Anomaly Detection by passive DNS analysis of Alexa Top Domains

Ayush Sharma, Harshitha Ramamurthy, Hirva Shah, Pragna Hasthanthar

***Abstract*—The Domain Name System is a fundamental component of Internet since it maps easy-to-remember domain names to IP addresses. Therefore, it is usually the primary target for most of the malicious attacks such as DNS Poisoning and Rogue DNS servers. With the help of 0x20 bit encoding [10], the problem of DNS Poisoning is mitigated to quite a large extent although it has a minor requirement that the authoritative nameserver should be able to preserve the case of the DNS query. In addition, it is usually difficult to detect the rogue DNS server above the stub resolver [11]. We propose an anomaly detection system which would be able to raise a red flag in case of DNS Poisoning and malicious DNS authority by passive DNS analysis of domain names and then comparing them with the 0^{th} day cluster of the database. We perform the passive DNS analysis for 30 days by querying the WHOIS server of CYMRU, compare the network profiles of the domain names crawled with the 0^{th} day cluster and categorize the domain names as anomalous and non-anomalous depending upon the changes in the cluster of a domain name. In the process, we also create a WHOIS repository for Alexa domain names which is faster to query than the WHOIS server.**

I. Introduction

The Domain Name System is fundamental to internet traffic today. The concept of DNS took birth around three decades ago and its importance has not diminished [13]. The internet depends on DNS, if DNS ceases to function, so does the internet. Thus, DNS is an attractive protocol for attackers to exploit the vulnerabilities in a network. In addition to DNS being extensively used, its growth continues to be fast. It has been reported that the volume of DNS queries has climbed over 200 percent in the last few years.This is because of the growth in entertainment sites, social media, search engines and e-commerce websites. The burden on DNS increases as sites become more sophisticated. This is because, these days, a single web-page will have images, links to other web-pages, icons etc. For example, a popular page like cnn.com requires over one hundred lookups. As the web grows more sophisticated, it means that DNS requests also increase [14].

Our project involves creating a passive DNS database of the domains on the Alexa top 1 million sites. This DNS database is fed to our clustering model which clusters the domain names based on the network features. The network features that we have considered for this project are BGP features, AS features, duration of IP allocation and TTL features. We considered network features since [1] mentions that most legitimate and professionally run internet services have a stable network profile. On the other hand, less benign domain names change their profiles frequently, like fast-flux networks. One example where the network features will enable to identify anomalous behavior is the case where a domain name d resides in a larger number of different networks as compared to benign domain names. We extract 12 features in total namely number of distinct IP addresses, BGP prefixes, AS numbers, countries, duration of IP allocated, TTL, unique IP addresses, BGP prefixes, AS numbers, countries and extent of uniqueness of RHIPs (IPs, BGP, AS, CC). With the help of all these features, we attempt to detect anomalies by observing if the domain names we crawl in Alexa change clusters as per our clustering module. We have also fed a known set of domain names belonging to the Zeus botnet domains and malicious domains from $malwaredomains.com$ to our clustering module. We recommend further inspection of those domain names if they move into the cluster which contains the Zeus domain names. [17] stated that recent botnets like Conficker, Kraken, Torping etc have used DNS based domain fluxing. The features that we have chosen will detect domain fluxing and thus cluster those domains involved in domain fluxing with the Zeus domains. Thus our project makes these primary contributions-

1) We have built a passive DNS repository for the domain names present in the Alexa top 1 million list of websites by querying the WHOIS server.
2) We have performed semi-supervised X-means clustering based on the network profiles of the domains
3) We categorize domains into two categories, anomalous and non-anomalous on n^{th} day relative to the 0^{th} day.

II. Background and Related Work

The Domain Name System (DNS) is a globally distributed database that maps names to network locations, thus providing information critical to the operation of most Internet application and services. DNS was designed to allow the database to be maintained in a distributed manner [5]. It was also designed to not have obvious size limits for names, name components and the data associated with a name [4]. DNS employs two main concepts to work- zones and caching. Zones are portions which are controlled by an organization. This organization assumes the responsibility of making its clients visible to the other zones. Caching is a mechanism by which a request made by a client can be stored for a specific period of time in case the same client or other clients make the same request again [6]. These two factors contribute to the scalability of DNS. They also seek to reduce the load on the root servers at the top of the name space hierarchy because caching reduces delays in responses to the clients. They also reduce wide-area network bandwidth usage. The hierarchical structure of DNS is such that a domain name is made up of labels. The DNS namespace

is usually visualized as a tree where each node in the tree corresponds to a label. The domain name of a node is the concatenation of all the labels on the path from the node to the root of the tree. The main components that make up the architecture of DNS are the name-servers and resolvers. A name server hosts information which can tell a client where a particular domain name is present. A resolver is a client-facing component which finds the name server that is able to answer a query generated by a client.

