A review of current DNS TTL practices

Conference Paper · September 2015

CITATION
1

READS
1,662

3 authors, including:

Ignus Van Zyl
Rhodes University
1 PUBLICATION **1** CITATION
SEE PROFILE

Barry Irwin
Noroff University College
183 PUBLICATIONS **877** CITATIONS
SEE PROFILE

A review of current DNS TTL practices

Ignus van Zyl[1*],Lauren L. Rudman[2*], Barry Irwin[3*]
[*] *Security and Networks Research Group Department of Computer Science Rhodes University, Grahamstown, South Africa*
[1]g11v0032@campus.ru.ac.za,[2]g11r0252@campus.ru.ac.za, [3]b.irwin@ru.ac.za

***Abstract*—This paper provides insight into legitimate DNS domain Time to Live (TTL) activity captured over two live caching servers from the period January to June 2014. DNS TTL practices are identified and compared between frequently queried domains, with respect to the caching servers. A break-down of TTL practices by Resource Record type is also given, as well as an analysis on the TTL choices of the most frequent Top Level Domains. An analysis of anomalous TTL values with respect to the gathered data is also presented.**

I. Introduction

The Domain Name System (DNS) is used to map host names to physical IP addresses [1]. A DNS consists of 3 main components, the domain name space and resource records (RR), name servers and resolvers. According to [2], the domain name space and resource records are "specifications for a tree structured name space and data associated with the names". The name servers hold information on the domain tree's structure and the resolvers extract information from name servers in response to client requests. Throughout the Internet there are many DNS servers, arranged in a hierarchical tree structure with each node having a set of resource records [3]. A host on a network only needs to know the physical address of a name server and the name of a resource in order for it to retrieve the physical address of the resource [4].

Name servers have become a critical resource on the World Wide Web [5] and if a DNS server is unavailable, then a host may not be able to access any resource on the network [1]. This was partially solved by having Primary, Secondary, Tertiary and even Quartiary nameservers [1]. A server's availability can also be affected negatively by large volumes of traffic. This in turn can make a DNS lookup cost longer and decrease performance of networks. This is where DNS caching comes into play [4]. The caching of DNS records is performed to lower the load on DNS servers and decrease the price of DNS lookups. DNS responses are cached for use by later queries. Caching is essential to the success of the DNS system [4]. When a host has both the authoritative and cached information available locally, the authoritative information is preferably used.

Each cached record has time-to-live (TTL) field, which is a 32 bit unsigned integer. It is measured in seconds and is the length of time a resource can be retained in a local cache [6]. Each resource record's TTL field is set by the administrator of a DNS domain. Good practice according to [7] is to initially set the TTLs to high values, and then lower them if a known change will occur. Common TTL values to set can range from anywhere between a day (86400) and a week (604800), but can be set to below a day before a change of the data. This will ensure that the DNS caches are not storing outdated information for too long [7]. However if the TTL of a record is set too low, a server will have an increase in traffic flow with lots of repeat requests and if set too high, new information will not get distributed in a reasonable amount of time. High TTL values can be beneficial as they can help minimise traffic and mask periods of server unavailability [8]. A short TTL can be beneficial because it minimises periods of resource record inconsistency. A TTL value must be chosen to allow information stored in the cache to be as good as the authoritative information while decreasing and load balancing the traffic flow to servers. The lowest value a TTL can have is zero. A resource record with this value will not be cached and therefor force a resolver to query the zone's parent's name servers [9].

According to RFC 1033 [7], which was written in 1987, the minimum for a TTL is a day. Mockapetris states that the recommended TTL value for host names is two days [4] and [10] recommends one to five days as typical values. For records that do not change often a large value like one to two weeks is recommended.

Sometimes administrators mistakenly assign TTLs as if they are assigning priorities to the records, which is bad practice. When administrators expect frequent changes to domain information, they often set the TTL of a record to a low value, even if the changes they make are very rare. In this situation, the TTL should be set high. Setting the TTL too high, one of several years, can create a security concern and can result in the distribution of bad data, which can affect the integrity of the network addressing mechanism [4]. Checking for excessively long TTLs of arriving requests and either discarding them or limiting the TTL to one week is suggested as good practice [11].

