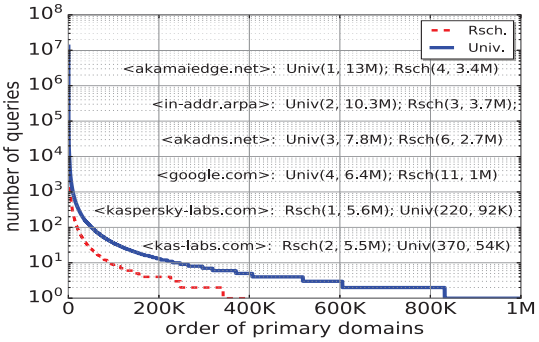


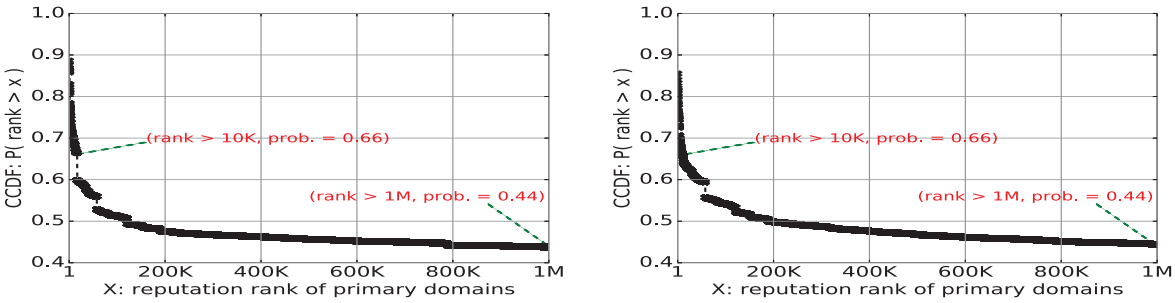
Fig. 1. Number of queries per unique primary domain, over a week (Rsch: 397K, Univ: 1.1M).

Unqualified query names contain no delimiting dots (*e.g.,* "`top_10_banks_offering_attractive_home`") or their top-level-domain is pure numeric (*e.g.,* "`129.178`"). After removing unqualified names, outgoing DNS queries in total span respectively 2.2M and 6.2M distinct fully qualified domain names (FQDN).

These FQDNs are rooted themselves in 397K and 1.1M distinct *primary domains* (*i.e.,* one level under "`com`" or "`co.uk`"). Fig. 1 shows the number of queries for each unique primary domain over the entire dataset, ordered from most queried on the left, to least queried on the right. There is a small number of domains on the left that predominate with very high query counts, followed by a long-tail of domains, all of which receive a fairly small number of queries (*i.e.,* less than 1000 over a week). It is seen that the top 4K (out of 397K) and 9K (out of 1.1M) domains respectively in the research institute and the University comprise the head in their respective curve. For example, only three domains namely "`akamaiedge.net`", "`in-addr.arpa`", and "`akadns.net`" contribute to 15% of total queries generated by University hosts. In the research network, on the other hand, top three domains of "`kaspersky-labs.com`", "`kas-labs.com`", and "`in-addr.arpa`" contribute to 17% of total queries. We note that queries for "`in-addr.arpa`" correspond to reverse DNS lookups which are commonly used by email servers to check and see if the message came from a valid server. Many email servers will reject messages from any server that does not support reverse lookups since spammers typically use invalid IP addresses.

In terms of queries "reputation", we used Majestic dataset [41] which is free and updates on a daily basis – Majestic is a reverse search engine that computes the number and strength of links to a domain (it is a measure of trust instead of traffic estimates) [42], [43]. To get a sense of reputation and probability of typical ranks, we show in Fig. 2 the complementary cumulative distribution function (CCDF) of the reputation rank for primary domains queried in both organizations. We can see that 44% of total queries, in both organizations, are not listed in the top 1M domains of Majestic domains ranking (*i.e.,* CSV dataset released on 7-Aug-2018). Also, only 32% and 34% of queries in each network are among the top 10K most popular domains. In our Majestic dataset, "`google.com`", "`facebook.com`, and "`youtube.com`" are top three ranked domains respectively.

Considering the load of DNS queries generated by en-



(a) Research institute. (b) University campus.

Fig. 2. CCDF of reputation rank: (a) Research institute, and (b) University campus.

TABLE II
A SAMPLE LIST OF MALICIOUS AND NORMAL DNS QUERIES WITH UNUSUAL LENGTH.

Query name (FQDN)	Security
6e517f3.grp10.ping.adm.cdd2e9cde9fee9cdc8.cdd0e8e9c8fce9d2e9fecdc4.c597f097ce87c5d3.ns.a23-33-37-54-deploy-akamaitechnologies.com	Malicious
708001701462b7fae70d0a2843292043 6f70797269676874.20313938352d32303031204d696372.6f736f667420436f72702e0d0a0d0a0.433a5c54454d503e.cspg.pw	Malicious
PzMnPiosOD4nOCwuOzomPS4nNjovPS8uOzsnNCstODkjOCwoMwAA.29a.de	Malicious
bwzm133h9gb3pp9s6l3mu7r73sh.arm2513pu79r9.1z19e1bgm1hwu8z6u2.9rzlkhbvi45gaag52t3rqtqd2t.p2gliv6gklwzvvlt2jp1z6li7v.avqs.mcafee.com	Normal
0.19.6ce.71c.444.25.41.0.0.0.4.27.0.0.0.0.0.0.0.0.0.9efc95e03d7f3a4ae446ecd0d049e5ae9e016ee33703c9cb3506cad4bbd98bc.b.f.00.s.sophosxl.net	Normal
p4-ces3lawazdkbw-qlrq5qalxdt7tycq-385202-i1-v6exp3.ds.metric.gstatic.com	Normal
_ldap._tcp.AWS._sites.dc._msdcs.AD.us-east-1.ec2-utilities.amazonaws.com	Normal

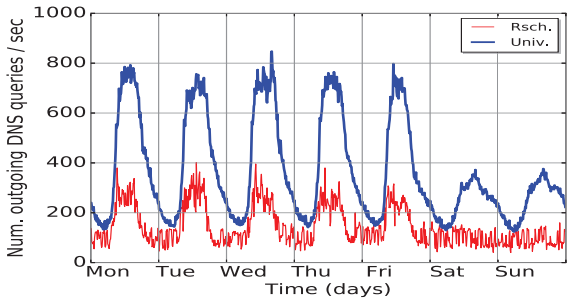
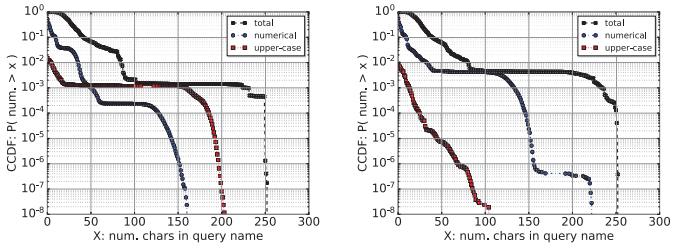


Fig. 3. Real-time number of queries.



(a) Research institute. (b) University campus.

Fig. 4. CCDF of number of characters in query name for: (a) Research institute, and (b) University campus.

terprise hosts, shown in Fig. 3, we see that the number of packet-per-sec in the research network varies between 50 to 400 depending on the day of week and peak/off-peak hours. For the University network, on the other hand, a larger variation is observed – *i.e.,* 150 to more than 800 pps.

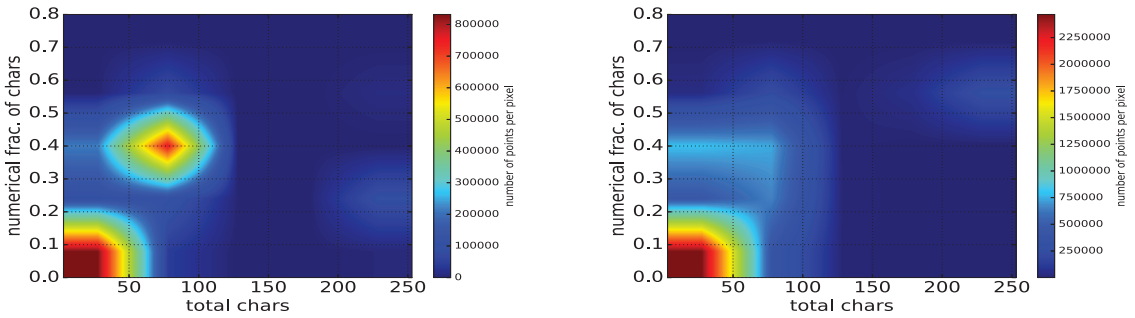
B. Query Name Attributes Engineering

We now look at the attributes of the query name (FQDN) in each DNS query generated by enterprise hosts that are relevant to differentiating benign and malicious DNS queries traffic. Our aim is to use only "Stateless" attributes which can be derived from individual DNS query packets, independent of time-series characteristics of queried domains or hosts DNS activity – there is no overhead in computing these attributes in real-time. Our attributes are inspired by various prior works (referred against each attribute).