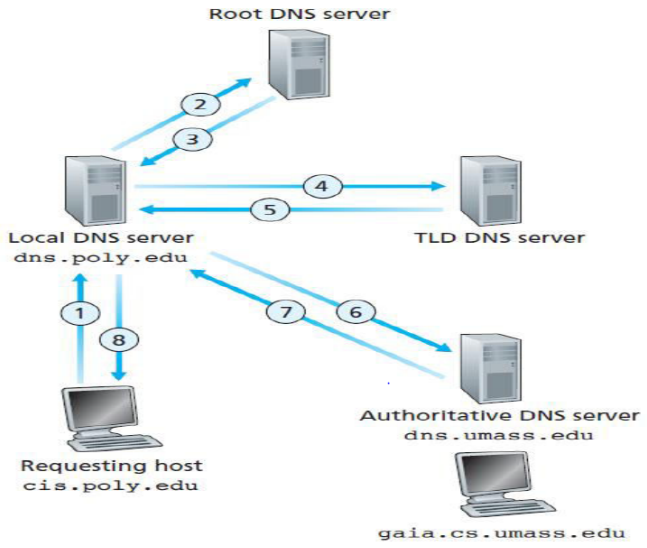

Fig. 1: DNS Resolution Process

We derive our features from two major passive DNS monitoring systems: **Notos** and **Exposure**. Notos is a dynamic and comprehensive reputation system for DNS which outputs the reputation scores for domains. Notos uses network and zone based features to dynamically assign reputation scores to new and unknown domains with high scores being assigned to legitimate domains and low scores to malicious domains. Notos uses historical information about legitimate and malicious domains to train its classifier and establish the ground truth and thus is able to build a model that assigns reputation scores to new domain names. We use network features based on RHIP such as total number of IPs associated with a domain name, the diversity of their geographical locations, the BGP prefixes and also the number of distinct autonomous systems to implement our clustering and observe the agility with which the domain names in the Alexa top 1 million change clusters [4]. Exposure is another system that extracts features from DNS traffic and employs large-scale, passive DNS analysis techniques to identify malicious domains. It used 15 behavioral features to carry-out its analysis. Four of its features are based on DNS Answer-Based Features such as the number of distinct IP addresses and the number of distinct countries. We consider these features in our system. Exposure also uses TTL Value-Based features, similarly we consider the TTL values that WHOIS returns as one of our features.

III. System Overview

Our system is made up of a data collection module, feature extraction module and a clustering module as shown in Figure 2. Our data collection module builds a passive DNS database by resolving the top 100 thousand and bottom 100 thousand from the Alexa top 1 million list. In addition to this, we also resolve the top 500 sites in the 14 different categories present on the Alexa website. We use dnspython to resolve the domain names and the resulting IP addresses are fed to the WHOIS server which returns more information pertaining to the domain name. The data we log for a domain name under investigation are- all the IP addresses associated with it, BGP prefixes, the different countries the IP addresses belong to, the different ASes the IP addresses belong to, TTL value and the day that IP address was allocated to the domain. We perform post-processing on the raw data we have logged to transform it into a format readable by the WEKA clustering module. We have configured the WEKA clustering module to perform X-Means clustering on the files we input. The output of the clustering module are domain names sorted into clusters based on the features we have considered. We make a comparison of the clusters from the 0th day till the Nth day and reason out why certain domains change clusters if they do so. The observation will let us identify if there is any anomaly in the domain name resolution.

IV. Creating passive DNS Repository

A. PycURL and Scapy

Anomaly detection systems flag observed activities that deviate significantly from the established patterns as anomalies. To build such a system, we first need to understand and establish the acceptable behavior or the expected normal characteristics. For building an anomaly detection system based on DNS information, the first step is to create a local database on which more elaborate queries are possible. To create such a passive DNS database, one of the methods that has been used is to obtain domain name system data from production networks as described in Weimers [17] method. Capturing traffic from a production network requires access to the recursive resolvers to log the queries, and also requires a lot of data reduction as there will traffic from tens of thousands of hosts. But since we have access to the most popular 1 million sites listed provided by Alexa, we know what are the exact queries that need to be sent and therefore we have resorted to periodically polling for DNS records and gathering them.

Gathering information about the infrastructure for the most popular domains has been carried out by generating DNS queries for the domains by means of actual HTTP GET requests. The HTTP GET requests were generated using the cURL library which is a free and easy-to-use client-side URL transfer library. Although it is more popular as a command line tool for getting or sending files using URL syntax, the cURL library also uses the DNS system by default to resolve the domains or host name. This property has been used to generate requests for the set of domains which have been considered

