September 17, 2013

Via Electronic Mail to:
comments-name-collision-05aug13@icann.org

Re: Public Comments on Proposal to Mitigate Name Collision Risks, by Google Inc.

Since the publication by ICANN of the Interisle Consulting Group[1] report entitled "Name Collision in the DNS"[2] on August 5th, 2013, the DNS community has engaged in considerable discussion about the topic of potential name collisions as the result of the introduction of new top-level domain names (TLDs). ICANN opened a public comment period that ended on August 27th, and a subsequent reply period that ends today. Most discussion to date has centered around the Day In The Life (DITL) root server data that serves as the foundation data for the Interisle report. However, due to caching built into the Domain Name System (DNS), root server data may not be representative of queries issued by end users.

This document provides data from Google Public DNS, Google's recursive DNS service. Gogole Public DNS serves queries directly from end users, and, therefore, should be more representative of the end user experience. Overall, we find that root server data tends to include a greater proportion of queries for nonexistent TLDs than in our data set, although in some specific scenarios the root server data tends to include a much smaller fraction of queries than the recursive servers do. In addition, we examined the frequency of queries for non-existent domain names for both proposed and existing TLDs and find that some unregistered second level domain names (SLDs) within popular existing TLDs receive significantly more queries than all but a handful of proposed TLDs. Although the root server data provides a useful starting point for the discussion, we believe that considering recursive DNS data should help inform both risk assessment and mitigation strategies in ongoing discussions about potential name collisions.

Google Public DNS

Google Public DNS, commonly known by its IP addresses, 8.8.8.8 and 8.8.4.4, is a very large, publicly available recursive DNS resolver service provided by Google for Internet users to use instead of, or in addition to, the DNS servers provided by their Internet Service Providers (ISP). Launched in 2009, Google Public DNS is intended to offer Internet users a recursive DNS service that is consistently safe and fast. Google Public DNS is committed to adhering to Internet standards related to DNS and does not block, filter, or alter results. Over the years, this

[1] We wish to thank Interisle for their cooperation in helping to understand and react to their report.
[2] http://www.icann.org/en/about/staff/security/ssr/name-collision-02aug13-en.pdf

service has become quite popular and is now used by people throughout the world - as of March, 2013, it serviced over 160 billion queries per day (an average of 1.85 million queries per second).[3]

Methodology

The data presented in this submission covers a three day period - from September 8, 2013, through September 10, 2013.[4] Although only a randomly sub-sampled portion of all queries are logged, during this time over one hundred and fifty billion queries were recorded in the logs for the Google Public DNS service. In fact, this data set is larger than all three of the data sets analyzed in the Interisle report combined.

Importantly, in order to protect the privacy of Google Public DNS users, client IP address is not included in the log data considered as part of this analysis. Similarly, the data presented here is highly aggregated, consistent with Google's privacy policy, and in order to avoid releasing any potentially sensitive information.

In order to better compare frequency of certain classes of queries between different data sets, query counts presented below are represented in "parts per million" (PPM), or how many times the particular query would be seen in a representative set of a million queries. As an example, the frequency at which queries included `com` in the TLD position of the relevant query streams was 219,382 for the root servers versus 495,105 for Google Public DNS, or approximately 21.9% and 49.5% of the totals, respectively. These substantial differences in frequency are largely the result of caching by recursive servers, as described below.

All references to DITL data, whether from the root servers or recursive servers, are simply restatements of data included in the Interisle report. PPM calculations are necessarily imprecise because to determine the PPM, the query frequency has to be compared to the total number of queries in the data set, and the Interisle report contains imprecise expressions of the total number of queries due to rounding of numbers in the description of the data set (e.g., "39 billion" for the total number of queries in the 2013 DITL root server data). As a result, PPM measurements for DITL data are precise to only two significant digits. In many cases, numbers are represented here with only two significant digits to avoid providing a false sense of precision, but where this requires rounding beyond the nearest integer, we have simply rounded to the nearest whole number.

