Leakage of .onion at the DNS Root: Measurements, Causes, and Countermeasures

Aziz Mohaisen, *Senior Member, IEEE, Member, ACM*, and Kui Ren, *Fellow, IEEE, Member, ACM*

***Abstract*— The Tor hidden services, one of the features of the Tor anonymity network, are widely used for providing anonymity to services within the Tor network. Tor uses the .onion pseudo-top-level domain for naming convention and to route requests to these hidden services. The .onion namespace is not delegated to the global domain name system (DNS), and Tor is designed in such a way that all .onion queries are routed within the Tor network. However, and despite the careful design of Tor, numerous .onion requests are still today observed in the global DNS infrastructure, thus calling for further investigation. In this paper, we present the state of .onion requests received at the global DNS and as viewed from two large DNS traces: a continuous period of observation at the A and J DNS root nodes over a longitudinal period of time and a synthesis of Day In The Life of the Internet data repository that gathers a synchronized DNS capture of two days per year over multiple years. We found that .onion leakage in the DNS infrastructure to be both prevalent and persistent. Our characterization of the leakage shows various features, including high volumes of leakage that are diverse, geographically distributed, and targeting various types of hidden services. Furthermore, we found that various spikes in the .onion request volumes can be correlated with various global events, including geopolitical events. We attribute the leakage to various causes that are plausible based on various assessments, and provide various remedies with varying benefits.**

***Index Terms*— DNS, privacy, security, Tor.**

I. Introduction

THE Domain Name System (DNS) is one of the pillars of the Internet today, serving as a directory for domain names. In essence, DNS is mainly used for mapping domain names, names that are easily accessible by humans, to Internet Protocol (IP) addresses of Internet resources, such as web servers, mailing hosts, and other online services. The DNS is a hierarchical naming system for computers, services, and resources connected to the Internet, where the top of the hierarchy is the DNS root. The hierarchical nature of the DNS creates certain levels of dependencies between various administrative domains, and the resolution of a single domain name would require collaborations among those domains. For example, `www.example.com.`—when resolved recursively, would require the collaborations of the root of the DNS, the authority server for the top level domain (TLD) of `com`, the authority server for the second level domain (SLD) of `example`, and the authority server of the third level domain `www`. Currently, the root consists of a combination of 13 groups of DNS servers located globally around the world. Each of those servers is named in the form X.root-servers.net, where X is a character in the range of A through M. These roots are responsible for the delegation of top-level-domains (TLDs) such as .com [45].

It is well known within the Internet research and engineering community that many installed systems on the Internet query the DNS root for a wide range of TLDs that are non-delegated and will ultimately result in an error, which is commonly referred to as an NXDomain [57]. Many of these installed systems depend implicitly or explicitly on the indication from the global DNS that the domain name does not exist for their operation. For instance, many internal networks use a domain name suffix list that is not currently delegated in the global DNS, such as .corp and .home [21]. In both of those examples, the non-delegated suffix is important for networks operation and is mainly used for naming and resources discovery within those networks.

Due to the recent delegation of new gTLDs within the global DNS [4], several studies have measured the amount of internal namespace leakage to the DNS roots [3], [32]. These unintended leaked DNS queries have been shown to expose sensitive private information and present potential new security threat vectors [3], [32], [58]. During the analysis of potential colliding name spaces within the global DNS, queries suffixed by .onion appeared to be one of the more prevalent non-delegated TLDs at the global root DNS, which triggered our initial exploration and study.

Tor [26] is one example of a system that exploits the absence of a non-delegated namespace within the global DNS system for its internal use and operation. While Tor in general provides anonymity to users, hidden services, a unique feature within Tor, provide additional anonymity for servers running as hidden services. To identify these services and route requests associated with them, Tor uses the .onion namespace [64]. Hidden services today are widely used, due to their advantage of concealing the location of servers providing such services and making their takedown hard to conduct without the collaboration of Tor or using very sophisticated attacks [15], [40], [61]. Services that want to conceal their location are not limited to illicit services and underground forums and marketplaces (although many of the services that use Tor hidden services are of such nature [49]), but also include popular services, such as social networks, including Facebook.

Manuscript received March 21, 2016; revised September 21, 2016; accepted June 13, 2017; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Chen. This work was supported in part by the NSF under Grant CNS-1643207 and in part by the Global Research Laboratory Program of the National Research Foundation (NRF) through the Ministry of Science, Information and Communication Technologies and Future Planning under Grant NRF-2016K1A1A2912757. A preliminary version of this work has appeared in ACM WPES 2014 [69]. *(Corresponding author: Aziz Mohaisen.)*

A. Mohaisen is with the Department of Computer Science, University of Central Florida, Orlando, FL 32816-2362 USA (e-mail: mohaisen@ucf.edu).

K. Ren is with the Department of Computer Science and Engineering, The State University of New York at Buffalo, Buffalo, NY 14260-2500 USA.

Digital Object Identifier 10.1109/TNET.2017.2717965

Furthermore, recently hidden services have been suggested as a tool for seizure-resistant top-level domain name design [55], highlighting the variety of useful applications they support.