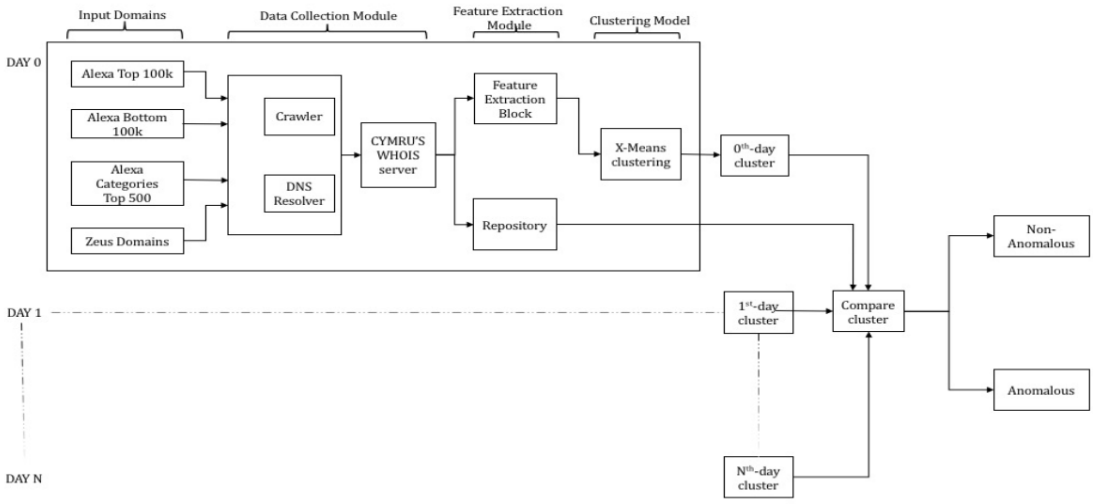

Fig. 2: System Overview

for carrying out the DNS analysis. This method is simpler and faster than crafting a DNS packet to query the DNS servers for gathering information. The cURL library is thread-safe and has support for threaded resolver which is very convenient for having a scalable polling mechanism which has been adopted. The DNS queries were generated by using Pycurl, the python interface to libcurl. The network requests for DNS resolutions via HTTP GET were driven asynchronously with the multicurl interface in Pycurl as we were able to establish at least ten concurrent, fast and reliable connections. Based on the computing resources available on the node which was used to establish these network connections, we decided to have 6 concurrent connections. The next step after sending out DNS queries is to be have a mechanism to capture the DNS replies that were provided in response to the queries. We have used PCAP, a packet capture library which provides a high level interface to packet capture systems. By using the appropriate filters for DNS ports, we have been able to capture the raw traffic into pcap files which have been then fed to the packet analyzer in our system for analysis. Scapy, a python based powerful packet analysis tool has been used to implement the packet analyzer for extracting the details contained in a DNS packet. To extract the various details that are required to be stored in the DNS database, it is essential to understand the makeup of a DNS packet. The DNS packets have a structure that is shown in the figure 3. The header describes the type of the packet and the fields contained in the packet. The other main fields correspond to the questions directed to the name server, the Answers to these questions, the Authority and Additional sections. The DNS servers put the Resource Records (RR) meant for the client in the Answer, Authority and Additional sections. The fields of the DNS packets which we have checked to gather our information are the QR field(one bit) which is set to 1 to indicate a response or 0 for query, the AA field (one bit) which when set to 1 indicates that responding name server is an authority server, the ANCOUNT (16 bits) field which specifies the number of resource records in the answer section. We have extracted all the resource records in a DNS response packet and stored the domain name and TTL. Since we focused on storing all the IP addresses that mapped to a given domain, the RDATA field has been stored only when it is an IP address which is the case for a TYPE A resource records.

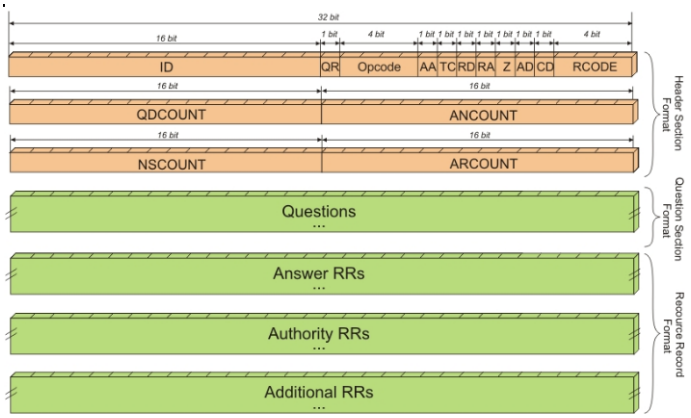

Fig. 3: Structure of a DNS packet

B. DNS Resolver and WHOIS

The process of retrieving web pages using PyCurl was generating HTTP traffic. The web pages were downloaded onto the VM and we employed a daemon to carry out periodic cleanup to delete these files so as to not run out of space. Since all the information required for the analysis was present in the DNS response we decided to use a DNS resolver to reduce the load. Thus, we decided to use dnspython to build a DNS Resolver. *dnspython* is a DNS toolkit for Python which

supports almost all record types and also supports EDNS0. The high level classes perform queries for data of a given name, type, and class, and return an answer set. The low level classes allow direct manipulation of DNS zones, messages, names, and records. We have used the high level classes to perform queries for the domain names taken from Alexas top sites list to capture the information contained in the A- records of a DNS response. We have also considered the case where the absence of a DNS resolution results in NXDOMAIN response.

