eyeDNS: Monitoring a University Campus Network

Chandan Chowdhury
Dept. of Computer Science
Kansas State University
Manhattan, KS, USA
chandanchowdhury@ksu.edu

Dalton A. Hahn
Dept. of Computer Science
Kansas State University
Manhattan, KS, USA
hahnd11@ksu.edu

Matthew R. French
Dept. of Computer Science
Kansas State University
Manhattan, KS, USA
matthewf@ksu.edu

Eugene Y. Vassermann
Dept. of Computer Science
Kansas State University
Manhattan, KS, USA
eyv@ksu.edu

Pratyusa K. Manadhata
Micro Focus
Sunnyvale, CA, USA
manadhata@alumni.cmu.edu

Alexandru G. Bardas
EECS Department
University of Kansas
Lawrence, KS, USA
alexbardas@ku.edu

***Abstract*—The Domain Name System (DNS) is responsible for mapping human readable domain names to internet protocol (IP) addresses. DNS is a ubiquitous part of internet and intranet communication, making it a convenient and comprehensive source for data to infer network health, performance, and security. A victim of its own success, monitoring real-time DNS traffic is a challenge due to sheer volume: huge amounts of DNS packets flow through a typical enterprise in a single day. In this paper, we describe *eyeDNS*, a scalable and extensible system for near real-time aggregation, storage, analysis, and visualization of DNS traffic collected by a hardware back-end. We report on eyeDNS's deployment and data collection on a large public university's network over a timeframe of 15 months. Moreover, we leveraged data from the following 6 months to validate findings made during the initial timeframe. With fast query response, aggregation, and visualization of DNS data, eyeDNS helped identify instances of anomalous network use, malware-specific behaviors, and scamming activities. eyeDNS is currently being used by the university's security personnel and has demonstrated its effectiveness in extracting trends and outliers from large volumes of DNS data collected from a diverse environment, where even commercial tools struggle to provide timely and actionable analysis.**

***Index Terms*—DNS, packet filtering, network traffic**

I. Introduction

Considering its magnitude and impact, the modern internet can be viewed as a utility, similar to the telephone system. While phone books are used to navigate the telephone system, the Domain Name System (DNS) is the internet's equivalent of a phone book, designed to provide a consistent name space used for referring to resources [1]. Without such a system it would be extremely difficult to navigate the modern internet with its more than a billion websites [2]. This analogy can also be extended to enterprise networks which combine private intranets with public network segments.

Disruptions and attacks on a network's DNS infrastructure, e.g., DNS zone poisoning [3] and using DNS traffic to target other entities [4], are very disruptive to a significant number of users on the network. For example, in October 2016, an attack on a global DNS service provider, *Dyn*, rendered almost half of the world's top internet services inaccessible for hours in North America and Europe [5]. Similarly, Malwares and bots use Domain Generation Algorithms (DGA) to hide their communication with remote command and control (C&C) under legitimate DNS traffic [6].

Monitoring and analyzing DNS traffic provides visibility into the communications of a network's internal parties, including malicious communication. The research community has been studying the DNS ecosystem and analyzing DNS traffic for a long time. Work in this area has focused on detecting various security issues, misconfigurations, and misuse/abuse [3, 6–9]. Though DNS servers can log DNS traffic, it is often disabled for performance reasons, or, only DNS queries are logged, not responses. DNS responses may have more security-relevant information than queries. For example, attackers may hijack a benign domain and point it to an IP address they control [10]. We need to collect DNS responses to be able to detect the malicious mapping.

The overarching goal of this work is to demonstrate the flexibility and power of DNS traffic analysis in a large US university campus network. We provide visibility into the campus network by analyzing its DNS traffic in near real-time. Our campus network has over 218,000 registered users and more than 42,000 hosts. To monitor the resulting large volume of DNS traffic, we developed a pipeline that leverages a DNS packet capture back-end to collect campus DNS traffic and constructed a software framework, *eyeDNS*, to store, aggregate, visualize, and analyze the collected data.

Challenges. A US research university campus network is very dynamic, open, and diverse by its nature with multiple departmental networks, off-campus students, residence halls, contractors, financial services, and research groups across various disciplines. Although a generic campus wide security policy is followed, the "open" nature of an academic network makes it difficult to apply a single security policy across the campus. For instance, our campus network hosts legacy software and hardware components. Parts of the network still use old operating systems, e.g., Windows XP, and old networking hardware, e.g., hubs. Even though the campus

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

network employs firewalls, log aggregation systems, and deep-packet inspectors, their functionality is limited on older network segments. Our DNS framework complements these tools and provides campus networking and security personnel additional visibility into different segments of the network. This additional visibility led to identifying domain names used in malware campaigns, uncovering signs of scams, and near real-time tracking of anomalous behaviors on the network.