According to RFC 1035 [44], the total length of a domain name (dots included) is restricted to 255 characters, and domain names are represented as a sequence of "labels" separated by dots. The maximum length of a label is 63 characters. It has been shown that DNS can be used for malicious purposes in the form of DNS tunneling or exfiltration in which valuable information (*e.g.,* credentials, credit card, or control messages) is embedded in the sub-domain portion of a query name. Malware applications typically embed stolen data [45] into the subdomain part of a DNS query for a domain where the name server is under control of an attacker. A DNS query for "`exfiltrated-data.example.com`" would be forwarded to the name server of "`example.com`", which would record "exfiltrated-data" and decode and decrypt the sensitive information from that subdomain field.

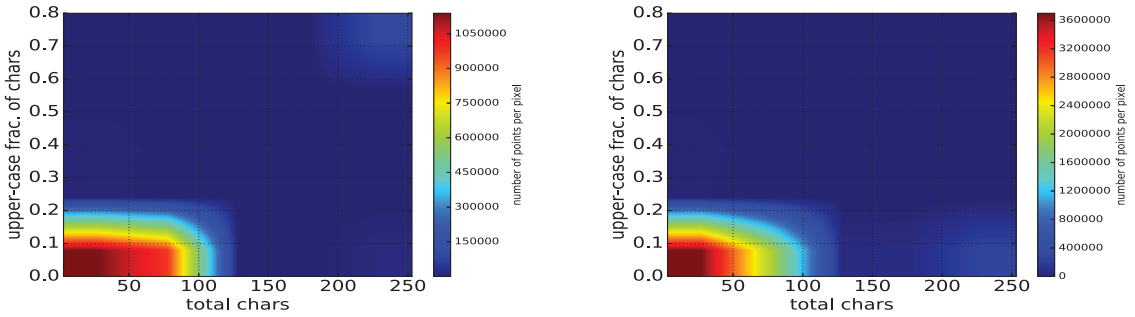
Table II lists samples of malicious [6], [15], [46] and benign query names with "unusual" length and string pattern. For example, the top two malicious query names in this list respectively contain 129 and 136 characters. We note that the sub-domain portion of these query names comprises random-looking strings with a significant number of uppercase and numerical characters, and is fairly long. For example, the second malicious query name from the top (*i.e.,* for "`cspg.pw`") contains 38 numeric characters (*i.e.,* 28%), and the third malicious query name (*i.e.,* for "`29a.de`") contains 38 numeric characters (*i.e.,* 28%) contains 23 uppercase letters (*i.e.,* 39%). Given these observations, we define our attributes by three main categories namely characters count, entropy (an indication of randomness) of string, and length of discrete labels in the query name.

1) Count of Characters: The total number of characters is an important attribute since more characters imply that the



(a) Research institute. (b) University campus.

Fig. 5. Scatter density map of numerical fraction of characters vs. total length of query name for: (a) Research institute, and (b) University campus.



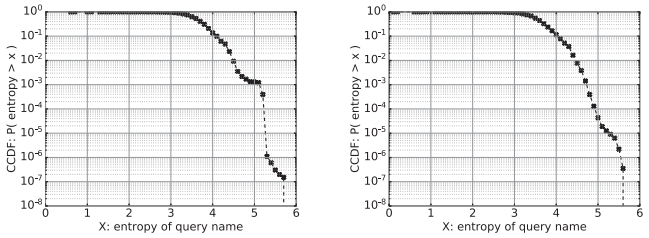
(a) Research institute. (b) University campus.

Fig. 6. Scatter density map of upper-case fraction of characters vs. total length of query name for: (a) Research institute, and (b) University campus.

query name probably carries embedded information for an outside host. In Fig. 4, we plot the distribution of character count for query names in our dataset to understand the typical value of these attributes.

Total count of characters in FQDN: [40] We can see that more than 99% of host queries in both organizations contain less than 80 characters, as shown by black cross markers in Fig. 4. Only a very small fraction of query names (*i.e.,* about 0.3%) are really long, each with more than 100 characters. It is important to note that anti-virus applications tend to exchange legitimate data (*i.e.,* for signature lookup) over DNS [38]. For example, in Table II the first two "normal" query names correspond to "McAfee" and "Sophos" anti-viruses. Interestingly, primary domains "`mcafee.com`" with 1.9M queries (average query length of 84 characters), and "`sophosxl.net`" with 145K queries (average query length of 106 characters) are among top ten frequent domains seen in our dataset from the Research institute and the University network respectively. Since the exfiltrated (or Command & Control) message is carried by the sub-domain portion of an FQDN, we use the **count of characters in sub-domain** [33] as our second attribute.

Additionally, we use the **count of uppercase characters** [34] and **count of numerical characters** [34] in a query name to determine if it is benign or malicious. This is because the fraction of uppercase and numerical characters becomes high in encrypted/ encoded data [34] – however, not all encrypted data is malicious. In Fig. 4, it is seen that only about 1% of all queries in each organization contain more than 30 numerical characters. Unsurprisingly, the upper-case character is very rare in domain names generated by hosts in both enterprise networks – at least 98% of queries contain no upper-case character, and less than 0.2% of queries contain more than 10 capitals.



(a) Research institute. (b) University campus.

Fig. 7. CCDF of entropy of query name for: (a) Research institute, and (b) University campus.

To better understand the distribution of various characters in query names, we plot the scatter density maps of total characters count versus numerical fraction in Fig. 5, , and total characters count versus uppercase fraction in Fig. 6 – dark red areas depict higher density of points and dark blue areas highlight the lower density of points. In Fig. 5, it can be seen that the numerical fraction of a query name typically stays below 20% (mostly less than 10%) when the query name has less than 60 letters (*i.e.,* dark red area on the left bottom of plots). Interestingly, for the Research Institute shown in Fig. 5(a), we observe a crowded region around 40% of numerical letters when the FQDN length is between 60 to 80 characters. In Fig. 6, we see that the fraction of uppercase letters is below 10% for short query names (*i.e.,* less than 80 characters for the Research Institute and less than 50 characters for the University), and it tends to zero when query names get longer.

2) Entropy: Random ("not-readable") sub-domains are common in DNS exfiltration/tunneling queries due to use of encryption and/or encoding [38]. **Entropy** [40] is a

TABLE III
ENTROPY VALUE FOR A SAMPLE LIST OF QUERY NAMES.

Query name (FQDN)	Entropy
www.google.com	*2.84*
202.135.201.205.23000000000012.sb-adfe2ko9.senderbase.org	*3.75*
708001701462b7fae70d0a28432920436f70797269676874.20313938352d32303031204d696372.6f736f667420436f72702e0d0a0d0a0.433a5c54454d503e.cspg.pw	**3.92**
0.19.6ce.71c.444.25.41.0.0.0.4.27.0.0.0.0.0.0.0.0.0.9efc95e03d7f3a4ae446ecd0d049e5ae9e016ee33703c9cb3506cad4bbd98bc.b.f.00.s.sophosxl.net	*3.98*
6e517f3.grp10.ping.adm.cdd2e9cde9fee9cdc8.cdd0e8e9c8fce9d2e9fecdc4.c597f097ce87c5d3.ns.a23-33-37-54-deploy-akamaitechnologies.com	**4.50**
PzMnPiosOD4nOCwuOzomPS4nNjovPS8uOzsnNCstODkjOCwoMwAA.29a.de	**4.59**
f4a55fc3f30keaayaayqivpqaggkbqggudp6hm-yacnusej1525121392-sonar.xy.fbcdn.net	*4.78*
DIYNBPRYAOK5CVUWA.ns1.logitech-usa.com	**4.86**
0ca7d.1.288.WYB52Q2ZPIU2SEUTDDDGEJDQFAO6F2C53AVC6IVAZZLR2PJHEWQWRFG6Z2NPQ3J.CQ4888.1d19d9c4.cnr.io	5.10
X2AR6GEQVHCSMXKFUNVIZU67PVMD5EF3N74E4TLOEOYK47WEXKMQ.hash.rocketeer.ct.googleapis.com	*5.27*

TABLE IV
NUMBER OF CHARACTERS IN FQDN FOR SELECTED DOMAINS IN OUR DATASET.