WHOIS is a query/response protocol that is widely used to query databases that contain information about the Internet resources like domain names, IP addresses associated with the domain names, autonomous system etc. WHOIS is a TCP based protocol which delivers its content in a human-readable format. The WHOIS server listens on TCP port 43 for requests from WHOIS clients. The WHOIS server closes its connection as soon as the output is finished. The closed TCP connection is the indication to the client that the response has been received.We have used the IP to ASN mapping service provided by Team CYMRU since this is one of the the most stable free service available. Team CYMRU provides a number of query interfaces that allow for the mapping of IP addresses to BGP prefixes and Autonomous System Numbers (ASNs), based on BGP feeds from their 50+ BGP peers. This information is updated every 4 hours. We have used the query interfaces provided for the traditional WHOIS protocol to query the WHOIS server- whois.cymru.com. The country code, registry and allocation date provided by this service are all based on data obtained directly from the Regional Internet Registries (RIR) including: ARIN, RIPE, AFRNIC, APNIC, LACNIC. Therefore, this data is as accurate as the data present in the RIR databases.

All the above data has been obtained from the DNS resolver and WHOIS for the list of domains by spawning sub-processes in python. We have mapped equal subsets of the input domain list to each sub-process. We observed better results with multiple process than with multiple thread based queries in python as the Global Interpreter Lock(GIL) can be effectively side-stepped using multi-processing. Also, to prevent any waiting and blocking after each sub process has completed its task, the individual subprocesses are designed generate separate output files with the collected DNS data which is then concatenated in the feature extraction stage.

C. Scrapy

Alexa releases a csv file of the top 1 million sites everyday, but it does not do the same for the different categories it has listed on the website. Hence, we used Scrapy to get the list of top 500 domains in different categories like Arts, Business, Games, Health etc. Scrapy is a free and open source web crawling framework which has been written in Python. Scrapy is written in Twisted, a popular event-driven networking framework for Python [6] The architecture of Scrapy has been shown in Figure 4.

The Scrapy Engine is responsible for controlling the data flow between all components of the system, and the triggering

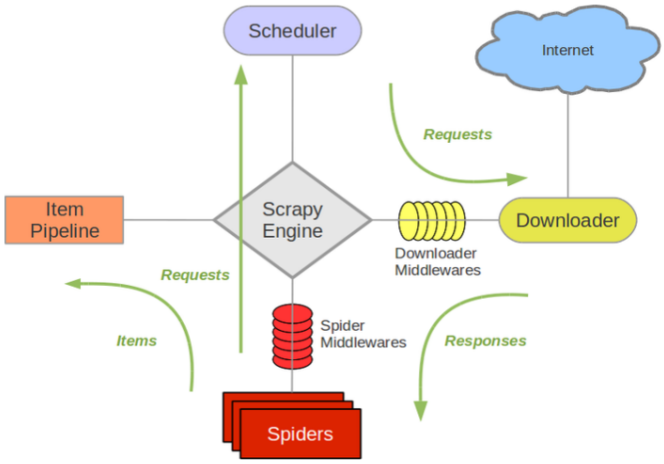

Fig. 4: Architecture of Scrapy

events when certain actions occur. The Scheduler is like any other scheduler, it enqueues the requests from the engine. The Downloader will fetch the web pages and make it available to the engine which in turn feeds it to the spiders. The Spiders are the classes which do the crawling. They are written by Scrapy users who can custom code it according to their requirements. The spiders can parse the response, extract items or recursively crawl.

To our end, we used a spider to crawl the Alexa website for the list of the top 500 domains in the 14 different categories on the Alexa website. The different categories are News, Regional, Arts, Business, Adult, Health, Home, Kids and Teens, Recreation, Reference, Science, Shopping, Society and Sports. We use the Rules object to specify the behavior in which we want to crawl the site. And then using the Link Extractor object we specify how the links will be extracted from the crawled page. These urls are parsed and stored as an item which we then output to a csv file.

V. Feature Extraction

To make a conclusive observation of the DNS infrastructure of the websites featured on the Alexa list, we extract network based statistical features. In this section we discuss about the features extracted and the reason behind their selection.

For a given domain name, we extract a number of statistical features from its RHIP (Related Historic IPs) which exhibit how the network operators who manage the domain under investigation allocate resources. A general behavior associated with malicious domains is the agility. The reason for exhibiting DNS agility is to evade the public blacklists and takedowns. Following are the major groups under which our features fall into:

1) **BGP Features**: We monitor all the distinct IP addresses contained in the response packet of a domain and further use it to generate the different BGP prefixes it resolves to, by querying for an individual IP address

from CYMRU's WHOIS server. Most of the content delivery networks(CDNs) resolve into same BGP Prefix in a single query and maintain those prefixes for subsequent queries. These network profiles will differ from a malicious domain as it would change the BGP Prefix upon subsequent queries.

2) **AS Features and IP Allocation period**: We also extract the geographical information from the RHIPs obtained from WHOIS, calculate the number of unique AS numbers a particular domain resolve to and the date at which a particular IP was allocated to a domain. The network profiles created for BGP features follow a similar trend for AS features. IP allocation period reflects the stability of a domain name with respect to an IP address.
3) **TTL features**: In addition to using the above mentioned features from Notos, we use TTL associated with a domain to gain further resolution into its network behavior. A very low value of TTL will be associated with a malicious domain name. Same would be true for a content delivery network as well. But CDNs usually return a lot of IP addresses in response to a query as compared to a malicious domain name

We extract 12 features in total namely number of distinct IP addresses, BGP prefixes, AS numbers, countries, duration of IP allocated, TTL, unique IP addresses, BGP prefixes, AS numbers, countries and extent of uniqueness of RHIPs (IPs, BGP). Most of the domain names present in top Alexa list will have a very stable network profile wherein they will remain in their respective clusters. In an event where one of the domains change their cluster, we record it as anomalous and non-anomalous domains.