This paper presents our observations and findings after storing, processing and analyzing DNS data from an operational campus network for 15 months (January 1, 2016 - March 31, 2017). Moreover, we utilized 6 months of subsequent data to validate some of the original findings. To the best of our knowledge, this is the first such study and we hope that our findings will be relevant to other similar environments.

The paper makes the following key contributions.

1) We designed and built a framework to store, visualize, and analyze information about DNS queries & responses.
2) We demonstrate that DNS data provides visibility into what is happening on a campus network of a US public research university.
3) We show that eyeDNS is a flexible and extensible framework that can interact with other security (DNS) frameworks/tools to uncover and validate various information.

II. Related Work

The research community has extensively studied the DNS ecosystem using a variety of techniques and has focused on a wide variety of topics such as DNS performance analysis, WHOIS registration data analysis [11], and vulnerabilities in the dynamic DNS update procedures [3]. Researchers have proposed multiple systems to distinguish malicious domain names from benign domain names [7, 12–14]. These systems rely on the intuition that the characteristics, e.g., registration information and access pattern, of malicious domain names differ from benign domain names. Hence they collect passive DNS information, compute several features for each domain name, and then use automated classification and clustering approaches to identify and label malicious domains. These techniques are complementary to our work and can be leveraged on the data our framework is collecting and storing.

DNS analysis is especially effective in identifying so called DGA malware. Malware often locate with their C&C servers through DNS queries and then contact them to receive instructions. Modern malware use DGAs to generate unique domains in order to evade detection by domain blacklisting [7]. However, differences in behavior and syntax of algorithmically generated domains and benign domains can be exploited to identify DGA-based domains [6, 15–17]. Our eyeDNS framework expands the work of Plohmann et al. [6] to detect and report algorithmically generated domains in our campus network. Also, the flexibility of eyeDNS allows us to leverage other systems' DGA-based domain detection abilities and detect other malicious activities like online scams such as [18].

The DNS Statistics Collector (DSC) provided by DNS-OARC [19] is another framework for analyzing DNS data. It can capture DNS data, aggregate counts based on record types, response codes, query names and even TLDs. However, to the best of our knowledge DSC lacks features such as a built-in check against whitelist and blacklist or providing host and domain level details in an automated fashion.

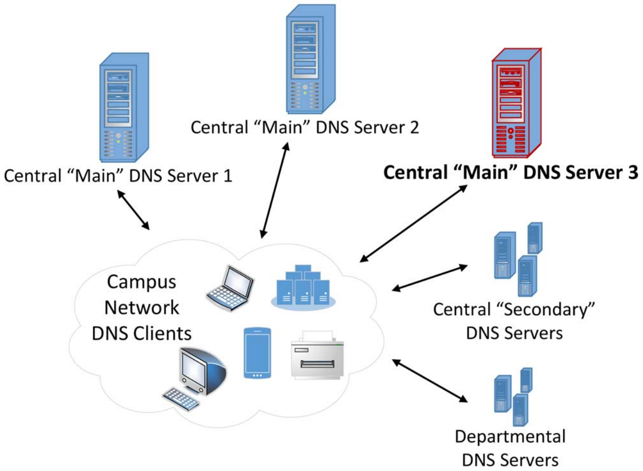
Central "Main" DNS Server 1
Central "Main" DNS Server 2
Central "Main" DNS Server 3
Campus Network DNS Clients
Central "Secondary" DNS Servers
Departmental DNS Servers

Fig. 1: Campus DNS Infrastructure Overview – The infrastructure is highly distributed with servers located in different data centers. We are monitoring Central "Main" DNS Server 3.

III. Data Collection

Over 15 months we have collected and analyzed DNS traffic from our university campus network. As of April 2017, the university has around 24,000 enrolled students, and the campus network had over 218,000 registered users (students, faculty, administrative personnel, etc.) and more than 42,000 hosts (devices) per day may be connected to it. The public portion of the network contains a publicly routable */16* block of IP addresses, with various VLANs subdividing the block into various network segments. The wireless segment uses private, non-internet-routable, IP address ranges, for registered and guest users, VPN clients, and the residence halls.

A. Data Source

Our network uses a DNS infrastructure distributed between central and departmental servers (see Figure 1). The three main recursive DNS servers are located in different data centers across campus. Usually, the IP addresses of these servers are "pushed" through DHCP to the hosts. Depending on the network segment a host is connected to, the DNS servers may be pushed in a different order or even changed to secondary or departmental servers. Secondary central servers serve (usually) as backup while departmental servers are managed by departments (authoritative for the department's infrastructure). These departmental servers leverage the central DNS servers to answer external DNS queries (central DNS servers drop DNS queries originating from non-campus IP range).

Data Capture Mechanism. The DNS capture box (see Figure 2) is a standalone host that uses a network card with a high-speed network accelerator (part of the packet capture module) enabling real-time network monitoring.