primary domain	# FQDN	# unique FQDN	frac. Numerical (%)	frac. Uppercase (%)	avg. Length
mcafee.com	1.9M	571K	39.4	0.31e-3	84.01
sophosxl.net	145K	41K	47.5	0.11	106.7
spotify.com	84K	819	7	0	41.7
cnr.io	121K	113K	19.97	70.08	209.8
e5.sk	66K	131	13.8	0.15	129.06

measure to determine the degree of non-readability (or strength of encryption) and uncertainty in a string. We use Shannon entropy [47] which takes a discrete random variable X as input (*i.e.,* DNS query name in our case), and mathematically is given by:

$$H(X) = -\sum_{k=1}^{N} P(x_k)\log_2 P(x_k) \quad (1)$$

where $P(x_k)$ is the probability of the k-th symbol (*i.e.,* lower-case/upper-case letter, numerical, dot, or hyphen) in the input string X containing various characters where N is the total number of unique characters. We note that only specific letters can be used in a valid DNS query name [44] (*i.e.,* 52 alphabetic and 10 numeric characters, a hyphen, and dot, thus $N = 64$). This means that the entropy value of a query name will take a value between 0 and $\log_2(64) = 6$ [48]. Table III shows the entropy value for a sample list of query names, both benign and malicious. For example, the entropy of a simple query name such as "`www.google.com`" equals to 2.84, and it gets a higher value for a more random string such as the last entry in Table III, a query for "`googleapis.com`" whose entropy value is 5.27. We also observe that the entropy value of malicious queries (highlighted in bold text) varies and is not necessarily higher than of benign queries. In Fig. 7, we plot the CCDF of entropy for all FQDNs queried by hosts of the two organizations during a week. It can be seen that the entropy value for more than 90% of query names is less than 4 in both networks, and having a entropy greater than 5 is less likely (*i.e.,* lower than 0.1%).

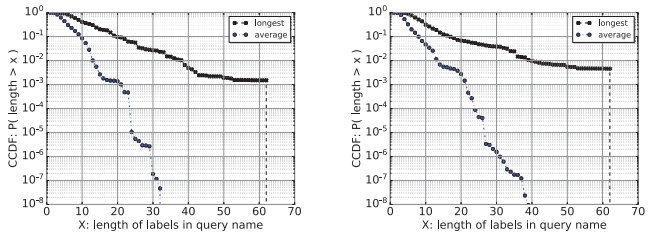
3) Labels: This category comprises two attributes of labels inside a FQDN. For example, in the query name "`www.scholar.google.com`", there are four labels separated by dots. We use the **number of labels** [35] as our sixth attribute. This is because DNS exfiltration/tunneling traffic tends to use certain patterns of labels in their query names. Table V shows the label patterns for five selected domains in our dataset from the Research institute network. We abstract a label pattern by an array (samples are shown in the second column) whose elements indicate the length (*i.e.,* character count) of the corresponding label in the query name – *e.g.,* the pattern for "`www.scholar.google.com`" is represented by (3,7,6,3). We see that queries for each of the primary domains, listed in Table V, appear in various number of patterns – the primary domain is obtained by combining the top level domain (TLD) and the second level domain (2LD) (*e.g.,* in "`www.scholar.google.com`", the primary domain is "`google.com`"). For example, the domain "`sophosxl.net`" is queried by 2208 distinct label patterns during one-week period of our dataset, and each pattern is seen in 66 queries on average. For "`e5.sk`" domain, on the other hand, we observe only 10 unique patterns, each repeats more than 600 times.

TABLE V
LABELS PATTERN IN QUERY NAMES FOR SELECTED DOMAINS.

primary domain	sample patterns	# unique patterns	avg # queries / pattern
sophosxl.net	(1, 63, 63, 36, 16, 1, 2, 1, 8, 3) (1, 63, 63, 18, 40, 1, 2, 1, 8, 3)	2208	66
mcafee.com	(3, 11, 7, 4, 4, 4, 3, 1, 26, 4, 6, 3) (3, 11, 1, 4, 4, 4, 3, 1, 26, 4, 6, 3)	316	6K
spotify.com	(48, 48, 48, 48, 16, 2, 7, 3) (23, 2, 7, 3)	43	1.9K
cnr.io	(5, 1, 3, 63, 63, 63, 30, 8, 3, 2) (5, 1, 3, 63, 63, 63, 8, 8, 3, 2)	46	2.6K
e5.sk	(63, 63, 63, 24, 1, 1, 2, 2) (63, 63, 18, 1, 1, 2, 2)	10	660

Another interesting observation is that queries for three domains namely "`sophosxl.net`", "`cnr.io`", and "`e5.sk`" have several labels with 63 characters (*i.e.,* the max limit according to RFC), whereas queries for "`spotify.com`" and "`mcafee.com`" do not use label length greater than 48 and 26 characters respectively. For our last two attributes, we use **maximum label length** [35] and **average label length** [35] in a query name. Fig. 8 depicts the CCDF of the longest and the average label length for FQDNs observed in the two organizations. It is seen that for 90% of queries their longest label does not exceed 20 characters and their average label length is 10 characters (or less) in both networks. On the other hand, only about 1% of queries have the longest label of more than 40 characters in the Research and the University networks respectively.

Summary: Our main achievement in this section is to identify and capture eight attributes (from the query name section of each outgoing DNS request packet) that collectively have strong predictive power in determining whether the query name is normal or malicious. The attributes include: **(1)** Total count of characters in FQDN, **(2)**



(a) Research institute. (b) University campus.

Fig. 8. CCDF of length of labels in query name for: (a) Research institute, and (b) University campus.

count of characters in sub-domain, **(3)** count of uppercase characters, **(4)** count of numerical characters, **(5)** entropy, **(6)** number of labels, **(7)** maximum label length, and **(8)** average label length.

IV. Detection of Anomalous Queries

We now develop a machine learning technique to determine if a DNS query of an enterprise host is normal or not (*i.e.,* "anomaly detection"). By training a model with only normal query names we aim to detect new/unknown malicious attacks (*i.e.,* anomalous queries) which can be missed by the two-class classifier. The machine is invoked with the eight attributes of each DNS query explained in the previous section. To validate the efficacy of our models, in this section we extend our dataset by including additional records (78.8 M and 217 M qualified DNS queries from the Research Institute and the University campus networks, respectively) collected over the one week period of 6-Aug-2018 to 12-Aug-2018 - a summary of this additional dataset (*i.e.,* days 8-14) is shown in Table VI. In total, we analyze 14 days worth of DNS queries from the two enterprises.

A. Machine Training

We train our anomaly detection machine with benign data from four days of our dataset – we keep the remaining ten days worth of data for testing. Ground truth of benign domains in the literature is largely drawn from highly ranked popular domains [18]. For example, Alexa top-ranked domains are commonly used – Alexa no longer publishes free top one million sites. We note that Alexa ranking is based on the browsing behavior of Internet users (*i.e.,* estimate of global traffic to a domain). As a result, some malicious domains may appear among top K Alexa domains due to a burst of requests from a high number of infected clients querying them [49]. We, therefore, use an alternative, Majestic Million [41] that releases a free dataset of top 1M domains and updates it on a daily basis. Majestic ranks sites by the number of subnets linking to that site – it is a measure of trust instead of traffic estimates [42], [43]. For the benign training instances, we only use the top 10,000 primary domains in the Majestic list. We also include FQDNs for "`sophosxl.net`" domain which is not among the top 10K Majestic dataset – the Majestic dataset is used as a reference of domain reputation to determine whether a queried domain is benign or not.

TABLE VI
Summary of additional dataset (days 8-14) used for evaluation.

	Research	**University**
Total Outgoing DNS queries	79.6M	228M
Outgoing DNS queries (IPv4)	62.6M	182M
Outgoing DNS queries (IPv6)	17.0M	46M
Outgoing DNS queries (only qualified)	78.8M	217M
Unique query names (FQDN)	2.1M	6.1M
Unique primary domains	382K	1.15M

B. Algorithms and Tuning Parameters

The objective is to maximize the detection of anomalous queries while reducing the rate of false alarms (*i.e.,* incorrectly detecting a normal query as anomalous or vice versa). Many of supervised machine-learning algorithms for detecting anomalies such as one-class SVM and Replicator Neural Network suffer from high false alarms since they are optimized for profiling the inlier behavior rather than detecting anomalies. We employ "Isolation Forest (*iForest*)" [50] which is an effective algorithm in detecting anomalous instances in high-dimensional datasets with minimal memory and time complexities.