On the 16^{th} day, while manually inspecting the database we came across 160 domains that resolved to private IP addresses as shown in Figure 5. From the point of view of DNS resolution system, it can be accounted as an anomaly if not malicious as private IP addresses are not supposed to be advertised. We just include a simultaneous check for detecting this specific type of anomaly as a feature.

Rank	Domain	RHIP	TTL	AS	BGP Prefix
8956	phncdn.com	127.0.0.1	86400	NA	NA
9618	adsupplyads.com	127.0.0.1	900	NA	NA
10190	pnu.ac.ir	10.8.212.1	1200	NA	NA
993890	bauermedia.group	127.0.53.5	3600	NA	NA
996794	caradisiac.dev	127.0.53.5	3600	NA	NA
999641	groundwork-core.dev	127.0.53.5	3600	NA	NA
Zeus	childrentrainingcentre.com	127.0.0.1	1800	NA	NA
Zeus	cynthialemos1225.ddns.net	0.0.0.0	360	NA	NA
Zeus	nancycemt1225.ddns.net	0.0.0.0	360	NA	NA
Zeus	sandokan66.no-ip.info	0.0.0.0	360	NA	NA
Zeus	z0bu.dynu.com	127.0.0.1	60	NA	NA

Fig. 5: Domains resolving to private IP addresses

VI. Clustering Module

After extracting features, we use Weka to implement X-Means clustering on the network features extracted out of pDNS repository. X-means basically covers the limitations posed by Simple K-Means clustering algorithm. K-Means is computationally intensive, requires the number of clusters information apriori and usually settles for some local minima. X-Means mitigates all of the above mentioned limitations (first and second, three to an extent) and it reveals the true number of classes in an underlying distribution and it is much faster than using accelerated K-Means for different values of K (clusters). We input the features extracted from the domains and get following clusters:

1) Popular Domains: This class consists of a large set of domain names under the following DNS zones: google.com, yahoo.com, amazon.com, ebay.com, msn.com, live.com, myspace.com, and facebook.com. Their network behavior is distinct from the other profiles as they use a large set of IP addresses but the BGP prefix and AS usually remain same.
2) Common Domains: Common domains cover most of the domains found in Alexa list. They exhibit a fairly stable network profile as they don't advertise a lot of IP addresses and have on average, a high TTL value associated with the IPs.
3) CDN Domains: This class consists of domains like panthercdn.com, llnwd.net, cloudfront.net, nyud.net, nyucd.net and redcondor.net. Their network profile is different from other categories in that they advertise several IP addresses but their geographical locations and BGP prefixes are more diverse than popular domains.
4) Dynamic Domains: This class includes a large set of domain names registered under two of the largest dynamic DNS providers, namely No-IP (no-ip.com) and DynDNS (dyndns.com).
5) Malicious domains: This cluster comprises of blacklisted zeus domain names. Due to their DNS agility, their network behavior can be distinguished from the other clusters. They also have very small TTL value. By the virtue of their profile, some of the akamai domains will also fall into this cluster. They can be separated into a different cluster by zone clustering.

VII. Data collection and Analysis

This section summarizes the data collection stage and challenges faced. It also covers analysis of the clusters and the results obtained from comparing clusters from different days.

A. Data collection

We use two kinds of mechanism for data collection. Initially we use a crawler designed from pycurl to visit the top Alexa sites and capture packets in a pcap file. We then use the pcap file to extract features. We use this crawler to construct pDNS database from 19th October 2015 to 4th November 2015. We switch to using DNS resolver for populating our database from 5th November 2015 to 17th November 2015 due to following reasons:

1) We ended up generating a lot of HTTP traffic on the VM for the days we were performing crawling using all the cores of the VM.

2) We only needed DNS Resource records for the purpose of updating our database therefore, all the other data is unnecessary
3) Implementing a DNS resolver instead of a crawler increases the rate at which we update our database.