The capture box has a filtering module, to differentiate benign and malicious traffic using whitelists and blacklists.

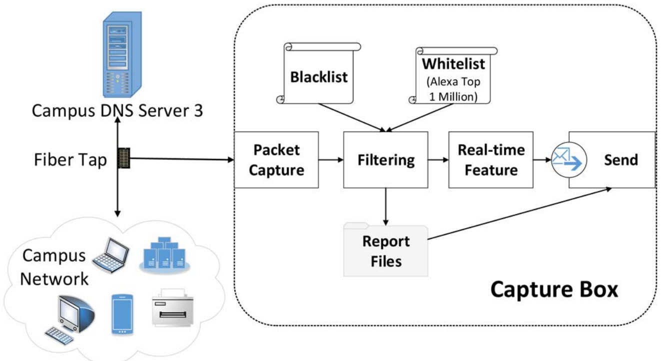

Fig. 2: Capture Box Close-up – This host is responsible for processing raw DNS traffic and sending the generated DNS data to eyeDNS.

The box is connected to a passive fiber TAP which mirrors all traffic to and from the campus DNS server 3 to the capture box. Furthermore, the capture box has a “real-time feature” that enables it to generate instant messages (in syslog format [20]) with information about potentially interesting DNS packets (e.g., NXDOMAIN responses). Syslog messages and reports are automatically sent using the delivery module to the eyeDNS framework leveraging a set of scheduled tasks.

B. *Whitelisting and Blacklisting*

At its simplest, a blacklist is a list of potentially bad domains, while a whitelist is a list of good domains. A DNS query packet with a single domain name (qname) will first be checked against a blacklist. If the qname or any of its ancestors is present in the blacklist, we classify the packet as potentially malicious (domain names are hierarchical; thus, for example, `x.com` is an ancestor of `a.x.com`). The intuition is that if a domain is labeled malicious, then all of its subdomains are also likely to be labeled malicious. However, if the qname is not malicious, and either the qname or any of its ancestor domains is present in the whitelist, then we label the packet as benign. The intuition is that if a domain is benign, then all of its subdomains are benign, unless a subdomain is specifically known to be malicious. In case a packet is neither benign nor potentially malicious, then we classify it as unknown (graylisted). Moreover, if a request packet has multiple qnames, then we consider all qname fields; if all are benign, then the packet is identified as benign.

On the other hand, response packets are more complex since they contain multiple resource records of different types and possibly originating from multiple external nameservers. For these packets we consider multiple relevant fields such as qnames, IPv4 addresses, IPv6 addresses, CNAMEs, TXT records, and MX names. If at least one field in a packet is blacklisted, then we label the packet as potentially malicious. If all the fields in the packet are whitelisted, then we label the packet as benign. Otherwise, we mark the packet as graylisted.

In this work, we used the Alexa Top One Million list of domains as a whitelist to classify benign domains. This list has been used extensively in previous works related to DNS [6, 7, 13, 14, 21–26]. The blacklist was being supplied free of cost by a commercial vendor for research purpose, however, since October 2016 we have not received updates as the part of business was sold by the vendor. While eyeDNS has the capability to leverage any white- or blacklist provided to it in order to classify domains, in our experience, we found commercial blacklists to be unreliable and limited, as also noted in previous work by [6, 7, 9, 13, 14, 17, 18].

C. *DNS Data*

The DNS capture box generates a set of report files every hour. These reports contain overall hourly statistics (e.g., number of DNS packets, DNS queries, responses that were processed, number of packets that are on the whitelist) as well as specific information about hosts and domain names (e.g., which hosts queried a specific domain and the number of times this happened). These hourly report files allow the eyeDNS framework to provide near real-time data for use in statistics analysis and anomaly detection.

Under normal circumstances the eyeDNS framework receives 24 sets of *report files* a day. Our “day” starts at 11:00PM and ends the next day at 11:00PM. During the analyzed timeframe (for accuracy purposes: December 31, 2015 11:00PM to March 31, 2017 11:00PM), the eyeDNS framework received 10,805 sets (batches) of report files out of 10,944 total possible batches (i.e., 98.73% generated/received report sets). The missing report sets were not generated due to outages or capture box operating system updates and restarts.

D. *Visualization Framework – eyeDNS*

eyeDNS is a web-based framework that stores, aggregates, and visualizes large amounts of campus DNS data. Figure 3 pictures the main modules of the eyeDNS framework: receiver-parser module, database, and web portal.

Implementation. The receiver-parser module includes a series of Python scripts that process the DNS data received from the capture box and inserts it in the database. We are using MongoDB as our database and store the database content on a Network Area Storage (NAS) device. The web portal uses PHP to query the database which returns the data in JSON format. Custom JavaScript with various libraries, CSS and HTML5 are mainly used on the client side to display information based on user requests and filter parameters (date range, host IP, domain name etc.).