The iForest algorithm [50] works based on the concept of isolation without employing any distance or density measure. This algorithm aims to isolate test instances by randomly selecting a feature, and then randomly selecting a split value from a range (within min and max obtained from training) values of the selected feature. Then, the score is calculated as the number of conditions (path length) to check for isolating a test instance. Note that isolating normal instances require more conditions. To avoid issues due to randomness, the process is repeated several times, and the average path length is calculated and normalized.

Algorithm Tuning: We used `scikit-learn` and its APIs, an open-source machine-learning package written in Python, to train and test our machine. We have used three tuning parameters for iForest during the training phase namely the number of trees (*n_estimators*), height limit of trees (*max_samples*), and contamination rate. We tune the value of each parameter while fixing the other two parameters and validate the accuracy of our machine for both benign and malicious instances (that we have the ground truth)in both organizations. The default value for the number of trees is 100, the height limit of trees is set to "auto" (implying 8 given the size of our dataset), and the contamination rate is 10%.

To tune the algorithm, we require ground-truth for both benign and malicious instances. Our ground-truth for benign instances are chosen based on the top 10K domains of the Majestic list (§III-A)) – we have 1.7 M instances for the research organization and 4.8 M for the university campus network). For the ground-truth of malicious instances we generated DNS exfiltration queries with our open-source tool, forked from an open-source project called "DNS Exfiltration Toolkit" (DET) [51]. We ran our tool on a machine inside the University network that exfiltrates the content

TABLE VII
DETECTION ACCURACY OF GROUND-TRUTH INSTANCES AFTER TUNING.

	Benign	Malicious
Research Institute	98.44%	95.07%
University Campus	97.99%	98.49%

TABLE VIII
ANOMALY DETECTION FOR RESEARCH INSTITUTE.

Input	Output	Days 1-4	Days 5-14
Benign domains (top 10K)	normal	98.44%	98.35%
	anomalous	**1.56%**	**1.65%**
Others (beyond top 10K)	normal	78.43%	77.35%
	anomalous	21.57%	22.65%

of a CSV file containing 1000 samples of random credit card details (obtained from [52]) to an authoritative name server under our control located in the Research network. DET employs AES-256 encryption and uses two tuning parameters namely the max length of the query name (*i.e.,* 30 to 218 characters) and the max length of labels (*i.e.,* 30 to 63 characters) to diversify our synthetic malicious queries. We generated a total of 1.4M exfiltration queries that are publicly available at [53] in the form of a CSV file.

We found that setting the number of trees equal to 2 results in high accuracy of more than 91% for benign and 63% for malicious instances – increasing this parameter does not enhance the accuracy but increases the model size and prediction time. Having fixed the number of trees to 2 and the contamination rate to 10%, we varied the height of trees from 1 to 20. The detection performance rises by increasing the height limit of trees and gets stabilized at the value of 18 with the best accuracy of more than 90% and 98% for ground-truth benign and malicious instances respectively. We then fixed the number of trees to 2 and height limit of isolation trees to 18 to quantify the impact of contamination rate. Decreasing the contamination rate from 10% to 2% improved the performance of our model for both organizations as shown in Table VII, with the accuracy of more than 97% for benign instances and more than 95% for malicious instances.

To summarize, we found the optimal value of tuning parameters equal to 2, 18, and 2% respectively for the number of trees, the height limit of trees, and the contamination rate. For optimal tuning parameters, the iForest algorithm sets the threshold value of anomaly score to 0.54, distinguishing normal and anomalous instances.

Table X shows the performance of our machine (after tunning) for selected benign instances – for cross-validation. It can be seen that the rate of false alarms is mostly less than 5% in both organizations, though we see a higher false rate (*i.e.,* more than 10%) for "`in-addr.arpa`" and "`sophosxl.net`" domains in the University network. In the next section, we will pre-filter instances for these domains that are highly trusted (*i.e.,* certainly benign) without passing them to the anomaly detection machine.

V. PERFORMANCE EVALUATION

In this section we evaluate the efficacy of our scheme by: (a) cross-validating and testing the accuracy of the trained model for benign instances and quantifying the performance in real-time on live 10 Gbps traffic streams from the two organizations, (b) testing the detection rate for malicious DNS queries that we generate using our customized tool (*i.e.,* DET [51]) and an open-source tool (*i.e.,* Iodine [54]), (c) comparing our one-class classifier with a two-class classifier, and (d) drawing insights into the top three anomalous domains for which malicious DNS queries are made in the Research and University networks.

TABLE IX
ANOMALY DETECTION FOR UNIVERSITY CAMPUS.

Input	Output	Days 1-4	Days 5-14
Benign domains (top 10K)	normal	97.99%	97.83%
	anomalous	**2.01%**	**2.17%**
Others (beyond top 10K)	normal	70.57%	63.38%
	anomalous	29.43%	36.62%

A. Performance Metrics

We begin with three performance metrics, namely accuracy, anomaly score, and responsiveness of our models.

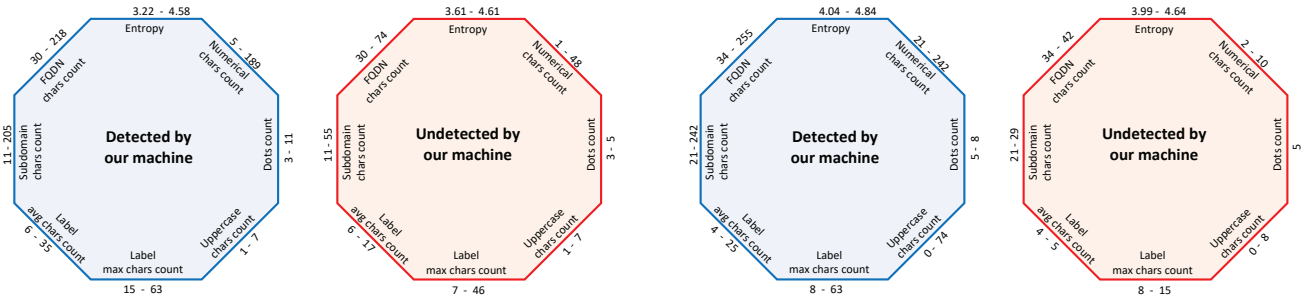
Accuracy: As mentioned in the previous section, we trained our model with benign instances from 4 days' worth of our data (*i.e.,* Days 1-4), and tested with all instances from Days 5-14 in addition to remaining instances from Days 1-4 that were not used for training (*i.e.,* "Others"). Tables VIII and IX show the rate of detection (*i.e.,* normal versus anomalous) for the benign and Others instances in the two networks – instances in the Benign category are among the top 10K of the Majestic ranking list, and instances in the Others category are beyond 10K. It can be seen that 98% of benign instances are correctly detected as normal during both cross-validation (*i.e.,* Days 1-4) and testing (*i.e.,* Days 5-14) phases. We note that our machine raises a false alarm for about 2% of benign domains, as highlighted in bold text.

To address this, we populate a whitelist of domains that are highly trusted. Our whitelist comprises only the top 100 domains from the Majestic ranking dataset (*e.g.,* "`google.com`", "`bbc.com`", "`amazonaws.com`") as well as popular legitimate (*e.g.,* "`akadns.net`", "`in-addr.arpa`", "`spotify.com`") and security services (*e.g.,* "`spamhaus.org`", "`senderbase.org`"). Note that these security services are using disposable domains (*i.e.,* "single-time use") for the purpose of signaling over DNS queries (*e.g.,* "`0.0.0.0.1.0.0.4e.135jg5e1pd7s4735ftrqweufm5.avqs.mcafee.com`" [55]).

Employing whitelisted domains would slightly enhance detection. Our refined results are shown in Tables XI and XII. We can see a slight reduction in the rate of false alarms for benign domains – it is now capped at 1.20% for both networks, as highlighted in bold text. We note there are a total of 10K (out of 923K) and 15K (out of 1.4M) false alarms for benign instances in the Research and University network respectively.

TABLE X
PERFORMANCE OF OUR MACHINE FOR TRUSTED DOMAINS.

	Research institute				University campus			
primary domain	**normal**	**anomalous**	**Avg. query length**	**false-rate (%)**	**normal**	**anomalous**	**Avg. query length**	**false-rate (%)**
`akadns.net`	2.6M	24K	38	0.91	7.6M	191K	38	2.4
`googleapis.com`	165K	1.6K	76	0.96	526K	15K	76	2.7
`gstatic.com`	207K	362	69	0.17	835K	986	76	0.11
`in-addr.arpa`	3.7M	49K	26	1.32	9.2M	1.1M	26	10.7
`mcafee.com`	1.9M	735	84	0.03	635K	13K	88	2.01
`onmicrosoft.com`	22K	1.6K	51	6.55	201K	1537	53	0.75
`senderbase.org`	1.1M	14K	66	1.32	2.2M	2816	66	0.12
`sophosxl.net`	138K	6.5K	103	4.44	2.5M	394K	119	13.7
`spamhaus.org`	12K	597	31	4.7	947K	7.7K	32	0.81
`spotify.com`	579	31	45	5.08	468K	1.2K	168	0.25
Top 100 domains (*e.g.*, `google`, `apple`)	7.9M	135K	20	1.68	24M	351K	20	1.41



(a) DET tool. (b) Iodine tool.