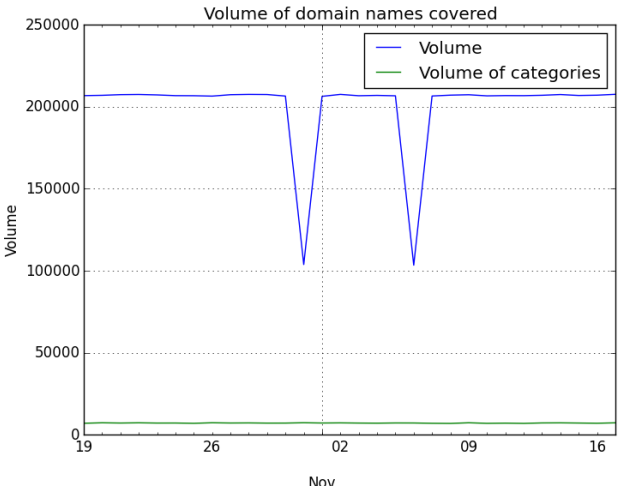

Fig. 6: Volume of domains covered

Therefore, we have the features extracted from our pDNS database of 30 days as shown in Figure 6 for which we consider 19th October as the zeroth day and compare the clusters of other days with it. The first spike in the graph corresponds to an error encountered in the program due to which the crawler stopped and the second spike corresponds to us killing the processes due to massive load on the VM. We also crawl Top Alexa sites based on categories. The categories are Adult, Arts, Business, Health, Home, Kids, News, Recreational, Reference, Regional, Science, Shopping, Society and Sports and each category contains 500 sites.

We also observe the distribution of RHIPs and geographical locations with Alexa ranks as shown in Figures 7-9. Here the bottom 100k domains are represented in the figure from index 100k to 200k. We also show the distribution of RHIPs while crawling Alexa sites according to alphabetically sorted categories in Figures 10-12. For the distribution of IP addresses, we see a sudden drop after initial 500 domains as most of the domains belong to the popular category. However, there are spikes present which indicate the presence of CDNs. The distribution of BGP Prefixes and countries follow a similar trend. The distribution of TTL is almost uniform with random spikes.

B. Analysis

We extract features for 30 days and observe anomalies by comparing clusters of n^{th} day with 0^{th} day. Since the number of anomalies observed are very few in case of categories as there are only 7500 domains and all of them belong to the top domains of Alexa, we can manually inspect the anomalies and label the respective anomalous domains back to their categories. We calculate the moving average of the frequency with which domain names change clusters and register an anomaly on the n^{th} day only in the case where the number of cumulative anomalies till the $n^{th} day$ of that domain crosses the moving average. We observe that over a duration of 30 days, 87 (1.16%) domains are labeled as anomalous out of which 80 are benign anomaly which means that they moved to another benign cluster. 7 domains moved to malicious cluster during the interval of 30 days. We cannot comment upon the exact reason as to why that might have happened in the absence of ground truth. We can however state that such kind of behavior should be of concern and can be due to poor maintenance of infrastructure, DNS Poisoning or Malicious DNS resolution path. Across all the categories, Adult has the maximum number of anomalies and all the malicious anomalies belong to this category. This could be due to the fact that these domain names are not handled in a more professional manner.

Taking Alexa Top100k and Bottom100k as input, we observe that 0.74% (1501) of the domains were labeled as anomalous as shown in Figure 13 out of which 12.25% are malicious anomalies. We again attribute these anomalies to infrastructure changes and in case of malicious anomaly, to DNS poisoning or corrupted resolution.

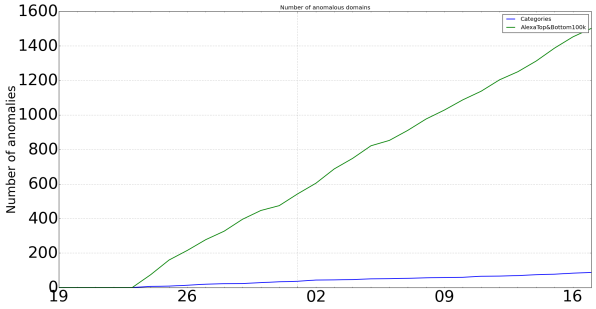

Fig. 13: Number of anomalous domains

We also look at some of the domains that shifted to the zeus cluster and manually about the reputation of these domains. Most of the domains as shown in Figure 14 are labeled as high risk domains by *scamadvisor.com* due to the duration for which these domains are alive and other factors for instance, of them being resolved to a high risk country. We also enumerate some of the benign anomalies that shifted to benign clusters. This behavior could be attributed to poor maintenance of infrastructure on the domain's behalf. We also try to validate our clusters by cross-referencing the IP addresses obtained with FireHOL IP blacklists and find that on average 121 IP addresses are blacklisted in one day. More than 70% of the domains pointing to these IP addresses are in the zeus cluster. We further note that our clusters capture the fast-fluxing ability of domains. If there are some IP addresses in the list which are malicious and do not exhibit such behavior, it won't be labeled correctly by our clustering module. To assess further risk, we look at all the IP addresses in our repository that point