There exists a clear conflict of interests between the internal namespace routing in Tor and the global DNS namespace when .onion URLs are shared or requested [30]. In fact, DNS leakage is a known and well-documented issue within the Tor community, although potentially for many reasons, and is not systematically studied. For example, many tutorials on the Tor website have been published giving users instructions on how to mitigate the leakage through the use of proxies, by disabling DNS prefetching within the browser or even by installing a local DNS server which rejects .onion addresses [29]. However, non-technical Tor users likely do not practice these mitigation steps due to their complicated nature, or because of their unawareness of those remedies (c.f. §IV-A).

The leakage of .onion requests to the global DNS roots clearly presents some risk to Tor users and also has implications that need to be explored. To this end, in this paper we present a first look at the .onion leakage at the DNS root. We use two root servers, A and J, that are operated by Verisign, and explore .onion resolutions seen at both of them over a period close to six months. We complement this measurement with a data set from all root servers over seven years, with a sample of two days per year, and highlight the persistent and growing trends in .onion leakage. Finally, we explore root causes, and highlight potential remedies.

Main Highlights

Our findings in this study have various interesting highlights, including the following. First, a large number of .onion requests for a variety of SLDs are observed at both the A and J root servers, as well as other roots observed in the DITL data set, and the requests originate from a diverse set of locations (at the recursive name server level). The requests are persistent over time, and their volume is increasing suggesting the relevance of the leakage as a phenomenon. Second, surges in the amount of .onion traffic and leakage into the public DNS coincide with major global, geopolitical and censorship-related events, which are easily observed in the leaked queries. Surgical analyses of those queries highlight various local trends in those events. Third, the .onion's traffic exhibits a heavy tailed distribution (with respect to the number of queries per .onion), and a very interesting weekly traffic pattern observed at a high granularity of source attribution, as consistent with other online services. Fourth, while the exact root causes are not easy to verify with certainty, we highlight various plausible causes of the leakage supported by various analyses and user studies. Fifth, we suggest various remedies based on our data analyses and provide a preliminary evaluation of their effectiveness and complexity.

Contribution

The main contributions of this work are as follows. We perform the first systematic and large scale measurement on the leakage of .onion pseudo domain names in the DNS infrastructure. We find that there is a large number of .onion domains that leak to the DNS root. Second, we explore the root causes of such leakage, and attribute that to various plausible reasons confirmed partly by various studies and cross-validations. Third, we study the implications of such leakage on the user privacy, and initiate for that line of work. Finally, we explore potential fixes to the problem of .onion leakage and associated cost.

Organization

The organization of the rest of this paper is as follows. In §II, we introduce the DNS profile of the .onion data collected. In §III, we examine longitudinal patterns of .onion traffic to the A and J root servers operated by Verisign from various network and second-level-domain (SLD) points-of-view, and highlight correlations between global events and increased .onion traffic volumes. In §IV, we explore potential reasons .onion traffic is being leaked to the roots. In §V, we highlight considerations within the Internet engineering community to address the use of non-delegated TLDs, including implications and remedies. In §VI, we discuss the related work. Finally, in §VII we will present our conclusions and discuss future directions in which we will further explore the .onion leakage.

II. Data Sets

In this paper we use several data sets and rely on various supporting studies for conducting this work. The first data set is from the resolution at the A and J root servers operated by Verisign, while the second data set is the "Day In The Life of the Internet" (DITL) managed by the Domain Name System Operations Analysis and Research Center (DNS-OARC). In the following we elaborate on those data sets and their nature. Other data sets, including a crawl of the domain names under the .com and .net TLDs are described briefly where they are used (c.f. §IV).

A. Roots A and J Data Set

As we mentioned earlier, the Internet root name servers consist of 13 identical and geographically distributed servers operated by different organizations. Among those 13 root servers, Verisign operates the A and J root servers in the DNS root zone. NXDomain (NXD) responses for the non-delegated TLD .onion were captured over slightly more than *six months* from both root servers starting on September 10th, 2013 and ending March 31st, 2014. The data set consists of approximately 27.6 million NXD records spanning 81,409 unique SLDs. The DNS requests originated from a wide variety of sources: in total, they are sent from 172,170 IP addresses, 105,772 unique /24 net blocks, and 21,345 distinct Autonomous System Numbers (ASNs).

During the multi-month collection period, numerous NXD TLDs appeared at the roots. Based on the total query volume, we ranked the various TLDs and found that the .onion TLD ranked 461 out of 13.8 billion TLDs. We further depict the traffic patterns and trends observed in the .onion TLD in §III.

B. DITL Data Set

The DITL data set is managed by DNS-OARC, and is a joint effort with CAIDA and ISC. The data captures synchronized

TABLE I
DITL DATA SET—ROOT SERVERS CHARACTERISTICS

Year	# roots	Root servers	# queries
2008	7	(a,c,f,h,k,l,m)	3,710
2009	8	(a,c,e,f,h,k,l,m)	13,343
2010	13	all	2,371,869
2011	11	all except b and g	691,385
2012	10	all except b, d, and g	693,524
2013	11	all except b and g	1,371,650
2014	9	all except b, d, g, and l	1,705,247