Fig. 9. Attributes of DNS exfiltration query names of: (a) DET tool, and (b) Iodine tool, detected vs. undetected by the University model.

TABLE XI
ANOMALY DETECTION COMBINED WITH WHITELISTING FOR RESEARCH INSTITUTE.

Input	**Output**	**Days 1-4**	**Days 5-14**
Benign domains (top 10K)	normal	98.92%	99.20%
	anomalous	**1.08%**	**0.80%**
Others (beyond top 10K)	normal	83.50%	87.48%
	anomalous	16.50%	12.52%

TABLE XII
ANOMALY DETECTION COMBINED WITH WHITELISTING FOR UNIVERSITY CAMPUS.

Input	**Output**	**Days 1-4**	**Days 5-14**
Benign domains (top 10K)	normal	98.92%	98.90%
	anomalous	**1.08%**	**1.10%**
Others (beyond top 10K)	normal	90.98%	81.74%
	anomalous	9.02%	18.26%

TABLE XIII
AVG. ANOMALY SCORE FOR RESEARCH INSTITUTE.

Input	**Output**	**Days 1-4**	**Days 5-14**
Benign domains	normal	0.36	0.36
	anomalous	0.59	0.61
Others	normal	0.44	0.43
	anomalous	0.64	0.65

Anomaly Score: Anomaly detection algorithms use this score to determine if an instance is classified as normal or anomalous. For the iForest algorithm, the anomaly score varies from 0 to 1, where 0 means purely normal and 1 indicates a definite anomaly. A value of an anomaly score of less than 0.5 is reasonable enough to be interpreted as normal [50].

TABLE XIV
AVG. ANOMALY SCORE FOR UNIVERSITY CAMPUS.

Input	**Output**	**Days 1-4**	**Days 5-14**
Benign domains	normal	0.39	0.39
	anomalous	0.57	0.58
Others	normal	0.43	0.43
	anomalous	0.63	0.62

Tables XIII and XIV show the average anomaly score (*i.e.,* normal versus anomalous) for the benign and Others instances in the two networks. It can be seen that the average anomaly score of benign instances during the cross-validation phase (*i.e.,* Days 1-4) is 0.36 and 0.39 which is well below the threshold value of 0.54 (obtained during model tuning in §IV-B) for the Research Institute and University networks respectively. Similarly, the average anomaly score of benign instances during the testing phase (*i.e.,* Days 5-14) is 0.36 and 0.40 for the Research Institute and University networks respectively.

Responsiveness: In terms of responsiveness, we have quantified the average time for extracting eight attributes and anomaly detection (via running prediction against the

TABLE XV
AVG. TIME COMPLEXITY OF OUR SCHEME.

extracting attributes	54 μsec
detecting anomalies	746 μsec
Total time per each query name	800 μsec

trained model) by testing more than 300 million DNS queries in our dataset from the two enterprise networks – our attributes extraction and anomaly detection engines run on a virtual machine using 4 CPU cores, 6GB of memory and storage of 50GB. As shown in Table XV, on average it takes 800 μsec to determine if a DNS query is normal or not. This indicates that our scheme can process approximately 1250 DNS queries per second, well above the actual rate of DNS queries in both organizations where the peak value is 800 DNS queries per second, as shown in Fig. 3.

B. Evaluating Models using Known DNS Exfiltration Data:

In this subsection, we evaluate the efficacy of our detection scheme using DNS exfiltration data (*i.e.,* ground-truth) including two large sets generated by our customized DET tool and the open-source Iodine tool, and a small set collected from publicly reported real malicious DNS queries.

Our DET Tool: We showed previously in Table VII that our models for the Research Institute and the University campus respectively were able to correctly detect 95.07% and 98.49% of exfiltration queries (generated by our DET tool) as anomalous instances.

In Fig. 9(a), we show the value of attributes for detected instances (blue octagon on the left) versus undetected instances (red octagon on the right) using the model generated from data of the university campus. Even though undetected instances were shorter both in total length and average label length, it is important to note that there is a fair overlap of value range comparing detected (*i.e.,* classified as anomalous) with undetected instances (*i.e.,* classified as normal) across all attributes, suggesting that attributes collectively would determine a fairly accurate output of our model. To explain it further, we look at the attributes of two pairs of FQDNs generated from our DET tool (one classified as normal and one classified as anomalous by the model of the research institute), as shown in Table XVI – normal and anomalous classified FQDNs are shown in bold and italic fonts respectively. We have obfuscated the actual primary domain used in the DET tool for privacy reasons. For each pair, we investigate distinguishing factors given some identical (or close) attributes, highlighted in bold in Table XVI. Starting from the top, we see six common attributes (character count, numerical character count, number of dots, maximum label length, the average length of labels, and sub-domain character count). However, two attributes namely entropy and upper-case have relatively larger values in the detected instance (*i.e.,* italic text). Moving to the second example where entropy, number of dots, and upper-case characters count are very close in two instances, the query length becomes an important factor for the model detecting or missing a malicious instance.

Iodine Tool: To further evaluate the efficacy of our scheme we used the Iodine tool [54] to generate an additional dataset of malicious DNS queries. Similar to our DET tool, we exfiltrated the same CSV file of 1000 samples of random credit card details. It took approximately 8 seconds to transfer the entire CSV file. We wrote a Python script to repeat the process with a delay (between runs) uniformly distributed between 20 and 40 seconds. We ran the script for three days. As a result, we captured more than 2.2 million unique instances – our Iodine dataset is also made publicly available [53]. Note that unlike for our customized DET tool, we only used Iodine with its default settings (*i.e.,* no variation of parameters in DNS queries). When this dataset was tested with our iForest model of the University campus, with the exception of 275 instances all others were correctly detected as anomalous. The average anomaly score was 0.86 and 0.49 for correctly and incorrectly classified instances, respectively. We also tested against the model of the Research Institute and found a small number of malicious instances (1837 out of 2.2 million) were missed – the average anomaly score was 0.67 and 0.45 (lower scores compared to the university model) for correctly and incorrectly classified instances, respectively.

In Fig. 9(b), we show the value of attributes for Iodine instances (detected versus undetected) when tested against the university in the same way as we did in Fig. 9(a) for DET instances. We can clearly see that undetected instances (red octagon on the right) have fewer numerical characters in their query name – 2 to 10 numerical chars versus 21 to 242 in detected instances. Additionally, it is observed that missed instances are relatively short (total chars count of 34-42), with a few uppercase chars (up to 8), and contain short labels (average about 5). Note that the length of DNS queries generated by Iodine is typically longer (average of 207 chars), but we intentionally diversified the query length (30 to 218 with an average of 64) in our custom DET tool which resulted in a slightly higher percentage of missed instances.

Real malicious DNS queries: Additionally, we tested 17 samples of DNS queries from known real malware reported on various forums [6], [15], [46], [56]. Top ten instances correspond to a POS malware, and the bottom eight instances were recently found as part of a new attack targeting networks of a private airline company [56]. Our trained model was able to detect all of them as anomalous instances. Table XVII gives the anomaly score of these known malicious domains, it can be seen that values are well above the average anomaly score of benign instances shown in Table XIII and XIV.

C. Comparing multi-class classifier with one-class classifier:

Existing proposals have predominantly used stateful attributes (*e.g.,* mean/variance of time-interval between a pair of DNS query/response, frequency of A, AAAA, TXT types resource records, or time between two DNS responses from

TABLE XVI
SAMPLES OF MALICIOUS QUERIES (DET) ALONG WITH THEIR ATTRIBUTES DETECTED/UNDETECTED BY THE MODEL OF THE RESEARCH INSTITUTE.

Sample FQDNs	# chars	entropy	# numeric	# dots	# upp.	max label.	avg label	# chars subdom.
s6wIrxk.36393735626336356366303833865.maliciousDomain.com	**58**	3.98	**31**	**3**	2	**30**	**13**	**39**
ClBQxLW.656661356534343938623539393265.maliciousDomain.com	**58**	4.24	**30**	**3**	6	**30**	**13**	**39**
J8tngo1.5306139323064623563663432613765643 6.maliciousDomain.com	63	**4.15**	37	**3**	**2**	35	15	44
dZrlKkg.645a726c4b6b677c217c337c217c37326138306562353738 30.333338633164363139383 7.maliciousDomain.com	101	**4.15**	64	**4**	**3**	50	19	82

TABLE XVII
ANOMALY SCORE OF QUERIES PUBLICLY REPORTED AS DNS EXFILTRATION.