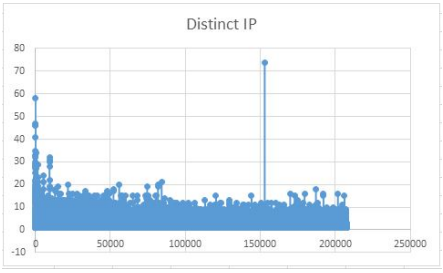

Fig. 7: Distribution of IP address with Alexa rank

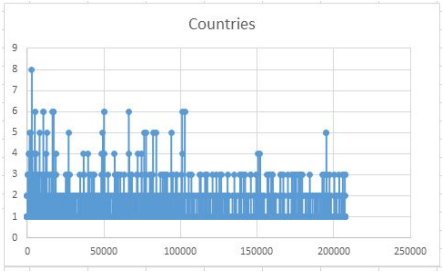

Fig. 8: Distribution of countries with Alexa rank

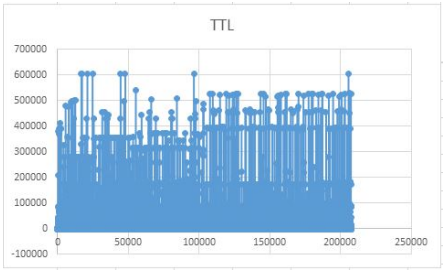

Fig. 9: Distribution of TTL with Alexa rank

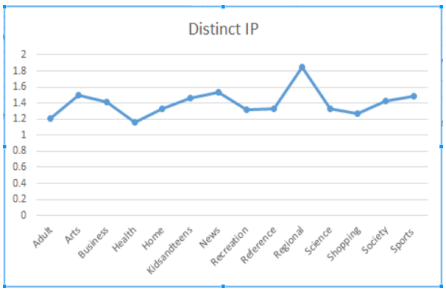

Fig. 10: Distribution of IP address with Categories

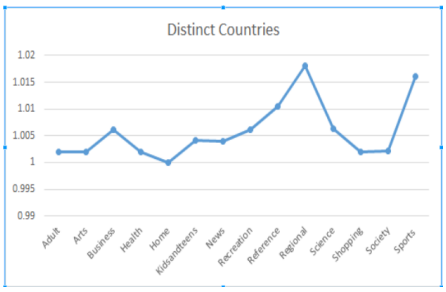

Fig. 11: Distribution of countries with Categories

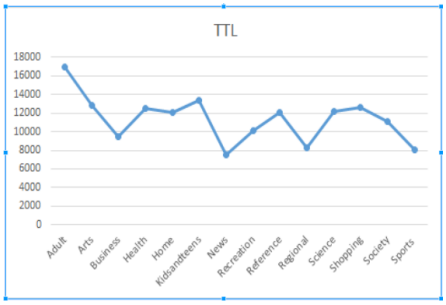

Fig. 12: Distribution of TTL with Categories

to multiple domains in the Alexa list and out of them there are 27 IP addresses that are deemed as malicious by the blacklist as shown in Figure 15.

```
Malicious Anomaly

29350,   bestfunny.video
156489,  vanciuvolentir89.wix.com
181162,  w88wvn.com
16906,   legiaodosherois.uol.com.br
55542,   salamsatudata.web.id
113791,  luvp-afsystem.com
158550,  tkwqnj.com
189588,  kanatawolf.tumblr.com
4122,    alwatanvoice.com
27299,   bjinxiu.net
```

Fig. 14: List of domains moving to zeus cluster

74.220.215.76	['agenciariotech.com', 'iycy.org']
192.107.16.41	['barclaycardus.com', 'findmybarclaycard.com']
173.214.189.213	['hholidayrentals.co.uk', 'hholidayrentals.com']
64.34.157.130	['zorgloob.com', 'vidangefosseseptique.co', 'gobiofrance.fr']
182.18.134.8	['tradewale.com', 'bizzduniya.com', 'loadkaro.com']
103.23.20.237	['cloud.id', 'segitiga.net', 'baitulherbal.com']
66.147.244.111	['foufou.com.tw', 'experience-ancient-egypt.com']
92.63.133.65	['xmlobjective.com', 'mozbot.co.uk']
65.99.237.165	['reversephonelookup.com', 'auto-delovi.org']
176.9.160.209	['irprog.com', 'shoptarh.ir', 'barbodtech.com']
108.168.206.102	['wurthtpms.com', 'livebareilly.in']
116.193.76.133	['mangcapquangfpt.com', 'dienthoaiso.vn']
69.89.31.69	['koalastothemax.com', 'tatasechallenge.org']
157.7.184.20	['wedding-navi.com', 'fishingshow.jp', 'sedori358.com']
69.89.31.104	['mealplanmaven.com', 'iig-llc.com']

Fig. 15: List of malicious IPs pointing to multiple domains

VIII. Discussion

This section discusses the limitations and challenges and also suggests future work that can be implemented to improve the performance of the system. Due to limited pDNS database, we were unable to establish the ground truth which restricts our system to unearth the root cause behind an anomaly. With aggregation of more passive DNS data, the system can be made more robust by developing a statistical supervised classifier. We also note that the passive DNS data collected was from a single exit point in Internet which limits the resolution into the overall snapshot of Internet. We tried to implement TOR and use a different exit node to collect passive DNS data from multiple points but it was difficult to scale the script to crawl domains of magnitude 200,000 in a single day. We leave addressing the scalability aspect for future work. Another challenge with querying a WHOIS database was increase in round trip time of the query certain times in the day which makes the process slow limiting the ability to increase the query rate of domains. Another possible feature overlooked in the system is the number of domains which get 'NXDOMAIN' response on the subsequent days given that they were resolved on 0^{th} day. We have included the capability to record all such instances in our codebase but we haven't included the

observations in our features as we didn't have this data for our initial days. We should also point out that dnspython library imported for the purpose of DNS resolver has the capability to support EDNS0.