Recursive versus (Authoritative) Root Name Server Data

So far, most of the discussion around potential name collisions as a result of the delegation of new TLDs has centered around data logged by various root servers. This data is useful in that it represents a true global snapshot of DNS traffic; however, it suffers from a number of limitations the authors themselves outlined in Section 4.3 of the Interisle Report. Most importantly, caching

[3] http://googleonlinesecurity.blogspot.com/2013/03/google-public-dns-now-supports-dnssec.html

[4] These dates were chosen to provide some amount of coverage from both the weekend (September 8th) as well as the work week (September 9th and 10th).

plays a key role in providing the DNS with the performance, scalability, and reliability required in order to support the global Internet. As Interisle noted in their report:

> *It has not been possible to tell if a lookup for whatever.newgTLD came from a home user's DSL router or from a name server at a major ISP providing DNS resolver service for millions of customers. Therefore the counts are likely to be distorted because of the effects of caching at intermediate resolving servers. Measuring the extent of that distortion will be very difficult. It would not be possible to compensate for the impact of caching without getting access to a lot of sensitive information from those operating very large resolver farms.*
>
> *This may mean that the counts of how "popular" a new gTLD string is in the current root server traffic could be too high or too low. For instance, millions of users at some ISP might issue lookups for whatever.exampleTLD but this might result in just one query at the root servers. Similarly, a new TLD might appear prominently in this report because of a large number of one-time lookups by resolving servers when in fact there are other proposed TLDs which are much more lowly ranked that are more commonly looked up on the Internet as a whole.*

In this document, we examine data from the world's largest recursive resolver farm to provide data that more closely represent the queries issued by end users than is observable at the root server level, as caching is less likely to have an impact closer to the client. As described above, data is presented in an aggregated manner to address potential sensitivities in the data set.

This analysis is important because caching will tend to reduce the amount of traffic that root servers answer for popular TLDs (popular TLDs are, in turn, made popular by frequently visited websites or other uses of domain names within the TLD; TLDs with popular sites that many people visit will tend to see a greater effect of caching than those that do not), and as a consequence the traffic seen for less popular and non-existent TLDs will tend to represent a greater fraction of the traffic at the root servers than actually issued by users.

Table 1 below shows the incidence of queries for eight selected TLDs across three data sets: the 2013 DITL root server data, the 2013 DITL recursive data, and the Google Public DNS data (all data is reported in PPM):

Root Rank	TLD	DITL - Root	DITL - Recursive	Public DNS
1	com	219382	540173	494995
2	net	129171	154743	154278
4	org	28197	47607	38032
5	home	26128	289	715
6	arpa	21692	147024	89470
15	de	7416	2024	5498
23	corp	3923	339	99
100+	mail	27	677	2211

Although there is some variance between both of the recursive data sources (Google Public DNS and the 2013 DITL recursive server data), some general principles are easily observed.

The root server data significantly understates the portion of user queries related to the top three delegated TLDs[5], com, net, and org, in addition to arpa which is the third-most common TLD within the recursive data sets. The query stream for com and arpa are particularly understated, with both recursive servers seeing more than twice the fraction of queries for com and over four times the fraction of queries for arpa as the root servers do. By contrast, the root server data tends to significantly overstate the fraction of queries for non-existent TLDs, such as those that have been applied for as part of this round of TLD expansion. In the case of corp and home, the fraction of queries is overstated by at least ten times as compared to either of the recursive data sets, and by nearly one hundred times when comparing traffic for home at the root versus that observed in the DITL recursive data set.

Chart 1 shows the overall effect of these differences, comparing the overall fraction of queries for existing, invalid, proposed, and potential TLDs (using the same definitions as in the Interisle report) between the Google Public DNS data and the root server data.

[5] Unfortunately, DITL data for recursive servers is not available (via the Interisle report) for TLDs not either already delegated or currently applied for; this makes a comparison for certain nonexistent TLDs (such as local, the third most common TLD in the 2013 DITL root data) infeasible.