Known Malicious FQDN	Anomaly Score
`708001701462b7fae70d0a28432920436f70797269676874.20313938352d32303031204d696372.6f736f667420436f72702e0d0a0d0a0.433a5c54454d503e.cspg.pw`	0.75
`9ad9ca2.grp10.tt1.dcd2fed0d2fefecdc8d2c4c8c8fecdde.e3e29f9a9ff9cbc79fdae3fcc4 d2c8c4cdd0feded295e9e9e9e9e9e9feea.e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9e9.e9e9e9e9e9e9e9e9e9e9.ns.a23-33-37-54-deploy-akamaitechnologies.com`	0.70
`PzMnPiosOD4nOCwuOzomPS4nNjovPS8uOzsnNCstODkjOCwoMwAA.29a.de`	0.68
`9ad9ca2.grp10.tt2.dcc8c8d0c8fccdd2fcd0dcdec8c8cdc8.e6dcc8c8d0c8fccdd2fcd0dcdec8c8cdc8e9dcdcdec8ded2feded0d2c8fc.ns.a23-33-37-54-deploy-akamaitechnologies.com`	0.67
`ZTEZGKDFA0KNGUCQI.ns1.logitech-usa.com`	0.65
`WQPKBPRYA0IVDUQWI.ns1.logitech-usa.com`	0.65
`QRBJBPRYA0JBKUGVI.ns1.logitech-usa.com`	0.65
`SXLXBPRYA0IVDUKTI.ns1.logitech-usa.com`	0.65
`SXLXBPRYA0IVDUKTI.ns1.logitech-usa.com`	0.65
`6e517f3.grp10.ping.adm.cdd2e9cde9fee9cdc8.cdd0e8e9c8fce9d2e9fecdc4.c597f097ce87c5d3.ns.a23-33-37-54-deploy-akamaitechnologies.com`	0.59
`RoyNGBDVIAA0.0ffice36o.com`	0.58
`iucCGJDVIBDSNF3GK000.0ffice36o.com`	0.58
`viLxGJDVIBJAIMQGQ000.0ffice36o.com`	0.58
`gLtAGJDVIAJAKZXWY000.0ffice36o.com`	0.58
`TwGHGJDVIATVNVSSA000.0ffice36o.com`	0.58
`1QMUGJDVIA3JNYQGI000.0ffice36o.com`	0.57
`t0qIGBDVIAI0.0ffice36o.com`	0.57

a given domain) along with a combination of stateless attributes to detect data theft over DNS protocol that is fundamentally different from our approach (*i.e.,* using stateless attributes only), and hence we cannot compare the performance of our model with prior research work. However, we compare the efficacy of our proposed one-class classifier (*i.e.,* iForest anomaly detector) with a two-class classification model (*i.e.,* Random Forest in [35], [57]) using the stateless attributes considered in this paper.

For two-class classification, we build a new dataset using 4-days worth of top 10K majestic million as benign (same as for the iForest model), and 1.4 million records generated by our DET tool as malicious instances. We split this dataset to 60% for training and 40% for testing. The accuracy of validation (on training data) and testing is about 99% and 97% respectively with the default parameters of Random Forest classifier. Also, we obtain the confidence-level of the Random Forest model to assess its confidence in making decision for test instances (benign or malicious). The model displays an average confidence of 99.9% for correctly classified benign instances – this measure is about 76% for misclassified benign instances. Similarly, when we present malicious queries from the DET tool, the average confidence is 99.9% and 70% for correct and incorrect predictions, respectively.

To further evaluate the efficacy of Random Forest, we tested instances from the Iodine dataset (not used in model training). The model was able to correctly detect only 0.001% of Iodine instances (2543 out of 2.2M) as malicious. Therefore, we tuned three parameters of the model, namely number of trees (1 to 200), number of selected attributes for each tree (1 to 8), and depth of tree (1 to 20). We used the same method as described in IV, and found the optimal values equal to 8, 5, and 9 respectively for the number of trees, the number of attributes, the depth. With these optimized parameters, the rate of correctly detecting Iodine queries as malicious improved to 54.8%. However, the average confidence of tuned model for correctly and incorrectly classified instances was 75% and 52% respectively – the model confidence becomes lower compared to its performance for the trained DET instances.

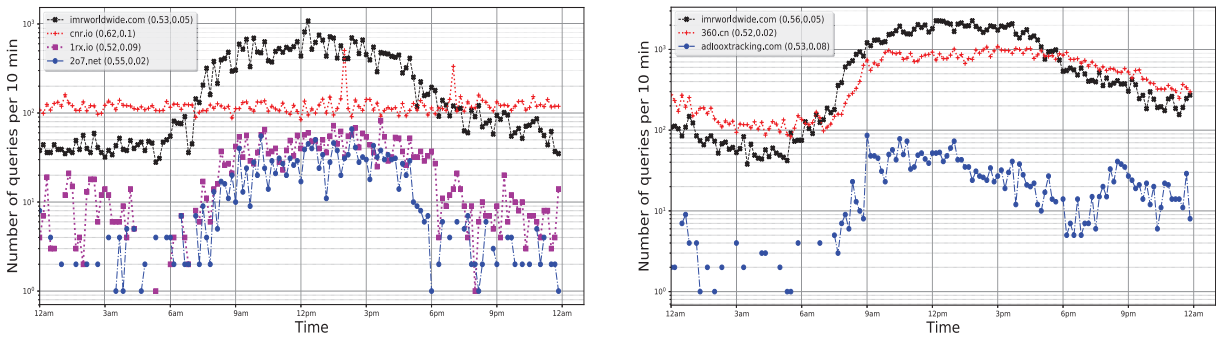
Moreover, we presented the 17 samples of publicly reported malicious DNS queries (listed in Table XVII) to the tuned Random Forest model, and found that 13 out of 17 instances were misclassified. This again proves that multi-class classifiers display poor resilience to morphed attacks that deviate from known attack signatures.

D. Malicious DNS Queries from Enterprise Networks:

We now look at real DNS queries of the two enterprises that were detected as anomalies in §V-A. Focusing on “Others" in Tables XI and XII (*i.e.,* domains that are not among top 10K Majestic ranking list), we see about 12% and 18% of instances in the Research and University networks respectively that are flagged as anomalous DNS queries. We have further analyzed these instances to gain insights into their primary domains. We found that only a few primary domains contribute more than 80% of anomalous instances in both organizations. Table V-C lists these top domains. Interestingly, domain “`imrworldwide.com`” is seen in both the Research and University networks – this domain is known spyware that tracks web activity of victim hosts [58]. We also see “`adlooxtracking.com`” the third top domain detected in the University network which is a notorious domain that redirects web users to phishing/unsafe webpages, resulting in freeware downloads [59]. In addition, our model detects suspicious domains “`2o7.net`”, “`cnr.io`”, “`1rx.io`”, and “`360.cn`” with a fairly significant number of queries. However, we cannot verify that they are malware or spyware – this may need further investigations into end-hosts that generate these queries.

TABLE XVIII
DETECTING WILD MALICIOUS DNS QUERIES FROM TWO ENTERPRISES.

	Anomalous domains	Daily avg. # queries	Daily avg. # enterprise hosts	Distinct types	Avg. TTL (sec)
Research Institute	`imrworldwide.com`	20.9K	12	A [64%], AAAA [36%]	1250
	`cnr.io`	17.4K	4	TXT [99.9%], AAAA [0.01%]	0
	`1rx.io`	2.5 K	10	AAAA [58.9%], A [41.1%]	45
	`2o7.net`	1.2K	8	A[53.4%], AAAA [46.6%]	144
University Campus	`imrworldwide.com`	80.4K	203	A[98%], AAAA [1.8%], CNAME[0.02%]	1781
	`360.cn`	69K	122	A[97.5%], AAAA [2.5%]	3139
	`adlooxtracking.com`	3K	51	A [88.1%], AAAA [11.9%]	903



(a) Research institute.

(b) University campus.

Fig. 10. Number of DNS queries for top malicious domains over a day.