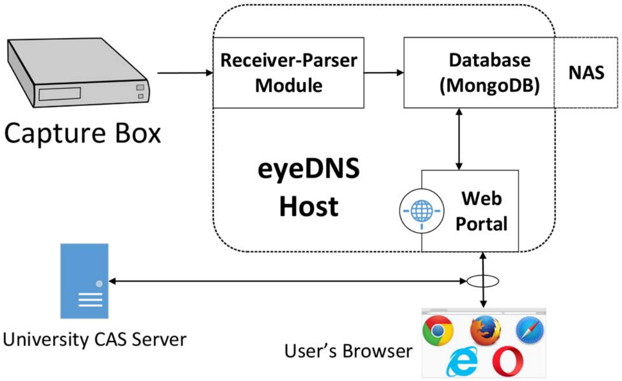

Fig. 3: eyeDNS Close-up – stores, aggregates, and visualizes large amounts of campus DNS data. Arrows indicate data flow directions.

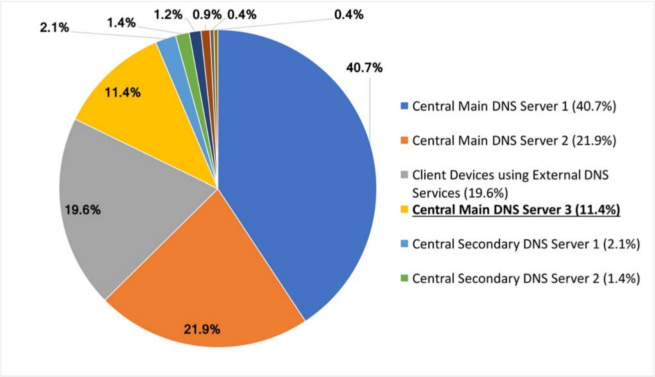

Fig. 4: External DNS Connections – Percentages distribution for "over-the-border" DNS connections. Data collection took place at the "Central Main DNS Server 3", identified in bold.

Functionality. eyeDNS provides multiple ways to examine and filter the collected data including statistical reports across a timeframe, visualizations of the queries and responses over time, and averages for various types of DNS queries and responses. The framework also allows users to examine query results of a particular host in a time range, as well as examine the results for a particular domain in a time range. Moreover, eyeDNS presents a range of statistics and information for each domain and host on the network. Also, on the statistics page, queries that generate invalid domain responses are recorded along with statistics about the host that created the query. Overall, eyeDNS provides a lens into very specific details of the collected data.

E. Statistics of Collected Data

Our dataset consists of DNS data computed from DNS query and response packets generated on our campus network. Over the course of 15 months (January 1, 2016 – March 31, 2017), we have been collecting these data points using a capture box hardware back-end. Our dataset was assembled from processing 640,891,500 DNS query packets and 696,879,237 DNS response packets. The capture box parses the collected packets, compares the DNS records to the built-in lists' entries (blacklist, whitelist) and produces the DNS data (report files and instant messages) sent to eyeDNS. The eyeDNS framework processes the received data, stores it in the database, and aggregates it for various visualization and analysis purposes (statistics of data).

Coverage. Our data was collected in a diverse and dynamic environment, including wireless and VPN connections, traffic to specialty sites covering numerous fields of research, and traffic to multiple international resources, as the university hosts students of many nationalities on campus.

Based on the layout of the network and the output from other campus network monitoring tools, Central "Main" DNS Server 3 (the server we are monitoring) handles between 11% and 30% of the DNS load targeted at the central DNS infrastructure. Figure 4 shows a snapshot of the campus DNS server load distribution. The chart, while helpful in visualizing coverage, is not representative of the continuing usage fraction of the monitored DNS server – the chart represents only the usage in April 2017. Based on discussions with campus network and security personnel the load distribution (percentage-wise) to the main DNS servers may vary decidedly over the year.

January 1, 2016 - March 31, 2017		
	Average <stdev>	
	DNS Queries	DNS Responses
Business hours	**131,159** <46,912>	**140,994** <46,957>
Non-business hours	**32,748** <16,721>	**36,215** <17,028>
Overall	**59,256** <51,992>	**64,439** <54,465>
Weekdays	**1,714,726** <555,143>	**1,857,962** <549,041>
Weekends	**677,358** <211,875>	**753,533** <202,939>
Overall	**1,419,652** <672,714>	**1,543,813** <689,911>
Spring	**42,383,404** <11,047,856>	**46,735,649** <9,451,150>
Summer	**35,046,415** <5,680,691>	**38,198,061** <5,522,259>
Fall	**50,493,527** <4,818,243>	**53,693,549** <4,916,530>

TABLE I: DNS Traffic Statistics - Depicted in this table are averages and standard deviations of DNS Queries and Responses for various time periods that provide insight into the usage of our network.