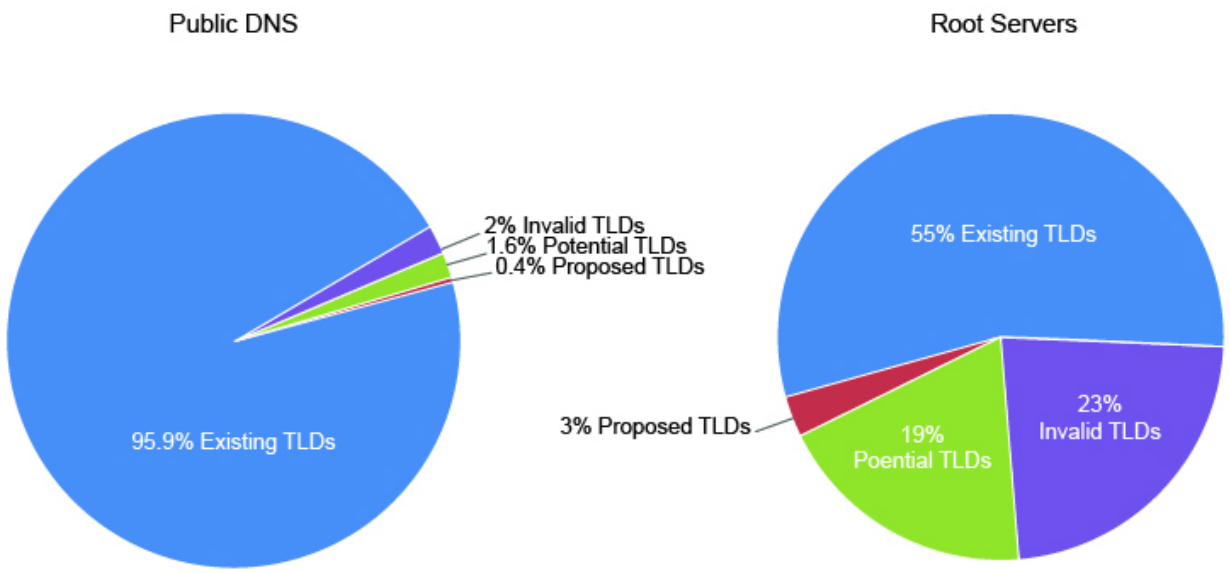

Table 2 shows the Top 100 TLDs[6] queried in the 2013 DITL root server data. This table is modeled on Table 3 from the Interisle report. In addition to the root data analyzed by Interisle, a sixth column has been added to indicate the frequency at which the same TLDs were observed in the Google Public DNS data set (All data reported in PPM).

Rank	TLD	Existing TLD	Proposed TLD	Potential TLD	Public DNS
1	com	219382			495105
2	net	129171			154312
3	local			64137	3550
4	org	28197			38041
5	home		26128		716
6	arpa	21692			89490
7	localdomain			15284	1495
8	internal			13050	138
9	ru	10944			24240

[6] Invalid TLDs that would otherwise qualify are not included in the list.

10	localhost			10623	895
11	cn	10056			23880
12	belkin			9974	197
13	lan			9305	556
14	uk	7901			3663
15	de	7416			5499
16	domain			7067	565
17	jp	6898			4899
18	br	6297			10748
19	info	6288			4204
20	edu	6041			1443
21	au	4036			1244
22	pl	3924			5278
23	corp		3923		99
24	nl	3722			2687
25	router			3593	57
26	tw	3538			1433
27	us	3441			2024
28	dlink			3240	116
29	tv	2860			2917
30	eu	2798			1035
31	fr	2767			2376
32	kr	2664			687
33	at	2485			466

34	ca	2464			1321
35	in	2433			4009
36	gov	2413			6056
37	it	2394			4460
38	biz	2359			3634
39	me	2245			15903
40	cc	2205			2087
41	ua	2111			1968
42	es	2057			1414
43	tr	1776			1638
44	invalid			1710	24
45	co	1691			1307
46	se	1660			801
47	id	1593			592
48	novalocal			1548	27
49	cz	1494			1985
50	ro	1397			1378
51	vn	1361			6644
52	homestation			1340	8
53	null			1284	20027
54	gr	1242			347
55	kg	1241			25
56	loc			1236	79
57	private			1215	25

58	arris			1199	31
59	ch	1175			871
60	mx	1165			745
61	ar	1163			1044
62	hk	1124			652
63	notinuse			1106	0
64	intra			1079	31
65	za	1073			560
66	bind			1069	17
67	be	990			798
68	gprs			977	0
69	nz	923			214
70	dk	913			890
71	dom			900	29
72	il	893			418
73	sg	838			241
74	pt	766			422
75	no	755			416
76	hu	743			700
77	cl	741			587
78	mil	738			144
79	html			730	16
80	sys			711	7
81	my	653			1039

82	sk	643			502
83	th	628			641
84	fi	625			335
85	tendaap			620	14
86	gateway			613	83
87	none			595	66
88	ws	569			384
89	ph	550			187
90	actdsltmp			542	12
91	server			530	29
92	pri			529	11
93	su	512			1736
94	intranet			510	37
95	ice		508		1
96	pvt			503	13
97	lt	500			145
98	la	493			912
99	minihub			492	0
100	asus			484	17

Table 3 presents data on the top 100 applied-for TLDs based on frequency in the 2013 DITL root data set, with PPM numbers for both the 2012 and 2013 DITL data sets (which is Table 4 in the Interisle report). The seventh column includes data from Google Public DNS servers with the incidence of these same TLDs.