TABLE II
SUMMARY OF TRAFFIC FROM ROOT SERVERS

Root	Organization	# queries	# years	Traffic (%)
A-root	Verisign	515,107	7	7.52
B-root	USC-ISI	97,119	1	1.42
C-root	Cogent	723,152	7	10.56
D-root	UMD	205,403	3	3
E-root	NASA	151,014	6	2.2
F-root	Internet Sys	763,663	7	11.15
G-root	Defense Sys	72,232	1	1.05
H-root	US Army	360,490	7	5.26
I-root	Netnod	975,579	5	14.24
J-root	Verisign	842,361	5	12.3
K-root	RIPE	733,951	7	10.71
L-root	ICANN	649,648	6	9.48
M-root	WIDE	761,009	7	11.11

and periodic measurements and data collection effort by root name server operators and other organizations (e.g., ISPs). The data set covers traffic capture of DNS resolution for a period of two days every year. While the data set captures traffic at the recursive level as seen by various organizations participating in the DITL data collection effort, we only focus on the root traffic. We do that to establish a guideline on how representative the data set obtained from the A and J root servers is, and to highlight the overall trends of .onion in the DNS over time. Furthermore, while leakage of .onion to the recursive is still a privacy threat, we believe measuring such leakage and characterizing it is an orthogonal work. We also exclude this part of work for ethical reasons.

In total, the DITL data set covers 8 years (from 2008 to 2015), with two days worth of traffic for each year. For most of this study, we use the first 7 years of DITL (from 2008 to 2014), and use the last year to confirm the persistence of .onion leakage. The data set captures traffic from all root servers (A through J), however not all root servers are present in all years, as shown in Table 1. For the years of DITL data set, we found 6,850,728 .onion queries for 18,330 unique .onion SLDs. The various queries are originated from 331,816 IP addresses distributed over 268,616 /24 subnetwork addresses. The share of each root of the .onion queries as a number and a percent are shown in Table 2 with further information on each root (including the operator).

III. CHARACTERISTICS OF .ONION LEAKAGE

To understand the leakage of .onion in the public DNS infrastructure, we rely on the two aforementioned data sets and provide an in-depth analysis. The main thrusts of analysis based on the A and J root node data set are as follows: 1) .onion traffic volume and diversity measurements in §III-A, 2) .onion strings (SLD) measurements and analysis in §III-B, 3) traffic source analysis of .onion queries in §III-C, and 4) a correlation between global cyber and geopolitical events on the one hand and trends in the volume of the leaked .onion strings on the other hand in §III-D. We complement those thrusts by studying and analyzing volumetric trends of .onion leakage from the DITL data set in §III-E.

A. Traffic Volume and Diversity Measurements

To better understand the overall traffic pattern, we conduct a longitudinal measurement of query volumes and diversity measures.

For this measurement we use three metrics of varying levels of granularity: 1) the raw number of total .onion requests, 2) the total number of the distinct slash 24 (/24) subnetwork addresses and 3) the total number of distinct autonomous systems from which the .onion queries are leaked. We identify autonomous systems by their numbers, and use ASNs and ASes to interchangeably refer to the autonomous systems and their numbers in the rest of this paper. For the /24 subnetwork addresses, we simply discard the least significant block of the IP address leaking the .onion query, and aggregate the number of requests per /24 subnetwork address. For ASN association, we map IP addresses to their home AS using an off-the-shelf commercial-grade mapping service [25].

Results of the three metrics are shown in Figure 1 for the A and J root data in §II-A. Overall, we notice a substantial number of leaked .onion queries to the public DNS infrastructure, represented by the A and J root servers, thus supporting prior anecdotes on .onion leakage. In particular, from Fig. 1(a) we observe that during the period covered by our measurements the numbers of queries leaked were more than 70,000 queries per day. Notice that this number represents a lower bound on the leaked queries from end-users and the actual number of leaked queries might be substantially larger. In particular, given that negative caching—in which negative resolution results are cached by DNS resolvers—is widely deployed today for resolution efficiency, some queries might not be sent to the root and resolutions are likely to be performed using previously cached answers. Second, we observe abrupt spikes across the three different metrics. Third, we observe the fast uptrend of growth across the three metrics, by more noticeably with the raw queries and /24 subnetwork characterization. Fourth, we notice a diurnal pattern, especially observed at a higher level of granularity, e.g., /24 subnetwork and ASN. In the following we elaborate on each of those findings.

Raw Queries: Growth Trends: While we use data that spans about six months, a period that is relatively large to characterize local trends in the leaked .onion queries, we also observe global characteristics on the growth of the leakage that are both interesting and alarming. For example, also in Fig. 1(a), we observe that the total number of queries substantially grows over the relatively short period of time: compared to the 70,000 queries initially observed at the start of the study, we observe a steady growth to more than 200,000