The significance of the monitored DNS data does not necessarily lie in the overall quantity of observed DNS packets, but rather that the traffic originates from many diverse sources. We observed traffic from 55 out of the 103 (53%) total buildings on campus, including student housing. We also recorded 54.28% of the VPN IP address block (from outside the university network). Finally, we saw 48% of the three */16*s and one */21* IP address blocks assigned to the wireless range. By analyzing the IP address assignment pattern for the wireless network, we conclude that they were assigned uniformly at random, and hence are not biased.

All in all, we recorded (and observed in near real-time) DNS traffic from more than 50% of the campus buildings (university halls and housing) as well as covered around 50% of the IP blocks assigned to the virtual infrastructure (VPN and wireless network segments). A uniform random distribution of IP address assignment further suggests that the data are representative of the entire university campus.

Overall usage. Over 15 months we recorded an overall average of 59,256 DNS query packets and 64,439 responses per hour with high standard deviation values (see Table I). We have also observed significant differences between business hours versus non-business hours loads (Table I) or weekdays versus weekends (Table I).

Academic year. Due to our monitoring environment, we investigated DNS traffic loads during different times of the academic year. Table I shows monthly averages during the Spring, Summer, and Fall semesters. As expected, we noticed higher averages during the Fall and Spring semesters when more undergraduate students are present on-campus. More-

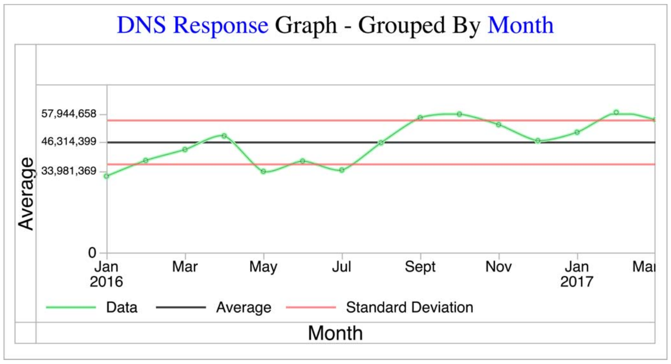

Fig. 5: Overall monthly distribution of all the DNS responses from January 1, 2016 to March 31, 2017

over, Figure 5 shows the monthly average distribution of DNS responses over the analyzed timeframe which roughly covers two Spring semesters and one Summer and Fall semester.

IV. Analysis and Findings

Using eyeDNS we were able to gain a unique perspective into what is happening on a campus network of a US public research institution. This perspective provides additional visibility for the campus IT security and networking personnel.

A. Algorithmically Generated Domains

Malware, especially botnets [27], remain in contact with their C&C servers to receive updates and instructions. Bots use DGAs to generate a large number of domain names and query the domain names to obtain their C&C servers' IP addresses. The generated domains are time dependent, i.e., a unique batch of domains are generated in each time period. The attacker, usually, registers one of the domains in that time period and uses the domain as the rendezvous point. If the domain is discovered by defenders and is blacklisted, the attacker registers and uses a different domain, thereby rendering domain blacklisting ineffective [7].

We analyzed our dataset in an attempt to uncover such algorithmically generated domains. In this effort we leveraged DGArchive, presented in [6]. DGArchive allowed us to determine whether a domain name was dynamically created by malware using DGAs. Moreover, DGArchive is able to compute domains that will be queried by malware in the future. From eyeDNS, we obtained the list of unique domains which are not whitelisted, a total of over 1.4M records, and recorded the result from DGArchive for those domains.

DGArchive Database Results. DGArchive maintains a database of archived domains generated by reverse engineering malware-DGAs and enumerating all possible domains that a given DGA could create. This allowed us to query their database for domains appearing in our dataset to see if these domains are associated with a DGA family and if so, when the algorithmically generated domain was, is, or will be active.

Table II captures the results; we recorded a total of 5,588 hits. A hit is a domain associated with a DGA family used in malware campaigns. We can observe that the most popular DGA family in 2016 was "necurs_dga" while 2017 (January to March 2017) was dominated by "virut_dga_34f6b17c". It is worth mentioning that Necurs is a well-known botnet with varying activity periods [29].

DNS queries for the algorithmically generated domains were issued after the domains were active (e.g., Families active 2008 - 2015), while the domains were active (Active "Hits"), or before they will be active, "Future Hits". For instance, in 2016, all domains (qnames) associated with the "necurs_dga" DGA family were active when campus clients issued 16,585 DNS queries. In 2017 (January to March), out of 53,341 queries to domains associated with "virut_dga_34f6b17c", 34,098 of the queries were targeting future "virut_dga_34f6b17c" domains that may be active after the end date of our current dataset (March 31, 2017).

The active timeframe for a domain used by a DGA family was usually between 1 and 10 days. However, we encountered a few families where the domains are active 1970 - 2106 ("forever"). All queries issued to these domains are active hits.