Insights: To better understand these DNS queries detected as malicious, we have further analyzed their corresponding DNS responses – note that DNS responses are exclusively used in this section for drawing further insights into anomalous queries. As mentioned above, Table V-C lists the top malicious primary domains along with their statistics including daily average number of DNS queries generated for each domain, daily average number of enterprise hosts querying for each domain, distinct DNS types with their distribution, and the average of TTL values (specified in their corresponding response). For the research institute, we can see that "`imrworldwide.com`" is queried more than 20,000 times a day (on average) and only 12 unique hosts (*i.e.,* IP addresses) make these queries. Analyzing these IP addresses, we found (by reverse lookup) that five of them are recursive resolvers of the research institute – having recursive resolvers as querying hosts is also observed for other anomalous domains. Focusing on seven hosts which are regular clients, four were found actively making anomalous queries on four days, while the other three hosts do not display malicious behavior over the rest of the week of our analysis (though they are present on the network). This observation suggests that those four regular hosts are possibly infected by malware or spyware. Moreover, we found that of those four regular clients, three generated queries to all top malicious domains, except "`cnr.io`" over the entire week in the research network. Consistently generating anomalous queries over the week is seen in two regular clients for "`1rx.io`", and one regular client for "`2o7.net`".

Similarly, for the university campus, we see on average 80,000 daily queries for "`imrworldwide.com`" from an average of 203 unique enterprise hosts. By reverse lookup of host IP addresses, we found that seven of them are recursive resolvers and the remaining 196 hosts are regular clients – 150 of these clients consistently send queries for "`imrworldwide.com`" during the entire week and interestingly 130 of them fall under one subnet of size /24. Furthermore, we found that a total of 290 university hosts generate queries for at least one of the top three malicious domains (*i.e.,* "`imrworldwide.com`" or "`adlooxtracking.com`" or "`360.cn`") – of these hosts, 35 make queries for all of these top three malicious domains. By reverse lookup we discovered that 6 of them are recursive resolvers and the remaining 29 hosts are from the same subnet of size /24, indicating that this particular subnet might be infected by malware or spyware.

We further investigated the type field in DNS queries for these frequent malicious domains. "`A`"-type and "`AAAA`"-type records map domains to IPv4 and IPv6 addresses respectively. Our first observation by looking at distinct types of anomalous queries in Table V-C is that there is a much greater percentage use of IPv6 in the Research Institute than in the University network. Secondly, we observe that "`TXT`" strongly dominates the type of DNS queries for "`cnr.io`" in the Research Institute which clearly indicates a data exfiltration/tunneling over DNS [60]. Note that sophisticated attackers tend to use other types (*i.e.,* "`A`", "`AAAA`", "`CNAME`", "`NS`" and "`MX`" instead of "`TXT`") to hide their malicious activities over DNS.

DNS responses contain a Time-To-Live (TTL) field in seconds, indicating the duration for which a DNS resource record is to be cached on the host machine. According to RFC 1033 [61], it is important to set an appropriate TTL value, since very low values result in overloading the DNS server and very high values may limit the flexibility of changing resource records in real-time. According to RFC 1912 [62], it is recommended to set the TTL value between

one to five days. But, CDN (Content Distribution Network) services tend to use smaller TTLs for fast reaction to dynamic resource changes. Unfortunately, malicious entities also use small TTLs for minimizing their footprints and becoming more resistant against DNS blacklisting [22]. In Table V-C, we compute the average TTL for each of the top malicious domains. We observe that malicious domains use relatively smaller TTLs (*i.e.,* less than an hour), for example it is set to 0 in all of the DNS responses for "`cnr.io`". Another example is "`1rx.io`" for which the average TTL is 45 seconds.

We plot in Fig. 10 the query count (computed every 10 minutes) of top anomalous primary domains on day 6-Aug-2018, as an example – missing points in this figure correspond to zero query count over those 10-min epochs. Note that the mean and standard deviation of the anomaly score is shown next to each domain name in the legend. Our first observation is that the query count for all primary domains is higher during working hours (*i.e.,* increasing an order of magnitude at about 8 am, staying at a certain level, and falling back at about 5 pm), though the primary domain "`cnr.io`" in Fig. 10(a) displays a fairly consistent pattern of query count over a day (except one spike at around 2 pm).

Finally, looking at the anomaly score of queries for these selected malicious domains (as shown in the legend of Fig. 10), `cnr.io` domain has the largest mean value 0.62 (the closer to 1 means more anomalous). The average score for other malicious domains in both networks varies between 0.52 to 0.56 which is well above the average score for benign instances (*i.e.,* less than 0.40) reported in Table XIII and XIV.

VI. Conclusion

Enterprise networks are potential targets of cyber-attackers for stealing valuable and sensitive data over DNS channels. We have developed and validated a mechanism for real-time detection of DNS exfiltration and tunneling from enterprise networks. By analyzing DNS traffic from two organizations we have identified attributes of DNS query names that can be extracted efficiently in real-time distinguishing legitimate from malicious queries. We then developed, tuned and trained a machine-learning algorithm to detect anomalies in DNS queries using a known dataset of benign domains as ground truth. Lastly, we evaluated the efficacy of our scheme on live 10 Gbps traffic streams from the borders of two enterprise campus networks by injecting more than three million malicious DNS queries via DET and Iodine tools – our tools and datasets are publicly available. We showed that our solution outperforms the two-class classifier in detecting new malicious DNS queries. We have drawn insights into anomalous DNS queries by their anomaly scores, the trace of query count over time, enterprise hosts querying them, and TTL and Type fields of their corresponding responses.

VII. Acknowledgments

This work was completed in collaboration with the Australian Defence Science and Technology Group.

References

[1] J. Ahmed *et al.*, "Real-Time Detection of DNS Exfiltration and Tunneling from Enterprise Networks," in *Proc. Integrated Network and Service Management (IM)*, Washington, DC, USA, Apr 2019, pp. 649–653.

[2] Efficient iP. (2017) The Global DNS Threat Survey. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2HWYZ9k

[3] A. Greenberg. (2014) DNS attacks putting organizations at risk, survey finds. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2X7Az5Z

[4] B. Krebs. (2014) Deconstructing the 2014 Sally Beauty Breach. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/1FTBvKZ

[5] P. Rascagneres. (2014) New FrameworkPOS variant exfiltrates data via DNS requests. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2VpfVd5

[6] a. vault. (2015) BernhardPOS - New POS Malware. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/31kVKzZ

[7] C. Lynch *et al.* (2016) MULTIGRAIN – Point of Sale Attackers Make an Unhealthy Addition to the Pantry. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/1pgvJxn

[8] A. Shulmin *et al.* (2017) Use of DNS Tunneling for C&C Communications. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2SwS1KY

[9] R. Neumann *et al.* (2018) UDPoS – exfiltrating credit card data via DNS. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2wLapq4

[10] R. J. Dietrich. (2011) Feederbot - a bot using dns as carrier for its cnc. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/30ZZ3MN

[11] C. J. Dietrich *et al.*, "On Botnets that use DNS for Command and Control," in *Proc. IEEE Computer Network Defense*, Gothenburg, Sweden, Sep 2011, pp. 9–16.

[12] C. Mullaney. (2011) Morto worm sets a (DNS) record. Accessed on 10.06.2019. [Online]. Available: https://symc.ly/2JXKfZS

[13] T. Spring. (2016) Wekby apt gang using DNS tunneling for command and control. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/1ONTN1s

[14] S. Kathuria. (2015) DNS Firewall is not a Next Generation Firewall. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/1NbDIRp

[15] E. Brumaghin *et al.* (2017) Covert Channels and Poor Decisions: The Tale of DNSMessenger. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2R6OSEM

[16] M. Lyu *et al.*, "Mapping an Enterprise Network by Analyzing DNS Traffic," in *Proc. Passive and Active Measurement (PAM)*, Puerto Varas, Chile, Mar 2019, pp. 129–144.

[17] S. Zander *et al.*, "A Survey of Covert Channels and Countermeasures in Computer Network Protocols," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 3, pp. 44–57, 2007.

[18] Y. Zhauniarovich *et al.*, "A Survey on Malicious Domains Detection Through DNS Data Analysis," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 67, 2018.

[19] M. Antonakakis *et al.*, "Detecting malware domains at the upper dns hierarchy." in *Proc. USENIX Security Symposium*, San Francisco, CA, USA, Aug. 2011, pp. 27–27.

[20] ——, "From throw-away traffic to bots: Detecting the rise of dga-based malware." in *Proc. USENIX Security Symposium*, Bellevue, WA, Aug. 2012, pp. 24–24.

[21] S. Hao *et al.*, "An Internet-Wide View into DNS Lookup Patterns," VeriSign Labs, School of Computer Science, Georgia Tech, Tech. Rep., 2010.

[22] L. Bilge *et al.*, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis." in *Proc. USENIX Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, Feb 2011, pp. 1–17.

[23] M. Antonakakis *et al.*, "Building a Dynamic Reputation System for DNS," in *Proc. USENIX Security Symposium*, Washington, DC, USA, Aug 2010, pp. 18–18.