IX. Conclusion

We develop an Anomaly detection system by creating a WHOIS pDNS repository and extract features by querying the database to generate clusters according to the domains' network profiles. Depending upon which domains shifted to which clusters, we categorize the anomalous domains as either a benign anomaly or a malicious anomaly. Depending on which type of anomaly was detected on the n^{th} day, we can develop a warning system which raises red flags in events such as DNS Poisoning and corrupted DNS resolution paths. The system has the advantage of generating results fast and does not require a massive database to detect anomalies. It can be implemented at the stub resolver as well as at the DNS recursive resolver. After collecting sufficient amount of data, it can be used to establish ground truth and subsequently will be able to reveal the true reason for anomalous behavior.

References

[1] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. *Building a Dynamic Reputation System for DNS* In the Proceedings of 19th USENIX Security Symposium (USENIX Security 10), 2010.

[2] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. *Exposure: Finding malicious domains using passive DNS analysis*, in Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, February 2011.

[3] Computer Networking: A Top-Down Approach, James F. Kurose, Keith W. Ross

[4] The Dynamic DNS Infrastructure, David Holmes

[5] http://nms.lcs.mit.edu/papers/dns-ton2002.pdf

[6] Scrapy documentation: http://doc.scrapy.org/en/latest/topics/architecture.html

[7] https://tools.ietf.org/html/rfc3912 for WHOIS

[8] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. *From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware*. In Proceedings of the 21st USENIX Security Symposium, 2012.

[9] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou II, and D. Dagon. *Detecting Malware Domains at the Upper DNS Hierarchy*. In USENIX Security Symposium,2011.

[10] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, and W. Lee. *Increased DNS forgery resistance through 0x20-bit encoding: security via leet queries*. In Proceedings of the 15th ACM conference on Computer and communications security,2008.

[11] D. Dagon, N. Provos, C. P. Lee, and W. Lee. *Corrupted dns resolution paths: The rise of a malicious resolution authority*. In Proceedings of Network and Distributed Security Symposium (NDSS '08), 2008.

[12] Y. Chen, M. Antonakakis, R. Perdisci, Y. Nadji, D. Dagon, and W. Lee.*DNS Noise: Measuring the Pervasiveness of Disposable Domains in Modern DNS Traffic*. In Dependable Systems and Networks (DSN), 44th Annual IEEE/IFIP International Conference,2014.

[13] M. Cermak,P. Celeda, and J. Vykopal.*Detection of DNS Traffic Anomalies in Large Networks. In Advances in Communication Networking*.Springer International Publishing,2014.

[14] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, *Detecting algorithmically generated domain-flux attacks with DNS traffic analysis*, IEEE/ACM Transactions on Networking, vol. 20, no. 5, pp. 1663-1677, October 2012.

[15] V. Paxson, M. Christodorescu, M. J. J. Rao, R. Sailer, D. Schales, M. P. Stoecklin, K. T. W. Venema, and N. Weaver. Practical Comprehensive Bounds on Surreptitious Communication Over DNS. In Proceedings of the in the Proceedings of the 22nd USENIX Security Symposium. USENIX, 2013. .

[16] M. Caesar, and J. Rexford. BGP routing policies in ISP networks.Network, IEEE, 2005.

[17] F. Weimer. Passive DNS Replication. In FIRST Conference on Computer Security Incident, 2005.

[18] http://www.inacon.de/ph/data/DNS/DNS-Message-Format/

X. Appendix

We talked about the possibility of using NXDOMAIN responses we get as another feature. On average, 0.175% of the total domains crawled give NXDOMAIN responses. If a particular domain is agile, it is going to change the domains and IP addresses very quickly which will generate NXDOMAIN responses on the botnets that it was resolving to earlier. Therefore, it makes sense to build a feature around NXDOMAIN responses. The list of NXDOMAIN responses observed for top Alexa ranks is shown in Figure 16. The features involving NXDOMAIN would involve a hierarchical clustering used in Pleiades [8] and group together all the domains that generate NXDOMAIN responses. Then by implementing the concept of moving average to establish a threshold, we can filter out the persistent domains that resolve to NXDOMAIN.

Rank	NX_DOMAINS
3007	cpasbien.pw
4046	baimao.com
6945	jz123.cn
9002	anige-sokuhouvip.com
9765	204.12.226.68
11605	applicationdigita.com
13000	saic.gov.cn
13726	magnoft.com
15954	oraclecorp.com
20589	1wmsf.com
21199	wo99.com
22273	jooov.cn
23237	libersads.com
25188	lopsxqvibncpktmtmodesfpeqjtxidodelulsnlh.com
25286	taobaowang.hk.cn
27301	xiaoxiaoshuo.net
28271	uuyx.com
29513	pp55.com
29801	ipk.cn
30985	kredikartiborcutaksitlendirme-tumturkiye.com
33344	xxoose.net
37373	isns.hk
37923	ioage.com
38277	fixget.net
38620	zusun.co.kr
40506	jiekeqiongsi.jimdo.com

Fig. 16: List of top Alexa Ranks generating NXDOMAIN response