This paper reviews TTL practices with two intentions. The first is to gain an understanding of how domain TTLs are currently configured across the DNS infrastructure of the Internet. The second is to gain insight on the possible effects of these TTLs by analysing their configuration. Section II will give a brief overview of other research conducted on legitimate DNS TTLs. Section III will give an overview of the data used for analysis. Data processing techniques as well as an overview of TTL popularity in section IV. A review of corporate TTL choices is also carried out in IV, with an in-depth look at the TTL choices of Google, Facebook and Akamai, a content distribution network. Section V gives an overview of 0 TTL behavior, while the last section concludes the paper and suggests future work on legitimate DNS TTL

analysis.

II. Background

Krishnamurthy et al. [12] proposes that best practice for Content Distribution Networks (CDN) is to set the resource record's TTL low so that the "CDN can change a mapping quickly to facilitate load balancing among its servers". The paper explored how the DNS TTL value can effect the download time and what the benefits versus costs of DNS load balancing has. And that their results indicate that small TTL values used in CDNs does not generally result in better download speeds.

Wills and Shang [13] looks at the effect of caching on the DNS lookup time and the effect of the time-to-live (TTL) value for the cached DNS entries. It was concluded that most DNS queries are handled by the local cache which confirms that choosing the correct TTL is important as DNS responses need to give the correct data.

Chen et al. [14] reviewed the presence of disposable domains in DNS network traffic, noting that the use of disposable domains had been adopted up by anti-spam services such as spamhaus.org and mailshell.net in recent years.

III. Data

The data in question was captured across two productive DNS cache servers that exist in a monitored /24 IPv4 allocation falling within 196/8. Data analysed spans the period 12 January 2014 to 30 Jun 2014.

TABLE I
Dataset overview

Month	No. of unique IPs	No. of unique TLDs	Number of unique TTLs	Total packets in dataset
Jan	57795	112960	2714	7400721
Feb	89208	214254	6790	12496189
Mar	87316	224691	4316	14417630
Apr	57543	114731	2824	5920070
May	14047	16006	362	270504
Jun	65934	145221	2415	11339814

For the purpose of this paper, only responses to the caching servers are considered. This resulted in 51 million reply packets containing DNS domain TTL information, on which our analysis is based.

IV. Data processing and analysis

The dataset was initially preprocessed by top level domain (TLD) to create a new dataset, which represents the summarised practices of the TLD in question, rather than the various sub-domains of the TLD. With respect to this, cc is handled by splitting the domain and then pattern matching the generic TLD to ensure that it, as well as the domain, is preserved. The original dataset was also normalised through the retention of packets with a unique domain and resource record, which served as the filter key. This was done to give a representation of TTL frequency that was not skewed by different querying rates with respect to different domains, and also to mitigate the effect that any malicious domain TTL responses would have on the overall dataset; if they were present. The normalised packet numbers seen in table I represent unique response packets captured in the dataset.

The rest of section IV deals with the results from analysis done on the dataset mentioned, with respect to DNS TTL practices. The first subsection will compare the DNS TTL frequency between the standard and normalised datasets. The second subsection will look at the DNS TTL practices of the most frequent TLD domains. The third will discuss the inherent TTL practices seen for different resource record (RR) types.

A. Frequency of TTL in response packet

TABLE II
Top 10 observed TTL values

Rank	Jan	Feb	Mar	Apr	May	Jun
1	300	300	300	300	300	300
2	60	60	60	60	3600	60
3	3600	3600	20	3600	600	20
4	20	20	3600	20	60	3600
5	600	86400	600	600	900	30
6	86400	600	30	86400	86400	600
7	30	30	86400	30	3200	900
8	7200	900	900	900	20	86400
9	900	7200	1800	1800	30	120
10	1800	1800	3200	3200	1800	1800

Table II presents the frequency of DNS RR TTL values observed across the whole dataset. The data in table II has not been normalised and represents TTL frequency based on total packets and not total unique packets. Of these, 300 is the most frequently sent domain TTL in all of the months. It is expected for low TTLs to appear more frequently as the caching server will query the authoritative server of the domain more frequently. This, however, does not explain the 300 TTL consistently appearing more frequently than lower TTL values.