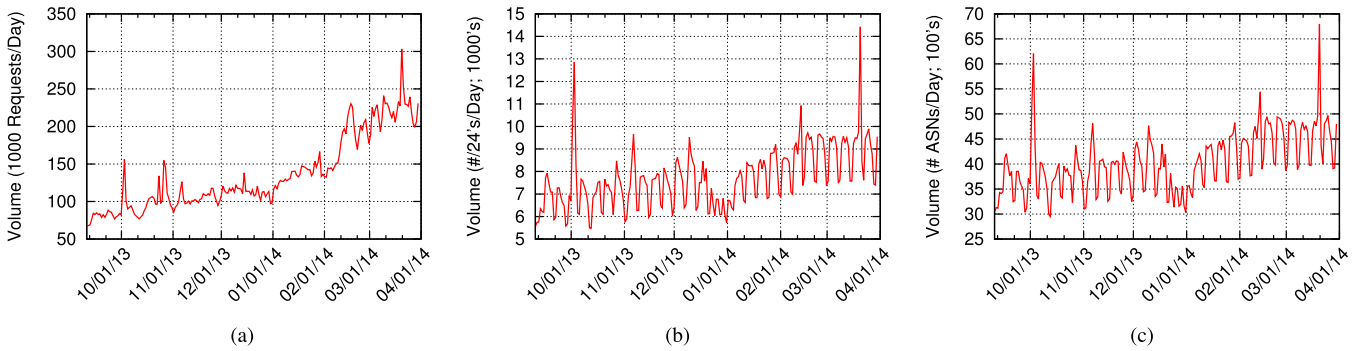

Fig. 1. The .onion traffic measurements observed at the root DNS nodes A and J. Notice the uptrend in the raw number of queries (in (a)), and the consistent (and slightly uptrend) in both the number of /24 networks (in (b)) and the number of ASNs from which queries are issued (in (c)).

queries per day towards the end of the study, corresponding to about 200% of growth over the initial number of leaked queries. However, such growth is even more overwhelming at peak times, with anomalies resulting in a growth of more than 300% over the initially observed queries. We notice, however, that those anomalies are repeated events, and explore their explanation in §III-D. Also, we observe that the measurement of the raw queries does not reveal any clear diurnal patterns, which is surprising.

The /24 Subnetworks: Diurnal Patterns and Growth: One common characteristic that DNS services—as well as many other online services—exhibit is the daily and weekly repeated patterns [24], [34], [42]. In particular, the main hypothesis accepted by many researchers and operators is that DNS queries and resolutions follow repeated daily patterns due to the operation cycles of systems, and use pattern of users. From our raw query measurements, we cannot find any obvious pattern. To this end, we focus on higher granularity characterizations, i.e., /24 measurements. The /24 plot is shown in Figure 1(b). These patterns and trends are clear in ".onion's" /24 measurements. The .onion leakage is like many other NXD TLDs at the root that have been shown to exhibit a regular weekly query volume pattern [67], indicating that it is more likely to be the result of an actual use, and not a result of automated queries that lack such pattern.

Similar to our findings with the raw queries, we also can notice the growth in the number of subnetworks from which the queries are originated, although at lower rate than with raw queries. For example, the 70,000 queries observed at the start of our data collection were originated from about 6000 /24 subnetworks. As the number of the queries tripled, the number of /24 networks only increased by 50%, as shown in Figure 1(b). However, this number of subnetworks is more than doubled at peak times (seen in the spikes in the same figure). The findings are interesting, and suggest the widespread of networks from which .onion domains are requested. Whether that is the case or not with coarser network granularity is what we examine next.

AS-Level Measurements: Spread and Growth: Autonomous systems are the coarsest granularity of networks managed by the same authority, and their number is a measure of the spread of hosts over networks. To this end, Figure 1(c) shows the number of ASes from which the various queries are originated. Interestingly, while the number of ASes in general is small compared to the total number of used ASes (i.e., 5% to 10% of total ASes over the entire measurement period), the ASes observed in our measurements include some of the largest on the Internet (more details are in §III-C). Furthermore, the spread of queries over such large number of ASes suggest that the leakage is not an isolated issue, and is rather a global leakage phenomenon. Finally, similar to the previous findings and consistent with the /24 subnetwork measurements, we observe about 50% of a sustained growth of the number of ASes over the measurement period, which reaches a growth of 100% at peak times (during abrupt query spikes).

Representation: The data presented in Figure 1 only represents measurements taken from the A and J root nodes. In order to gauge the total global DNS leakage of .onion requests, we can segregate the unique SLDs received at each root node and compare their overlap. This measure will provide us with the SLD root affinity and is a simple way of estimating total global DNS leakage if this trend was to be extrapolated over all roots.

Figure 2 depicts the number of unique SLDs observed at the A node, J node, and the combination of A and J nodes. In this figure, we can see that the combined A+J roots, on a daily basis, observe about 3300 unique SLDs, while each of the A and J nodes separately observe roughly 2500 unique SLDs—roughly 75% of the combined A+J root nodes. If we are to assume that the resolvers selection of root servers is random (which is the case), then we can estimate the average number of unique SLDs per day at all root name servers using Chapman estimator [22] at $\frac{(2500+1)^2}{1701} \approx 3677$ (with a variance of 812). We notice that the number of actual unique SLDs (from our measurements of DITL data set in §III-E) is about 4100 SLDs for the same year of 2013.