By leveraging the data from the next 6 months (April 1, 2017 – October 12, 2017), we discovered 2308 active hits for "necurs_dga" (making it the most consistent malware family for 2017 as well), two active hits for "wd_dga_efe2056d" [30] and two active hits for "ccleaner_dga_264bc2fd" [31]. However, we have not yet encountered any of the expected "Future Hits" from the initial timeframe.

By combing the eyeDNS data with DGArchive, we were able to provide the security team a different perspective into the activities on the campus network by identifying hosts which participated in malware campaigns and by keeping an eye on hosts that could potentially participate in the future.

B. Suspicious and Anomalous Findings

Over the course of 15 months we noticed several interesting events from usage spikes to risky or suspicious behaviors. Here, we focus on a few examples of these interesting events.

WPAD. We were surprised by around 10M DNS queries for domain names that start with "wpad". This covers approx. 1.5% of the total queries (e.g., wpad.users.campus was used in 4,832,029 DNS queries). WPAD (Web Proxy Auto Discovery) [32] is a protocol designed for web based clients to automatically obtain and configure proxy settings.

We tested different operating systems and discovered that by default Windows hosts were sending DNS queries for domains beginning with WPAD. From the queries we were able to identify, without performing a network scan, on-campus departments with Windows machines. Usually, queries for "wpad" domain names were generated by multiple hosts located behind a departmental DNS server. Unfortunately, all these system may have been exposed to a Man-In-The-Middle (MITM) attack by a host serving malicious Proxy Auto-Configuration (PAC) JavaScript files causing all web traffic to be routed through the malicious host [33]. Using eyeDNS we were able to identify this wide-spread risky setup within our network. Remediation actions are under way.

Signs of Scams. University campuses are often targeted by scamming schemes, from phishing emails requesting users

DNS data collected between January 1, **2016** - March 31, **2017**

Year	**DGA Family**		**Number of Domains (Hits)**	**DNS Queries**			**Example Domain Hits**
	Name	**Validity (per domain)**		**Total**	**Active Hits**	**Future Hits**	
2017	virut_dga_34f6b17c	1 day	427	53,341	5	34,098	`ahorre.com`
	conficker_dga_[id$_1$]	1 day	9	445	0	0	`byypq.com`
	nymaim_dga_[id$_2$]	1 day	4	92	0	9	`atjct.com`
	gozi_dga_9ff51775	5 days	1	6	0	6	`proposalspace.com`
	matsnu_dga_0990c739	3 days	1	46	0	2	`communication.com`
	modpack_dga_1e63a49d	7 days	1	0	0	0	`2s5m19yk.ru`
			443	*53,930*	*5*	*34,115*	
2016	necurs_dga_[id$_3$]	3 to 4 days	2,905	16,585	16,585	0	`npkxghmoru.biz`
	virut_dga_34f6b17c	1 day	409	770,601	38	526,758	`meiuio.com`
	bedep_dga_[id$_4$]	7 days	104	2,021	1,770	213	
	conficker_dga_[id$_1$]	1 day	10	4,132	0	1,396	`ccnks.org`
	modpack_dga_1e63a49d	7 days	10	584	521	48	`gvaq70s7he.ru`
	nymaim_dga_[id$_5$]	1 day	4	743	0	68	`ignum.com`
	qadars_dga_d4cc4071	7 days	2	1,125	0	0	`j8le7s5q745e.org`
	gozi_dga_[id$_6$]	3 days	1	589	0	86	`administrator.com`
	matsnu_dga_fad3b90f	4 days	1	196	0	13	`companyliving.com`
			3,446	*796,549*	*18,914*	*528,582*	
“forever”	vawtrak_dga_f26b9663	1970 - 2106 (49,711 days)	150	13,600	13,600	0	`tsfyvqh.com`
	banjori_dga_3c0dec8c		1	12	12	0	`kensmen.com`
	necurs_dga_aa359c41		1	66	66	0	`npkxghmoru.biz`
	simda_dga_b4e869a6		1	4	4	0	`nomadat.net`
			153	*13,682*	*13,682*	0	
“old”	Families active 2008 - 2015	1 to 10 days	*1,546*	*1,614*	0	0	`syguke.com`
	Totals:		*5,588*	*865,775*	*32,601*	*562,697*	

id$_1$ ∈ {3560e585, 69998fdd}, id$_2$ ∈ {5fff7b0c, e28d02bb}, id$_3$ ∈ {a471c084, aa359c41}, id$_4$ ∈ {1d8097c1, 21c17dc9}
id$_5$ ∈ {5fff7b0c, ae243a07, b80775ed, e28d02bb, f6abb372, fede145a}, id$_6$ ∈ {0995326d, 3840f96e, 90a631c6, 9ff51775}