TABLE III
Top 10 observed normalised TTL values

Rank	Jan	Feb	Mar	Apr	May	Jun
1	300	86400	86400	86400	86400	86400
2	86400	300	300	300	900	300
3	3600	3600	900	3600	3600	3600
4	900	900	3600	900	3200	900
5	7200	14400	14400	14400	300	14400
6	0	7200	0	3200	14400	28800
7	14400	43200	3200	43200	172800	0
8	600	0	43200	7200	43200	600
9	43200	172800	7200	600	600	7200
10	1800	600	600	1800	720	1800

Table III gives a ranking of the frequency of domains in the normalised dataset. The dataset consists of packets that have a unique key *(domain + resource record)*, ensuring that identical domain queries with different record types are not ignored while eliminating TTL frequency generated by multiple queries. As can be observed in table III, 86400 is usually the most frequent unique domain TTL.

It is strange to note however that 300, 600 and 900 appear consistently, indicating a trend among many domain owners

to set much lower TTL values than suggested in older DNS documentation [7]. 0 appears four times in the six month dataset. A TTL of 0 is usually indicative of a fast-flux botnet. The prevalence of value 0 TTLs will be investigated in section V. The normalised packet data for January represents 624955 unique DNS replies with TTL values. January is the only month where more unique domains reported a TTL of 300 as opposed to 86400.

B. Frequent TLDs of the dataset

Table III shows the TLDs responsible for the most queries and responses. The three "aka" domains as well as edgesuite.net make up the Akamai CDN family of servers. spamhaus.org, mailpolice.com and mailshell.net are all RBL filter services [14]. The large Amazon presence is due to their EC2 cloud web-hosting service.

TABLE IV
MOST FREQUENT TLD DOMAINS

Rank	Jan	Feb	Mar
1	akadns.net	akadns.net	akadns.net
2	akamaiedge.net	akamaiedge.net	akamaiedge.net
3	edgesuite.net	edgesuite.net	edgesuite.net
4	mailshell.net	mailshell.net	mailshell.net
5	mailpolice.com	mailpolice.com	mailpolice.com
6	akamaihd.net	spamhaus.org	akamaihd.net
7	google.com	akamaihd.net	spamhaus.org
8	rpdns.net	google.com	google.com
9	amazonaws.com	amazonaws.com	amazonaws.com
10	facebook.com	facebook.com	facebook.com

Rank	Apr	May	Jun
1	akadns.net	akadns.net	akadns.net
2	akamaiedge.net	mailpolice.com	akamaiedge.net
3	edgesuite.net	akamaiedge.net	edgesuite.net
4	skype.net	edgesuite.net	mailshell.net
5	mailpolice.com	spamhaus.org	akamaihd.net
6	akamaihd.net	skype.net	facebook.com
7	spamhaus.org	rpdns.net	instagram.com
8	google.com	akamaihd.net	google.com
9	amazonaws.com	facebook.com	cloudapp.net
10	facebook.com	google.com	amazonaws.com

The subsections of section IV-B are summaries of the TTL activity captured on the network with respect to organisational TTL practices within the corporate sphere.

1) Google: The following data encompasses top level domains related to Google services, and is not exclusive to the google.com domain. This was done in order to present a more comprehensive view of Google TTL practices.