Prior work studying multi-root distinct SLD overlap [67] has shown that the combined traffic observed at A+J constitutes approximately 40% of all observed distinct SLDs for various TLDs spanning the global DNS roots. While the .onion SLD root affinities and overlap between the A and J roots are comparable to the finding in the prior literature concerning other TLDs [67], the actual share of A and J for .onion, is unclear. Therefore, we postulate that the .onion traffic observed at A+J would continue such a trend and an appropriate sizing of total global .onion leakage could be roughly estimated based on those similarities. Based on the statistics in §II, we estimate

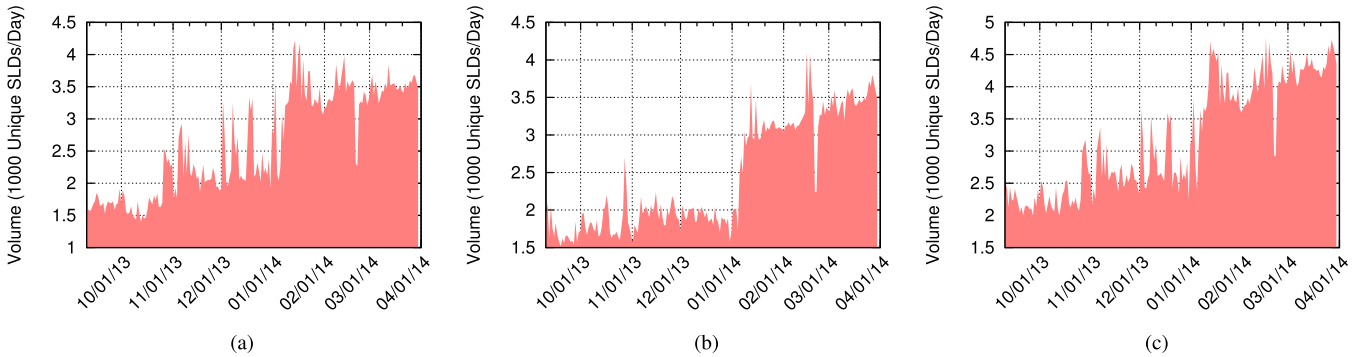

Fig. 2. The .onion traffic measurements observed at the root DNS nodes A and J used for the estimation of DNS root queries. (a) A. (b) J. (c) A+J.

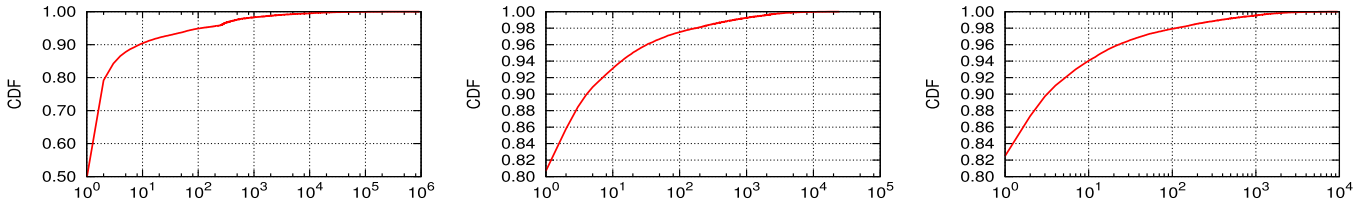

Fig. 3. The .onion traffic measurements observed at the root DNS nodes A and J and their diversity using number of queries from a single IP address (in (a)), from a /24 network (in (b)) and an autonomous system (in (c)) represented as a CDF.

the total number of .onion NXD records at 69 million over the same period of about 6 months. Notice that this calculation should not be used as a generalization, since it does not take into account any unexpected growth. However, this figure should be taken as a rough estimate of the total number of leaked queries.

B. Hidden Service and SLD Measurements

Figure 2 shows a few days in which the absolute number of distinct SLDs drastically increases from the average number of daily SLDs observed in the rest of the measurement period, which calls for further investigation (more details are shown in §III-E).

To this end, we now turn our attention to the overall distribution of requests for a given SLD within the .onion TLD to better understand the DNS request dynamics of all .onion SLDs. Figure 1 provides three different plots of various traffic diversity measurements, namely the number of total requests, the count of distinct /24 subnetwork addresses, and the total number of ASes from which queries for SLDs are received during our observation period. We note that those results are coarse grained, in the sense that they do not consider the contribution of the individual sources, /24 subnetworks, or the ASes of the total number of requests. To this end, we extended those measures to obtain the corresponding and complementary cumulative distributions in Figure 3. The CDF plots capture the number (as a fraction) of SLDs that receive requests by the given number of individual IP addresses of the x-axis (and the /24 subnetwork or AS, respectively)

Raw Sources: Clearly, and based on the results shown in Fig. 3(a), the vast majority of SLDs receive a minimal amount of DNS requests over the six months period covered in our data set. In particular, 50% of the SLDs receive only one request and nearly 90% of SLDs receive less than 10 requests. However, more interestingly, about 1% of the total number of unique SLDs receive more than 10,000 requests, whereas a small number of the .onion SLDs receive more than 100,000 requests. We explore those popular services in details in the subsequent subsections.

Subnetworks: A similar trend of traffic source diversity for the majority of SLDs is displayed at the /24 subnetworks as shown in Figure 3(b), although with a narrower distribution. We see that nearly 80% of SLDs receive requests from only one /24 subnetwork, more than 93% of the SLDs receive requests from less than 10 /24 subnetworks, and over 99% of the SLDs receive requests from less than 100 /24 subnetworks. However, a few SLDs are widespread over a large number of /24 subnetworks.