TABLE II: DGArchive “Hits” – Summarized results after checking the unique domain names in our dataset (excluded whitelisted domains) against DGArchive [6, 28]. We recorded a total of 5,588 hits, domain names used by malware-DGA families. Some of these domains were active (valid) when queried (“Active Hits”), others may be used in a future malware campaign (“Future Hits”) – After reverse engineering a DGA family, DGArchive computed based on the identified seed future domain names that may be used by that family.

to “validate” their accounts [34] to scam phone calls (e.g., technical support [18], IRS scam [35], international students called by fake USCIS [36]). The flexibility of eyeDNS allowed us to adopt many of the strategies presented in previous work [18, 37, 38] to find signs of malicious scam activity. Applying these strategies to our data, we found the presence of scam websites, “cheap domains”, and typo-squatted domains as studied in [18], as well as ad-based URL shortening services presented by Nikiforakis et al. [38]. While utilizing the list of typo-squatted domains provided by [18], we noticed that many of the domains present in the list were, in fact, not typo-domains (e.g., `google.com`, `buzzfeed.com`). The implementation of the typo-domain algorithm [37] did not account for duplicate letter switching that resulted in the original domain. Thus we removed the original domains from the list of typo-squatted domains and then identified typo-squatted domains in our data set as well. For example, `buzzfed.com` with 23,049 queries and `umblr.com` with 14,151 queries, and a total query count of 880,497 queries on 26,670 domain names by following procedures used in [18]. However, we noticed `yp.com` and `ft.com` in the top ten of our results, which are legitimate domains but which also have the appearance of a typo-squatted domain.

Network Rerouting. On June 22, 2016 between 12AM and 1AM, we saw an increase in all DNS packet counts from the usual average of 31,400 during that hour to a peak value of 284,595. All counts (whitelist, graylist, and NXDomain) except the blacklist count, went up. Surprisingly, after that hour, more than 50% of the top hosts (by query count) during that hour were not seen again in our logs. The cause of this increase was DNS traffic that is normally served by other DNS servers on campus being routed through our monitored server (due to maintenance on other DNS servers).

V. Limitations and Discussion

Deploying and maintaining a DNS traffic analysis pipeline on an operational network is challenging. Our main limitations relate to coverage and the blacklisting mechanism.

As noted, due to highly distributed DNS infrastructure we were able to monitor only one out of three main DNS servers and no secondary servers. Moreover, clients may use external DNS service and would not appear in our data (see Figure 4). Although we are analyzing a fraction of whole campus DNS traffic, this traffic originates from most of the campus buildings and network segments. Based on our observations, the data is representative of the entire campus (Section III-E: Coverage).

Throughout our work, we found blacklisting mechanism to be inconsistent and we encountered challenges from not labeling domains properly to marking benign domains as malicious. Moreover, some domains on our blacklist were "cleaned" and re-purposed. Thus we are currently using the blacklist as a guideline and evaluate the possibility of using a more developed approach.

Additionally, the capture box analyzes domain names only up to the second level domain. This may be an issue for certain analyses i.e., tracking users that clicked on links included in phishing emails. We are currently exploring different ways to capture the whole domain name.

Regardless of above above-mentioned limitations, eyeDNS is able to provide additional visibility into different segments of the network. This led to identifying domain names used in malware campaigns, uncovering signs of scams, and near real-time tracking of anomalous behaviors on the network. Future work includes enhancing our framework's real-time capabilities and evaluating in more details similar frameworks.

VI. Conclusions

In this paper, we introduce a framework, called eyeDNS, to collect, store, analyze, and visualize high-volume DNS traffic. We deployed eyeDNS in a large US public university campus network, collected and analyzed DNS data over a 15 month period, and showed that near real time DNS traffic analysis is feasible in a dynamic and open environment. Our system can provide additional visibility into a highly diverse environment and could identify anomalous behaviors such as DGAs. These insights have helped the campus networking and security personnel to better assess the health of the network, as well as to better understand different types of activities that take place within the campus IP range.

Acknowledgments. This work was supported by the Air Force Office of Scientific Research (FA9550-12-1-0106). Opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

References

[1] ISC RFC 1034, "Domain names – concepts and facilities."

[2] Internet live stats, "Total number of websites," http://www.internetlivestats.com/total-number-of-websites/, accessed 4/2017.

[3] M. Korczyński, M. Król, and M. van Eeten, "Zone poisoning: The how and where of non-secure DNS dynamic updates," in *ACM IMC*, 2016.

[4] A. McLean, G. Gates, and A. Tse, "How the cyberattack on Spamhaus unfolded," The New York Times, accessed 4/2017.

[5] Oracle Dyn, "Dyn analysis summary of Friday October 21 attack," https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/, accessed 4/2017.

[6] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *USENIX Security*, 2016.

[7] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for DNS," in *USENIX Security*, 2010.