TABLE V
NORMALISED GOOGLE.COM TTL PERCENTAGE FREQUENCY

	TTL values						
Month	86400	300	1800	60	293	3600	Other
Jan	57.554	22.136	11.314	4.990	1.405	1.336	1.265
Feb	54.622	23.336	12.993	4.266	1.487	1.164	2.326
Mar	51.069	23.850	11.212	9.332	1.102	0.648	2.787
Apr	60.980	18.001	8.860	6.975	1.414	1.508	2.262
May	62.393	22.792	3.133	4.843	4.274	1.709	0.856
Jun	48.771	16.530	25.316	4.914	1.117	1.340	2.012

As is seen in table V, roughly 60% of google.com domain TTLs have a length of 1 day. Most of these 86400 TTLs are with respect to subdomains connected to the google-mail infrastrucutre. The 86400 TTL has been recommended in previous RFC's [7], and is rather standard. It is interesting however that Google has such a large low TTL presence with respect to domains. 300 is almost always the second most frequent TTL with respect to google domains. A large portion of this can be attributed to google subdomain AAAA records, almost all of which have a TTL of 300. A large amount of the 1800 TTLs linked to subdomains in the googlevideo.com domain. This TTL is most commonly seen with A and AAAA responses for the aforementioned. The googlevideo.com domain is also responsible for a large amount of 60 TTLs as a result of CNAME queries, which is as a result of it being a CDN. The googlebot.com domain also has a large number of 60 TTL responses for A queries of its subdomains.

The 293 TTL is of interest as it is the only "unexpected TTL" in the sense that it is the only popular TTL that is not divisible by 60. All of the 293 TTL values are from responses to A and AAAA queries to google domains with the aspmx subdomain. These subdomains are identified as values used to configure MX records for Google Apps [1]. On the support page, Google recommends that the TTL be set to 3600 instead, which no doubt comprises some of the 3600 TTL data, but does not explain the low TTL presence with respect to this subdomain. {Could it maybe have been configured manually?}

2) Facebook: As with the Google data, the facebook TLDs include other domains used by Facebook to get a better idea of their overall TTL practices.

TABLE VI
NORMALISED FACEBOOK.COM TTL PERCENTAGE FREQUENCY

	TTL values						
Month	3600	86400	1800	300	900	7200	Other
Jan	47.748	21.622	10.360	5.856	3.603	3.153	7.658
Feb	46.988	30.522	0	6.827	3.212	2.811	9.63
Mar	44.351	30.126	0.837	6.276	3.347	2.510	12.553
Apr	48.438	30.729	0	6.250	2.083	3.125	9.375
May	58.219	28.082	0	5.479	0.684	3.425	4.111
Jun	64.634	15.548	0	5.183	2.134	3.963	8.538

Table VI describes the TTL frequency with respect to Facebook. It is clear that their domain TTL practices favour the 3600 and 86400 TTLs heavily over the others. The sudden dip in 1800 TTLs after January is as a result of channel.facebook.com subdomains switching from 1800 to 86400 in February. This was probably done in an attempt to decrease overall DNS traffic to facebook servers. This shift to decrease overall DNS traffic comes at the cost of limiting distribution of packets between different servers as well as DNS response time with respect to malfunctioning servers. The decrease in 86400 TTLs and subsequent increase in 3600 TTLs is as a result of channel-proxy.facebook.com subdomains now responding with the lower TTL in June. While this TTL is still low, it will not consume as much bandwidth through repeated queries while also increasing the flexibility of server distribution with regards to packet data. While the distribution of the other TTLs remain roughly

[1] https://support.google.com/a/answer/174125?hl=en

consistent throughout the dataset, it is interesting to note that Facebook, like Google, utilise low TTLs frequently in their DNS infrastructure. Of the domains present serving 3600 TTLs, A and CNAME records are the most common.

3) Akamai: Akamai is a content distribution network, and as such utilises low DNS TTL values to enable quick changes to the mapping of resources to aid in load balancing traffic to its servers [12]. Table VII gives a breakdown of the TTL distribution for akadns.net. 95% of the TTLs have a value of an hour or lower consistently throughout the dataset.