AS-Level Measurement: The distribution gets narrower at the AS-level, as shown in Figure 3(c), indicating less overall diversity in the networks from which requests are issued for the various SLDs observed in our data. About 94% of the SLDs originate from fewer than 10 distinct ASes, leaving very few SLDs with large amounts of traffic from a wide variety of network locations. This pattern is in line with the general traffic characteristics and trend for other non-delegated TLDs. However, more interestingly, we notice that the head of the AS and SLD request distribution is occupied by large ASes that host large numbers of users, or those that host open resolvers that are likely to be used by large number of users. In the subsequent sections we elaborate on those ASes.

SLD Lifespan: Further Temporal Characteristics: Next, we focus on the SLD lifespan indicated by how long it is queried. The time difference between the first and the last query of a given unique .onion string (in seconds) indicates the lifespan of a .onion leaked strings at the root during our observation time and is shown in Figure 4. Notice the 80%-∼3%-∼17% distribution of short-lived (less than 10 seconds), mid-lived (less than a day), and long-lived (more than a day) SLDs. Furthermore, we notice that the lifespan

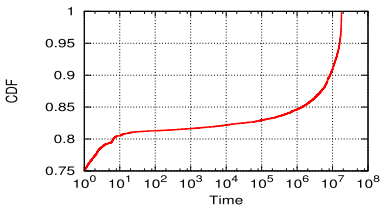

Fig. 4. The .onion SLD lifespan: notice the skewed distribution.

of 75% of the queried .onion strings is less than 1 second, indicating that they are perhaps the result of automated scan or user error that is unlikely to recur–which is partially verified by finding that many of those SLDs are sequential. On the other hand, 15% had more than 15.85 days, 10% had more than 98.11 days, 5% had more than 183.29 days, and 1% had more than 206.45 days, suggesting a persistent use scenario, as opposed to the previous scenario of short-lived SLDs. By correlating the number of queries per unique SLD and its lifespan, we obtain a positive and small correlation of 0.09, indicating that the popularity of a given is less likely affected by the lifespan of a domain name. This is particularly naturally understandable in light of the various plausible causes of the .onion leakage in the first place. More details on such correlation measurement are in the appendix.

Popular SLDs: Next, we shift our focus to those few but very popular SLDs within the .onion TLD. Table 3 provides a list of the most requested hidden services along with their total percentage of .onion traffic and the type of service provided using them. The mapping of SLDs to their type of service was constructed manually by searching for references of the hidden service online. The SLDs listed in the table have been anonymized (masked) for privacy concerns, where the first and last two characters of each SLD are shown. Notice that this is a best-effort attempt to hide the addresses, although anonymizing those services is outside of the scope of this study; a simple search on the most popular hidden services in the given category can easily reveal them

From the statistics shown in Table 3, we observe that nearly 27% of all .onion traffic belong to one hidden service whose focus is on Torrent tracking. The remaining traffic forms a long tailed distribution over the remaining hidden services with an emphasis on services surrounding search, commerce and currency exchange. The top 10 hidden services shown in Table 3 account for more than 38% of the traffic (i.e., total number of requests) observed over the total period of time of our data set at the roots A and J.

C. Traffic Source Measurements

The source IP addresses requesting the various .onion SLDs can be used to obtain various traffic source metadata, which are worth investigating to highlight the geographical and network diversity of the requested SLDs. To this end, we explore such metadata in details. In all of those analyses, we use an off-the-shelf commercial-grade geomapping service for IP addresses to the country of origin and ASN [25].

In Table 4, we examine the origination of the .onion DNS requests issued by recursive name servers to the A and J roots from a country perspective. To ensure that publishing those

TABLE III
MOST POPULAR SLD HIDDEN SERVICES AND THEIR TRAFFIC MEASUREMENTS

Rank	Masked SLD	Type of Service	Traffic (%)
1	Z6---------43	Hidden Tracker	26.5
2	DK---------II	Silk Road	2.1
3	DP---------PC	TorDir	1.7
4	SI---------FK	Silk Road	1.4
5	3G---------4M	Search Engine	1.3
6	JH---------JX	Tor Mail	1.2
7	XM---------SL	Search Engine	1.1
8	AG---------WW	Agora Marketplace	1.1
9	FO---------UI	Bitcoin	0.9
10	TO---------NS	TorLinks	0.9

TABLE IV
TOP GEOGRAPHICAL COUNTRIES AND ASNS REQUESTING "ONION" WITH COUNTRY CODE (CC), REQUESTS, AND TRAFFIC (%)

cc	Requests	%	cc	Requests	%
US	9,878,093	35.7	FR	670,103	2.4
RU	2,213,691	8.0	AU	510,745	1.8
DE	1,482,075	5.3	NL	454,441	1.6
BR	1,258,468	4.5	ES	448,171	1.6
CN	996,130	3.6	IE	425,469	1.5
GB	984,059	3.5	IT	423,550	1.5
KR	980,656	3.5	AR	387,594	1.4
PL	918,948	3.3	MX	363,389	1.3
CA	785,184	2.8	IN	295,122	1.0

TABLE V
TOP GEOGRAPHICAL COUNTRIES AND ASNS REQUESTING "ONION" WITH ASN, REQUESTS, AND TRAFFIC (%)

ASN	Requests	%	ASN	Requests	%
AS15169	2,267,250	8.2	AS6830	392,233	1.4
AS7922	1,222,955	4.4	AS20115	342,716	1.2
AS7018	654,680	2.3	AS3786	326,885	1.1
AS36692	571,609	2.0	AS28573	309,751	1.1
AS30607	561,349	2.0	AS5617	290,577	1.0
AS4766	560,739	2.0	AS3356	290,160	1.0
AS701	512,989	1.8	AS7738	284,726	1.0
AS7132	447,528	1.6	AS22773	273,845	0.9
AS22773	400,657	1.4	AS4134	258,832	0.9

statistics does not put the privacy of individual users at risk, we verify that IP allocations for all countries listed herein are large enough.