[8] P. Calyam, D. Krymskiy, M. Sridharan, and P. Schopis, "Active and passive measurements on campus, regional and national network backbone paths," in *ICCCN*, 2005.

[9] E. Stalmans, "A framework for DNS based detection and mitigation of malware infections on a network," in *ISSA*, 2011.

[10] A. Greenberg, "How hackers hijacked a bank's entire online operation," Wired, April 2017, accessed 10/2017.

[11] T. Lauinger, K. Onarlioglu, A. Chaabane, W. Robertson, and E. Kirda, "Whois lost in translation: (mis)understanding domain name expiration and re-registration," in *ACM IMC*, 2016.

[12] H. Zhang, M. Gharaibeh, S. Thanasoulas, and C. Papadopoulos, "Botdigger: Detecting DGA bots in a single network," in *IEEE TMA*, 2016.

[13] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, II, and D. Dagon, "Detecting malware domains at the upper DNS hierarchy," in *USENIX Security*, 2011.

[14] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A passive DNS analysis service to detect and report malicious domains," in *ACM TISSEC*, 2014.

[15] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in *USENIX Security*, 2012.

[16] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-based botnet tracking and intelligence," in *DIMVA*, 2014.

[17] S. Yadav and A. L. N. Reddy, "Winning with DNS failures: Strategies for faster botnet detection," in *SecureComm*, 2011.

[18] N. Miramirkhani, O. Starov, and N. Nikiforakis, "Dial one for scam: A large-scale analysis of technical support scams," in *NDSS*, 2017.

[19] DNS-OARC, "DNS-OARC – DNS statistics collector," https://www.dns-oarc.net/tools/dsc, accessed 01/2018.

[20] ISC RFC 5424, "The syslog protocol."

[21] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling, "Measuring and detecting fast-flux service networks," in *NDSS*, 2008.

[22] M. Jonker, A. Sperotto, R. van Rijswijk-Deij, R. Sadre, and A. Pras, "Measuring the adoption of DDoS protection services," in *ACM IMC*, 2016.

[23] R. Perdisci, I. Corona, D. Dagon, and W. Lee, "Detecting malicious flux service networks through passive analysis of recursive DNS traces," in *ACSAC*, 2009.

[24] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Efficient and accurate behavior-based tracking of malware-control domains in large ISP networks," in *ACM TOPS*, 2016.

[25] W. Scott, T. Anderson, T. Kohno, and A. Krishnamurthy, "Satellite: Joint analysis of CDNs and network-level interference," in *USENIX ATC*, 2016.

[26] D. Springall, Z. Durumeric, and J. A. Halderman, "Measuring the security harm of TLS crypto shortcuts," in *ACM IMC*, 2016.

[27] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet detection by monitoring group activities in DNS traffic," in *IEEE CIT*, 2007.

[28] Fraunhofer FKIE (administrated by D. Plohmann), "DGArchive," https://dgarchive.caad.fkie.fraunhofer.de/, accessed 5/2017.

[29] M. Kumar, "Botnet sending 5 million emails per hour to spread Jaff ransomware," The Hacker News, May 2017, accessed 5/2017.

[30] Kafeine, "Threat actor goes on a Chrome extension hijacking spree," https://www.proofpoint.com/us/threat-insight/post/threat-actor-goes-chrome-extension-hijacking-spree, accessed 10/2017.

[31] C. Talos, "CCleanup: A vast number of machines at risk," http://blog.talosintelligence.com/2017/09/avast-distributes-malware.html, accessed 10/2017.

[32] ISC RFC 3040, "Internet web replication and caching taxonomy."

[33] US-CERT, "Alert (TA16-144A) WPAD Name Collision Vulnerability," https://www.us-cert.gov/ncas/alerts/TA16-144A, accessed 4/2017.

[34] C. Hanson, "Be alert to email phishing scams!" https://news.wsu.edu/2015/01/05/be-alert-to-email-phishing-scams, accessed 5/2017.

[35] IRS, "IRS warns of latest scam variation involving bogus federal student taxes," https://www.irs.gov/newsroom/irs-warns-of-latest-scam-variation-involving-bogus-federal-student-tax, accessed 5/2017.

[36] M. Merchant, "Stay informed: Scams affecting international students and scholars," https://sites.utexas.edu/isss/2013/04/stay-informed-scams-affecting-international-students-scholars/, accessed 5/2017.

[37] Y.-M. Wang, D. Beck, J. Wang, C. Verbowski, and B. Daniels, "Strider typo-patrol: Discovery and analysis of systematic typo-squatting," in *USENIX SRUTI*, 2006.

[38] N. Nikiforakis, F. Maggi, G. Stringhini, M. Z. Rafique, W. Joosen, C. Kruegel, F. Piessens, G. Vigna, and S. Zanero, "Stranger danger: Exploring the ecosystem of ad-based URL shortening services," in *WWW*, 2014.