TABLE VII
TTL FREQUENCY FOR AKADNS.NET

	TTL values					
Month	120	300	3600	60	30	Other
Jan	58.024	27.802	4.955	3.111	2.103	4.005
Feb	56.881	28.135	5.143	3.225	2.391	4.225
Mar	55.272	28.723	5.299	3.152	2.989	4.505
Apr	57.415	27.813	4.761	3.029	2.193	4.789
May	50.844	32.457	1.313	7.317	3.313	4.756
Jun	54.533	30.131	4.224	3.282	2.986	4.844

akamaiedge.net shows a much greater disparity in TTLs, with around 90% of the TTL values set at 20 seconds, while the other most frequent TTLs are all 3 hours and higher. The two 0% values seen in table VIII are as a result of akamaiedge.net using the TTLs 32400, 21600 and 43200 which account for the 9% of other TTLs. None of these TTLs were recorded in the later months.

TABLE VIII
TTL FREQUENCY FOR AKAMAIEDGE.NET

	TTL values					
Month	20	7200	10800	14400	90000	Other
Jan	89.942	0	0.193	0	0.064	9.801
Feb	90.473	4.295	3.800	1.377	0.055	0
Mar	90.660	4.116	4.169	0.897	0.105	0.053
Apr	89.166	5.249	4.509	0.942	0.067	0.067
May	77.041	11.224	8.673	2.806	0.255	0
Jun	90.945	4.258	3.818	0.881	0.049	0.049

Table IX has an unusually high top TTL. The fact that only CNAME responses were seen for the edgesuite.net TLD does explain the higher TTL configuration.

TABLE IX
TTL FREQUENCY FOR EDGESUITE.NET

	TTL value					
Month	21600	3600	300	360	900	Other
Jan	62.428	31.308	3.405	0.906	0.828	1.125
Feb	65.701	28.294	3.375	0.872	0.731	1.027
Mar	64.984	27.068	5.344	0.806	0.751	1.047
Apr	60.072	30.832	6.319	0.921	0.671	1.185
May	52.800	38.909	4.509	2.109	0.436	1.237
Jun	66.361	28.386	3.065	0.537	0.607	1.044

akamaihd.net is interesting as it only responded with DNS TTLs of 300. 98.5% of these responses were to CNAME queries and the other 1.5% were comprised of A responses.

C. RR and TTL

TABLE X
MOST COMMON TTL FOR RRS IN NORMALISED DATA

Month	A	PTR	CNAME	TXT	MX	AAAA	NS	SOA	SRV
Jan	300	86400	3600	900	14400	300	3600	86400	300
Feb	300	86400	3600	900	14400	300	86400	86400	+
Mar	300	86400	3600	900	14400	300	86400	86400	300
Apr	300	86400	3600	900	14400	300	86400	7200	300
May	86400	86400	3600	900	14400	300	3600	300	300
Jun	300	86400	3600	900	14400	300	86400	86400	300
Dataset	300	86400	3600	900	14400	300	86400	86400	300

+ 86400/300/7200 tied

An interesting pattern emerges from the data in table VI. Of the recieved query responses, the PTR and SOA responses are the only two that show higher TTL values being more prominent. This is not to say that 86400 is not a prominent TTL with respect to other queries, as it ranks almost consistently second for A queries and appears in the top 5 TTL results for most others. The MX TTL results are particularly interesting as the 14400 TTL value seems to have been adopted by many unique domains. The 900 TTL TXT presence is explained by the TTL practices of spamhaus.org, which is responsible for almost all of the TXT queries captured in the dataset.

V. PRESENCE OF 0 TTL VALUES IN DATASET

A surprising number of records were observed within the dataset having a TTL value of 0. These most significant of these are hilighted in table III. Section V-A deals with 0 TTL configurations with respect to the presence of short lived or disposable domains, while Section V-B reviews non-disposable 0 TTL domain activity, which formed the bulk of the records observed.

A. Mailshell

99% of the unique 0 TTL packets in each dataset were generated by mailshell.net. This is as a result of their employment of disposable domains in their service [14]. The 0 TTL is set, in this instance, to ensure that DNS cache servers are not overloaded by creating cached records for multiple thousands of one-use domains, which would severely affect the performance and memory of the DNS caching sever in question. While both spamhaus.org and mailshell.net are both mentioned in [14], spamhaus domains did not result in a noticeable influx of 0 TTL packets .