The geographical distribution of .onion requestors deviates from the top-10 countries by directly connecting users as reported by the Tor project over the same period of time [65]. At nearly 36%, the US is 3 times higher than reported from Tor. Other countries such as Germany, France, and Spain also differed significantly, with 7.7%, 7.23% 6.17% and 4.8% respectively [63]. While it is clear that the leaked .onion queries to the global DNS roots and actual Tor connections are very different (e.g. measuring recursive name servers vs. direct connections), the variance in the distribution of the .onion requests may prove helpful in understanding the root cause of the leaked DNS queries and perhaps highlighting measures implemented by certain countries to address .onion leakage (at a state level).

AS-Level Characterization: Next, we explore the head of the distribution for ASes that generate the most amount of .onion traffic. With such a large percentage of .onion requests

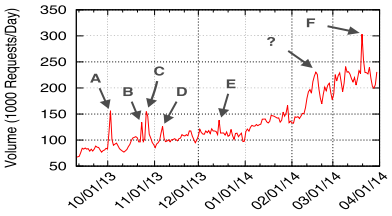

Fig. 5. Global Events and Elevated Request Correlation.

TABLE VI
GLOBAL EVENTS AND ELEVATED "ONION" REQUEST CORRELATION

Event	Date	Req.	Event
A	10/03/13	156,312	Silk Road Shutdown [2]
B	10/24/13	134,236	TorATM Traffic Spike [18]
C	10/27/13	154,855	URL Posted on Reddit [1]
D	11/07/13	126,398	New Silk Road URL [12]
E	12/15/13	138,231	Pirate Bay URL Posted [75]
F	03/21/14	303,347	URLs Posted on Reddit [5]

originating in the United States, it is not surprising to observe the major Internet Service Providers (ISPs) in Table 5 (AS7922 is Comcast and AS7018 is AT&T)— we note that all of the autonomous systems listed in Table 5 have large number of IP addresses allocated to them, thus publishing them does not put the privacy of individual users at risk. However, it is interesting to observe that nearly 8% of all .onion traffic originates from AS15169 (Google). We hypothesize that users and advocates of Tor would most likely not use their default ISP name servers and instead would choose to use public DNS providers such as Google Public DNS or OpenDNS (AS36692, which has a share of 2.06%). However, more surprising and related to this observation is to see that many .onion queries originated from AS15169 given that Google Public DNS has an intensive caching policy in use to avoid multiple queries to the root that would potentially result in NXD [39].

Given the special nature of the .onion TLD and its queries, and that they are not supposed to be exposed to the DNS infrastructure, a role that such providers may play in addressing the problem can include blocking such requests at the recursive level, which we suggest in §V-B. However, we notice also that such mitigation would not prevent such recursive servers (or proxies between them and users) from profiling hosts and their use.

D. Global Event Correlation

Global events, such as Internet censorship, political reform, and economic shifts, among others, spur the use of privacy enhancing technologies like Tor [44]. The total traffic volume measured on a daily basis in Figure 1 exhibits several spikes in which .onion traffic significantly increases from its moving average. In order to better understand these events, we cross-correlated the spikes with news stories on global events. For that, we used google search for searching for the relevant news stories to extract the events. For example, we start with the .onion services that are leaked as highlighted in Table 6, which are (in all cases) strings searchable in news websites. We did not provide such names in full in the paper for their privacy value. Table 6 lists the events and their impact on .onion traffic. These events typically manifest themselves in the form of

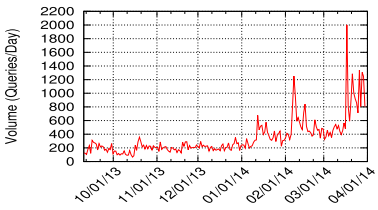

Fig. 6. The .onion traffic measurement and leakage from Ukraine.

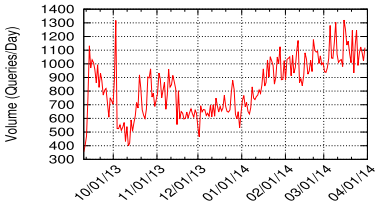

Fig. 7. The .onion traffic: Ukraine.

increased traffic from a specific geographical region or the predominance of queries for a particular SLD. Figure 5 (an annotated version of Fig. 1(a)) plots the events listed in Table 6 against the total daily .onion traffic volume, highlighting the spikes in relation with the rest of the volume over the entire period of our observed data. In the following we highlight in more details specific and noteworthy examples by surgically separating country-level traffic to observe local events associated with them via the volume of leaked .onion traffic.