[24] S. Hao *et al.*, "Monitoring the Initial DNS Behavior of Malicious Domains," in *Proc. ACM SIGCOMM Internet measurement conference (IMC)*, Berlin, Germany, Oct 2011, pp. 269–278.

[25] M. Mowbray and J. Hagen, "Finding domain-generation algorithms by looking at length distribution," in *Proc. IEEE software reliability engineering workshops*, 2014, pp. 395–400.

[26] M. Feily *et al.*, "A survey of botnet and botnet detection," in *Proc. ACM Emerging Security Information, Systems and Technologies*, Athens, Glyfada, Greece, Jun 2009, pp. 268–273.

[27] M. Khonji *et al.*, "Phishing detection: a literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

[28] V. Paxson *et al.*, "Practical Comprehensive Bounds on Surreptitious Communication over DNS." in *Proc. USENIX Security Symposium*, Washington, DC, USA, Aug. 2013, pp. 17–32.

[29] E. Cambiaso *et al.*, "Feature transformation and Mutual Information for DNS tunneling analysis," in *Proc. IEEE Ubiquitous and Future Networks (ICUFN)*, Vienna, Austria, Mar 2016, pp. 957–959.

[30] K. Born *et al.*, "NgViz: Detecting DNS Tunnels Through N-gram Visualization and Quantitative Analysis," in *Proc. ACM Cyber Security and Information Intelligence Research Workshop(CSIIRW)*, Oak Ridge, Tennessee, USA, Apr 2010.

[31] ——, "Detecting DNS Tunnels using Character Frequency Analysis," in *Proc. Annual Security Conference*, Las Vegas, USA, Apr 2010.

[32] C. Qi *et al.*, "A Bigram based Real Time DNS Tunnel Detection Approach," *Procedia Computer Science*, vol. 17, pp. 852–860, 2013.

[33] J. Liu *et al.*, "Detecting DNS Tunnel through Binary-Classification Based on Behavior Features," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, 2017, pp. 339–346.

[34] A. Das *et al.*, "Detection of Exfiltration and Tunneling over DNS," in *Proc. IEEE Machine Learning and Applications (ICMLA)*, Cancun, Mexico, Dec. 2017, pp. 737–742.

[35] A. L. Buczak *et al.*, "Detection of tunnels in PCAP data by random forests," in *Proc. ACM Cyber and Information Security Research (CISRC)*, Oak Ridge, TN, USA, Apr 2016, pp. 1–4.

[36] S. Schüppen *et al.*, "FANCI: Feature-based Automated NXDomain Classification and Intelligence," in *Proc. USENIX Security Symposium*, Baltimore, MD, USA, Aug 2018, pp. 1165–1181.

[37] P. Engelstad *et al.*, "Detection of DNS tunneling in mobile networks using machine learning," in *Proc. Information Science and Applications*. Springer, 2017, pp. 221–230.

[38] A. Nadler *et al.*, "Detection of Malicious and Low Throughput Data Exfiltration Over the DNS Protocol," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1709.08395

[39] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proc IEEE Security and Privacy*, Berkeley, CA, USA, May 2010, pp. 305–316.

[40] I. Homem *et al.*, "Harnessing predictive models for assisting network forensic investigations of DNS tunnels," in *Proc. ADFSL Digital Forensics, Security and Law*, 2017, p. 79.

[41] T. M. Million. (2018) Top 1 million website in the world. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2gbMh5P

[42] W. Rweyemamu *et al.*, "Clustering and the Weekend Effect: Recommendations for the Use of Top Domain Lists in Security Research," in *Proc. Passive and Active Measurement (PAM)*, Puerto Varas, Chile, Mar 2019, pp. 161–177.

[43] S. Kelkar *et al.*, "Analyzing HTTP-Based Information Exfiltration of Malicious Android Applications," in *Proc. IEEE TrustCom/BigDataSE*, New York, USA, Aug 2018, pp. 1642–1645.

[44] P. Mockapetris, "RFC 1035 Domain Names - Implementation and Specification," *Internet Engineering Task Force*, 1987.

[45] M. Lee and J. Schultz. (2016) Detecting DNS Data Exfiltration. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2WbFI7Z

[46] L. Mendieta. (2016) Three Month FrameworkPOS Malware Campaign Nabs 43,000 Credit Cards from Point of Sale Systems. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2BSw166

[47] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[48] B. J. Brewer, "Computing Entropies with Nested Sampling," *Entropy*, vol. 19, no. 8, p. 422, 2017.

[49] Y. Zhauniarovich *et al.*, "A Survey on Malicious Domains Detection Through DNS Data Analysis," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, 2018.

[50] F. T. Liu *et al.*, "Isolation forest," in *Proc. IEEE Data Mining*, Pisa, Italy, Dec 2008, pp. 413–422.

[51] R. Qasim. (2018) DET (extensible) Data Exfiltration Toolkit. Accessed on 10.06.2019. [Online]. Available: https://github.com/qasimraz/DET

[52] (2018) MasterCard Credit Card Generator. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2QLHBHo

[53] (2018) DNS Exfiltration Dataset. Accessed on 10.06.2019. [Online]. Available: https://nozzle-data.sdn.unsw.edu.au/

[54] (2019) kryo.se: iodine (IP-over-DNS, IPv4 over DNS tunnel). Accessed on 10.06.2019. [Online]. Available: https://code.kryo.se/iodine/

[55] Y. Chen *et al.*, "DNS noise: Measuring the pervasiveness of disposable domains in modern DNS traffic," in *Proc. Dependable Systems and Networks (DSN)*, Atlanta, GA, USA, Jun 2014, pp. 598–609.

[56] W. Mercer *et al.* (2018) DNSpionage Campaign Targets Middle East. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2BBXKZI

[57] F. Allard *et al.*, "Tunneling activities detection using machine learning techniques," *Journal of Telecommunications and Information Technology*, pp. 37–42, 2011.

[58] (2019) Easy removal method of Imrworldwide.com infection. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/31hOTY6

[59] (2019) How to remove adlooxtracking. Accessed on 10.06.2019. [Online]. Available: https://bit.ly/2XoKNet

[60] B. Yu *et al.*, "Behavior Analysis based DNS Tunneling Detection and Classification with Big Data Technologies," in *Proc. International Conference on Internet of Things and Big Data*, Rome, Italy, 2016, pp. 284–290.

[61] M. Lotter, "RFC 1033: Domain Administrators operations guide," *International Engineering Task Force*, 1987.

[62] D. Barr, "RFC 1912: Common DNS operational and configuration errors," *The Pennsylvania State University, Pennsylvania*, 1996.

Jawad Ahmed received his MS in Electrical Engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan in 2016, and BS in Electrical (Telecom) Engineering from COMSATS Institute of Information Technology, Islamabad, Pakistan in 2011. He worked as a lab engineer at NUST from 2012 to 2017. He is currently pursuing his Ph.D. from University of New South Wales (UNSW) Sydney. His major research interests include Software Defined Networking, Data analytics and Cybersecurity.

Hassan Habibi Gharakheili received his B.Sc. and M.Sc. degrees of Electrical Engineering from the Sharif University of Technology in Tehran, Iran in 2001 and 2004 respectively, and his Ph.D. in Electrical Engineering and Telecommunications from UNSW in Sydney, Australia in 2015. He is currently a lecturer at UNSW Sydney. His research interests include programmable networks, learning-based networked systems, and data analytics in computer systems.

Qasim Raza received his B.Sc Electrical Engineering from the University of Buffalo, NY in 2016 and his M.Eng in Telecommunications from University of New South Wales, Sydney, Australia in 2018. He is currently a network software engineer at Lumina Networks. His research interests include software-defined networking and network automation.

Craig Russell received his Ph.D. in Applied Mathematics from Macquarie University, Sydney in 1997. He is currently a Principal Research Engineer at CSIRO Data61 and has previously held commercial roles in the telecommunications and software industries. He has design, implementation and operational experience in a wide range of advanced telecommunications equipment and protocols as well as experience in developing software applications. His research interests are in software-defined networking and the application of machine learning techniques to solve problems in network security.

Vijay Sivaraman received his B. Tech. from the Indian Institute of Technology in Delhi, India, in 1994, his M.S. from North Carolina State University in 1996, and his Ph.D. from the University of California at Los Angeles in 2000. He has worked at Bell-Labs as a student Fellow, in a silicon valley start-up manufacturing optical switch-routers, and as a Senior Research Engineer at the CSIRO in Australia. He is now a Professor at the University of New South Wales in Sydney, Australia. His research interests include Software Defined Networking, network architectures, and cyber-security particularly for IoT networks.