B. Domains using 0 TTL

The following data reviews the 0 TTL domains that remain after the disposable mailshell.net domains have been filtered out.

TABLE XI
TOP 10 NORMALISED FREQUENT 0 TTL DOMAINS

Rank	Jan	Feb	Mar
1	outlook.com	outlook.com	outlook.com
2	espier.mobi	hichina.com	domobile.com
3	dstv.com	dstv.com	sharesdk.cn
4	nbpush.com	live.com	dstv.com
5	live.com	lyrics007.com	hichina.com
6	sinkdns.org	topnewinfo.cn	dressthat.com
7	supersport.com	supersport.com	sales200.com
8	live.net	live.net	live.com
9	greentreeapps.ro	greentreeapps.ro	joyogame.com
10	domobile.com	export-supply.com	goodphone.mobi

Rank	Apr	May	Jun
1	outlook.com	dstv.com	outlook.com
2	dstv.com	supersport.com	spotify.com
3	spotify.com	outlook.com	dstv.com
4	supersport.com	vitalteknoloji.com	supersport.com
5	live.net	live.net	live.net
6	oldmutual.co.za	perion.com	tedro2.fr
7	live.com	hostdns.ca	oldmutual.co.za
8	greentreeapps.ro	sunbird-images.com	live.com
9	vitalteknoloji.com	scrippsnetworks.com	wwiionline.com
10	miniclip.com	rdikids.org	vitalteknoloji.com

Outlook.com is almost always the leading contributor of 0 TTL responses, not including disposable domains. This is as a result of Microsoft configuring their outbound.protection.outlook.com replies to have a TTL of 0, most likely to prevent an overconsumption of DNS memory. live.com and live.net also fall under the outlook DNS infrastructure. Almost all of these queries are A queries. The three most interesting results here are `dstv.com`, `supersport.com` and `oldmutual.co.za`, not only because of the South African context, but also because all three of them (old mutual to a lesser extent) are among the top 10 contributors to the 0 TTL response traffic. A breakdown of these three domains will be given below.

1) dstv.com: Dstv subdomain responses have TTLs of either 600 or 0. All of the 0 TTL responses are for A queries, and have 18 individual subdomains responding with a 0 TTL.

2) supersport.com: The supersport responses are also all A queries. While there is a positive TTL presence for supersport CNAME queries, all A queries return a 0 TTL for seven subdomains seen in all six months and two subdomains seen in May and June.

3) oldmutual.co.za: The Old mutual domain contains 0 TTLs for six subdomains present in each month, including responses for A queries for `www.oldmutual.co.za`. As with the previous two, all of the 0 TTL queries are A queries.

It was suggested in [9] that a 0 TTL presence indicates that the owner of the domain is planning to change the way their domains are configured, and ensure that the expiring configuration is not cached. This does not seem to be the case with the three domains in question, as they sustain their TTL values throughout the 6 month period. One reason that this TTL value is set to 0 would be that it gives the managing entity of the authoritative server the ability to instantaneously reroute traffic to different servers for each query. While this has the benefit of allowing for maximum data distribution management with respect to servers, it creates a much larger consumption of network bandwidth at the authoritative server, as it is queried every time a query is processed for the relevant domain. Setting a TTL of 0 is detrimental to both bandwidth consumption and load experienced by the authoritative server of the domain [9], as the domain query is forwarded to the authoritative server every time the query is made by an end host, instead of being served by a local cache server.

VI. CONCLUSION

DNS domain TTLs minimise the consumption of bandwidth with respect to the DNS protocol by allowing caching servers to act as pseudo-authoritative servers for a time. As a result, early recommendations for TTL lengths were between 1 day and 1 week. This review shows that there is a strong trend towards using lower TTLs to enable quicker response to downed serves and to allow for more efficient distribution of server load balancing. While low TTLs are common for CDNs, lower TTLs have also been readily adopted by organisations with large IT infrastructures. It would not be surprising to see TTL values decrease even further, as improvements in bandwidth and data processing speed further mitigate the negative effects of low TTLs.