Proposed TLD	2012 rank	2013 rank	2012 PPM	2013 PPM	Public DNS PPM
home	1	1	10819	24434	714
corp	2	2	2233	3705	99.2
ice	21	3	33	507	0.77
global	4	4	197	317	6.32
med	29	5	23	277	1.29
site	3	6	237	275	19.8
ads	5	7	142	271	7.34
network	12	8	80	223	17.2
group	7	9	118	220	5.94
cisco	9	10	95	212	12.7
box	8	11	112	197	49.4
prod	14	12	75	180	13.8
iinet	6	13	139	139	1.37
hsbc	10	14	90	135	1.39
inc	11	15	84	134	4.06
win	18	16	46	133	1.25
dev	13	17	79	130	901
office	15	18	70	103	6.56
business	20	19	35	84	0.39
host	16	20	54	80	11.7
star	31	21	19	62	3.96
mail	25	22	27	61	2211
ltd	19	23	36	51	1.43

google	23	24	30	48	4.32
sap	169	25	2	44	1.11
app	17	26	47	44	9.44
world	27	27	26	42	1.14
mnet	30	28	21	40	0.96
smart	26	29	27	34	0.06
web	33	30	15	29	6.54
orange	32	31	17	27	17.5
red	24	32	29	27	1.23
msd	43	33	10	25	0.17
school	37	34	13	22	0.72
bank	39	35	11	20	0.45
casa	28	36	23	20	1.10
telefonica	45	37	9	20	0.36
zone	51	38	8	18	1.56
movistar	118	39	3	17	0.11
search	82	40	5	17	3.16
abc	41	41	10	17	0.70
llc	48	42	9	15	0.26
youtube	34	43	14	15	1.35
samsung	219	44	1	15	0.13
tech	68	45	5	14	0.29
hot	55	46	7	14	0.12
you	44	47	10	14	0.63

ecom	46	48	9	14	0.35
hotel	52	49	8	14	1.11
off	54	50	8	13	3.25
cloud	119	51	3	13	1.03
foo	62	52	6	13	0.12
new	36	53	13	13	3.13
bcn	93	54	4	13	0.37
free	81	55	5	13	0.20
top	53	56	8	12	0.29
one	63	57	6	12	0.44
bet	91	58	4	12	0.27
kpmg	949	59	0	12	0.00
wow	69	60	5	12	0.08
yahoo	47	61	9	11	11.7
blog	56	62	7	11	0.13
work	49	63	8	10	0.46
chrome	110	64	3	10	0.24
data	84	65	5	10	10.2
link	22	66	32	10	0.33
comcast	40	67	11	9	0.20
cam	80	68	5	9	5.16
gold	151	69	2	9	0.43
medical	67	70	6	9	0.17
live	75	71	5	9	2.67

art	77	72	5	9	0.14
olympus	66	73	6	9	0.24
city	73	74	5	9	0.24
sew	76	75	5	9	0.07
lanxess	60	76	7	8	0.14
center	106	77	3	8	0.28
ifm	99	78	3	8	0.05
law	87	79	4	8	0.38
goo	85	80	5	8	0.20
plus	141	81	2	8	0.25
apple	64	82	6	7	0.22
zip	96	83	3	7	0.20
gmail	117	84	3	7	2.77
house	38	85	12	7	0.23
company	95	86	4	7	0.11
itau	83	87	5	7	0.27
thai	131	88	3	7	0.11
show	74	89	5	7	0.31
college	153	90	2	7	0.22
taobao	155	91	2	7	0.18
amazon	152	92	2	7	4.72
schule	65	93	6	7	0.07
pub	127	94	3	6	0.57
bom	124	95	3	6	0.07

ibm	50	96	8	6	0.08
ericsson	105	97	3	6	0.03
vet	109	98	3	6	0.25
here	101	99	3	6	0.60
man	112	100	3	6	0.21

These tables make clear that the pattern observed above, that queries for nonexistent domains represent a much smaller fraction of traffic when considering user queries to recursive servers than in the query stream visible at the root servers. Indeed, with the exception of two TLDs: dev and mail (which we will discuss in more detail below) every proposed TLD in the top 100 of the root server data sees a greater proportion of queries at the roots as compared to the Google Public DNS data. The magnitude of this effect varies, but on average it tends to overstate the proportion of queries related to the proposed TLDs by nearly an order of magnitude.

"Dotless" Domains and Caching

There is an important exception to the general trend of root server data showing a larger proportion of queries for nonexistent domains. The undelegated mail and dev TLDs sees significantly more requests in the recursive data sets than at the roots. This occurs because of a fairly unique pattern of queries related to mail and dev combined with a difference in the way the DNS handles caching in the case of errors such as NXDOMAIN (negative caching) versus the case where a valid result is returned (positive caching).

Unlike most TLDs, many of the requests related to the mail and dev TLDs are for the "dotless" domain,[7] properly represented in the DNS as "mail." In fact, within the Google Public DNS data set, over 99% of all requests for both TLDs are for the dotless domain name. Other popular TLDs such as corp and home see less than .2% of all requests for the dotless domain, with over 99% including at least a SLD. This is significant because when the root servers return an NXDOMAIN response to a query for a nonexistent TLD, the response is specific to the exact request, which is being issued. Recursive name servers will cache this response so that they do not need to query the root servers again for the same request for up to 24 hours. However, if they receive a different query (for example, for a different SLD), the name server will issue another request to the root servers and receive another NXDOMAIN response. In other words, if a recursive resolver receives a request for foo.example shortly before it receives a request for bar.example, it will still have to issue two separate queries to the root servers and receive two different responses.

[7] This is not unexpected; it is likely that some users have mail clients configured to contact the host "mail", which may resolve in some contexts using a search path, but also has the potential to be interpreted as queries for the mail TLD.

This is in contrast to requests for already delegated TLDs, for which the root servers respond with a set of TLD nameservers that can be cached by the recursive resolver and used for future requests for that particular TLD. Once a user issues a query for `foo.com`, the recursive resolver will have a cache with the `com` nameserver records that it can use for the next 24 hours, so subsequent requests for `bar.com`, `google.com`, or `mylittlepony.com` would not require any additional requests to the root. In the case of `mail`, where almost all of the queries are for the same domain name, caching is much more effective so it does not suffer from most nonexistent TLDs' impediments to caching.

As a result of these effects, TLDs that have a very large fraction of requests for the dotless domain, or any specific domain name within the TLD, will see a lower proportion of queries for that TLD at the root level versus the recursive servers, so analyses based on root server data may overlook some high traffic TLDs. At the same time, risk mitigation for potential name collision in these TLDs is likely fairly straightforward — prohibiting registrations of relevant subdomains (ICANN already prohibits the use of dotless domains in proposed TLDs) would ensure that NXDOMAIN responses continue to be served for these requests, continuing the behavior that exists today for these queries.

Table 4 shows the top 10 proposed TLDs based on number of queries observed in the Google Public DNS data set with data on the proportion of queries for the TLD alone versus queries for subdomains. In addition, it includes the total number of SLDs observed in queries for the TLD, as well as the percentage of the total queries for the TLD represented by each of the top three SLDs. Finally, it includes the total number of SLDs that combine to represent 99% of the total traffic volume for the TLD.

The `management` TLD provides a simple example. Only .1% of the total queries are for the dotless TLD; 173 SLDs were present in the remaining queries; the top three of these, `mail.management`, `cpe.management`, and `system.management` represent 94%, 6% and .02% of the total queries respectively; and the top 2 SLDs cover over 99% of the total queries for the TLD.