A. Future Work

This work has highlighted high level observations within the data set under study. Further work should be carried out on the following aspects:

- An analysis of the relationship between TTL and refresh times of resource records by the caching DNS servers.
- An exploration as to the geolocation of authoritive Domain Name Servers for top level domains, and the latency involved in queries
- Further analysis of the resource records identified having values either just above or below 'common' values, and the possible benefits or causes for this.

ACKNOWLEDGEMENT

This work was undertaken in the Distributed Multimedia CoE at Rhodes University, with financial support from Telkom SA, Tellabs, Genband, Easttel, Bright Ideas 39, THRIP and NRF SA (TP13070820716). The authors acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the author(s) and that none of the above mentioned sponsors accept liability whatsoever in this regard.

APPENDIX

TABLE XII
RESOURCE RECORD TYPES

Resource record	Description
A record	Returns the IPv4 address for a host of the domain
PTR record	Returns reverse-mapped domain name of IP address
CNAME record	Returns an alias for an existing host given by an A RR
TXT record	Returns generic text associated with domain
MX record	Returns the mail servers for the domain
AAAA record	Returns forward mapping of IPv6 hosts as A does for IPv4
NS record	Returns the authoritative name servers for the domain
SOA record	Returns the key characteristics and attributes for the domain
SRV record	Allows for discovery of services provided by host

REFERENCES

[1] R. Aitchison, *Pro DNS and Bind*. Apress, 2005.

[2] P. Mockapetris, "Domain names: Concepts and facilities," RFC 882, Internet Engineering Task Force, Nov. 1983, obsoleted by RFCs 1034, 1035, updated by RFC 973. [Online]. Available: http://www.ietf.org/rfc/rfc882.txt

[3] J. Bound and Y. Rekhter, "Dynamic updates in the domain name system (dns update)," 1997.

[4] P. Mockapetris and K. J. Dunlap, *Development of the domain name system*. ACM, 1988, vol. 18, no. 4.

[5] M. Colajanni and P. S. Yu, "Adaptive ttl schemes for load balancing of distributed web servers," *ACM Sigmetrics Performance Evaluation Review*, vol. 25, no. 2, pp. 36–42, 1997.

[6] D. E. 3rd, "Domain Name System (DNS) IANA Considerations," RFC 6895 (Best Current Practice), Internet Engineering Task Force, Apr. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc6895.txt

[7] M. Lottor, "Domain Administrators Operations Guide," RFC 1033, Internet Engineering Task Force, Nov. 1987. [Online]. Available: http://www.ietf.org/rfc/rfc1033.txt

[8] J. Damas and F. Neves, "Preventing Use of Recursive Nameservers in Reflector Attacks," RFC 5358 (Best Current Practice), Internet Engineering Task Force, Oct. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5358.txt

[9] M. Larson and P. Barber, "Observed dns resolution misbehavior," 2006.

[10] D. Barr, "Common dns operational and configuration errors," 1996.

[11] P. Mockapetris, "Domain names: Implementation specification," RFC 883, Internet Engineering Task Force, Nov. 1983, obsoleted by RFCs 1034, 1035, updated by RFC 973. [Online]. Available: http://www.ietf.org/rfc/rfc883.txt

[12] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of content distribution networks," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. ACM, 2001, pp. 169–182.

[13] C. Wills and H. Shang, "The contribution of dns lookup costs to web object retrieval," Citeseer, Tech. Rep., 2000.

[14] 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), 2014 44th Annual IEEE/IFIP International Conference on*. IEEE, 2014, pp. 598–609.

Ignus van Zyl received his Honours in 2014, and is currently completing his CS Masters in Information Security at Rhodes University under Barry Irwin

Lauren L. Rudman is in her first year of Masters at Rhodes University researchign in the field of automated Malware Analysis

Barry V. W. Irwin founded and heads the Security and Networks Research Group at Rhodes University. His research interests include passive traffic analysis and National level Cyber defense.