TLD	PPM	TLD Only	Total SLDs	SLD #1	SLD #2	SLD #3	SLDs for 99%
mail	2211	99.8%	4350	0.10%	0.04%	0.005%	0[8]
dev	901	99.2%	5513	0.1%	0.1%	0.04%	0
home	714	0.2%	10343261[9]	17%	10%	9%	10025478
corp	99	0.2%	18076	9%	5%	5%	3772
management	77	0.1%	173	94%	6%	0.02%	2
box	49	2%	1967	96%	0.5%	0.4%	9
site	20	1%	22126[10]	19%	15%	8%	2511
orange	18	0.4%	1153	95%	4%	0.2%	2
network	17	62%	7894	14%	6%	3%	224
prod	14	0.1%	6246	13%	10%	8%	155

This data demonstrates that for many TLDs, the vast majority of traffic is associated with a small number of SLDs. In these cases, it seems likely that much of the possible risk involved with delegating the TLD could be mitigated simply by understanding and possibly reserving these SLDs.[11]

At the same time, many TLDs exhibit a very long tail of SLDs with only a small number of queries each. The most striking example of this phenomenon is home with 99.8% of the total SLDs observed within the TLD receiving only one query each; however, other TLDs, such as site (with 67% of SLDs receiving only one query each) exhibit the general pattern. A large fraction of these queries are for the random ten character strings described in Section 5.4.3 of the Interisle report. These queries are the result of the Chrome browser attempting to determine whether the computer's DNS configuration returns incorrect results for nonexistent domains; Chrome issues three separate queries to make this determination, so even in the unlikely event one of these strings were registered and allowed to resolve, the user would not suffer any adverse effect.

[8] Note that because the ICANN Board has recently disallowed dotless domain names, queries for the TLD alone are counted towards the total traffic needed to to cover 99% of queries before any SLDs are considered. As a result in the case of mail and dev, over 99% of queries are covered before a single SLD's queries are counted.
[9] 10321801 of the SLDs within home, or 99.8% of the total, received only a single query.
[10] 14866 of the SLDS within site received only a single query.
[11] This mitigation does introduce a dependence on the TLD name server infrastructure.

Incidence of NXDOMAIN responses

There has been considerable speculation about the incidence of NXDOMAIN responses in existing TLDs versus proposed TLDs. Table 5 shows the top 20 TLDs included in queries that resulted in NXDOMAIN responses within the Google Public DNS data set.

TLD	NXDomain PPM
arpa	54424
com	27098
net	11923
org	11656
ru	4366
local	3541[12]
mail	2208
localdomain	1494
kz	1270
in	981
br	980
info	976
unifi	940
cn	917
dev	901
localhost	894
home	715
wpad	593

[12] Because a small fraction of requests for non-existent domains returned statuses other than NXDOMAIN (notably SERVFAIL), numbers for nonexistent domains reported in this table are slightly lower than in previous tables.

biz	590
to	566

Although this data demonstrates that some existing TLDs are responsible for over an order of magnitude more NXDOMAIN queries than any proposed TLD, it is difficult to make direct comparisons between these numbers. Many of the queries resulting in NXDOMAIN responses for com, for example, are the result of lookups related to existing, high-traffic services such as yahoo.com, msn.com, google.com, h33t.com, and h3q.com.[13] NXDOMAIN requests for these domains pose little risk, as even if the domains in question begin to resolve, users will likely be directed to servers administered by the organization that they intended to reach.

However, in some cases, a large number of NXDOMAIN responses were recorded for specific SLDs that are not currently registered. Several unregistered com SLDs were responsible for millions of queries each;[14] the unregistered com SLD generating the most traffic was responsible for approximately 28 PPM of the query stream, which is a larger fraction of queries than all but six of the proposed TLDs. As Eric Osterweil of VeriSign recently observed, it is perhaps unlikely that system administrators will intentionally configure internal systems to make use of unregistered subdomains within valid TLDs, but it is easily possible that various systems could have unanticipated dependencies on unregistered domain names, as evidenced by VeriSign's own launch of the SiteFinder service in 2003.[15] More importantly, names that were previously registered and are subsequently re-registered pose opportunities for phishers, spammers, or other bad actors who can take advantage of the existing traffic and reputation of the domain name, potentially even masquerading as the previous registrant. While these risks may not be identical to those posed by undelegated TLDs, under some circumstances they may actually pose greater threats to end users.

Sincerely,

Ben Fried
Vice President and Chief Information Officer
Google Inc.

[13] h33t.com and h3q.com appear to be popular BitTorrent trackers.
[14] Because these unregistered com SLDs can be registered by anyone without any of the scrutiny provided by the new gTLD application process, we are not including the specific list of SLDs in this document. We would be happy to share specific findings with responsible interested parties.
[15] www.icann.org/en/groups/ssac/report-redirection-com-net-09jul04-en.pdf