Turkey: Certain global events such as the censorship of Internet domains in Turkey may span a longer period of time than a few days, which we attempt to understand and establish via the leaked .onion traffic. Figure 6 depicts the number of requests for .onion domains originating from Turkey over the multi-month collection period. There is a clear upward trend and a sudden increase in the second half of March 2014 when many DNS-based censorship events took place. The requests originating from Turkey during the censorship spanned hundreds of unique SLDs and were spread over several ASNs. However, also interesting is the number of spikes in .onion requests observed, which could potentially be attributed to various local events within the country.

Ukraine: November 2013 witnessed the Euromaidan (European square) demonstrations that led to the 2014 Ukrainian revolution. In Figure 7, we capture requests originated from Ukraine over time, and notice a substantial growth in the number of queries post September 15. For example, starting with only 400 requests per day, the number suddenly increase by more than 200%, which is sustained over time after mid January 2014. While there could be multiple explanations for the increase in the number of .onion requests, the type of hidden services queried and leaked at the root being topic-specific to the revolution highlight the great correlation between the increase in the volume and the political event.

E. Trends From the DITL Data Set

Now we turn our attention to the DITL data set described in §II-B and try to identify the prevalence of .onion leakage from all root servers. In analyzing the DITL data set we benefit from two aspects that are lacking in the A and J data set that we have examined so far: representation and longitude.

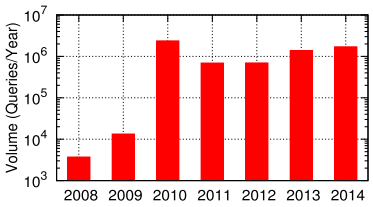

Fig. 8. Queries over time.

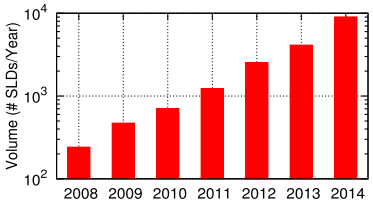

Fig. 9. SLDs with .onion queries over time.

The DITL data set collects queries from a large number of DNS servers, and gives an insight into the DNS resolution for multiple years. In doing so and by analyzing this data set, we concentrate on three aspects. 1) **The existence of .onion queries longitudinally:** given that the DITL data set covers a relatively longer period of time than the data set we used from the A and J root servers, we aim to examine whether the .onion leakage is a temporary event or lasting phenomena over that long period of time. 2) **Growth trends of .onion leakage:** we aim to examine whether there is a growth trend in the number of .onion requests, SLDs being requested, and IP addresses requesting those TLDs, and whether such trend is consistent over time. 3) **Representation:** how representative are the A and J root servers to the total queries at the DNS roots.

1) The Prevalence of .Onion Leakage: Table 1 summarizes the DITL data set, including the total number of queries observed in each year of the data set's life. We notice that while the phenomenon starts as a small set of queries in 2008, the total number of queries grows 3 orders of magnitude by the year of 2014, and persists over the years between them.

2) Growth Trends:

Number of queries: The results in Table 1, which are plotted in Figure 8, show a growth trend for the number of .onion requests observed at the root servers over time. This monotonic growth trend is interrupted by a sharp growth in 2010, where the number of queries increased two orders of magnitude more than in the previous year (2009), and dropped by one order of magnitude for the year of 2011. We notice that the sharp increase that interrupted the monotonicity in the growth of the number of queries over years might not be a determining trend. In particular, given the nature of the data set, a small event may actually cause a sudden surge in the number of queries, as shown in §III-D, where such surge does not persist as a trend. Indeed, we notice that this interruption of monotonicity is due to a single SLD (`z6---------43.onion`) for a tracker that attracted a large number of queries.

Number of SLDs: The total number of SLDs that attracted .onion traffic and seen at the root for the observation period grows exponentially, as shown in Figure 9. This trend can be used to precisely extrapolate the number of SLDs to be observed at the root unless the root cause of leakage is addressed. Note that, and unlike the interruption in the monotonic growth trend with respect to the total number of queries discussed earlier, no such interruption is introduced at the SLD level, given that the majority of added queries in the surge are due to a single SLD.

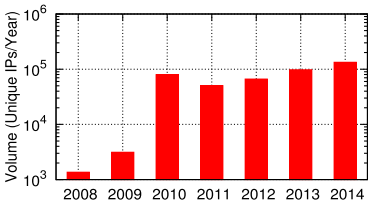

Fig. 10. Individual IP addresses originating the .onion traffic. Notice that the same growth trend shown in the number of queries is also reflected on the number of addresses. (Raw addresses).

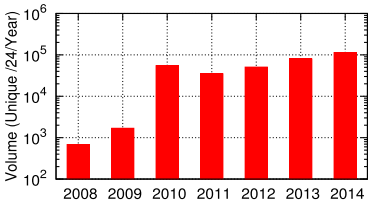

Fig. 11. Individual /24 addresses originating the .onion traffic. Notice that the same growth trend shown in the number of queries is also reflected on the number of addresses. (/24 subnetworks).

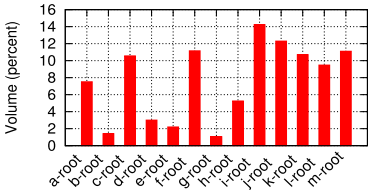

Fig. 12. The percentage of queries observed at each root, with A and J having 7.5% and 12.3%, respectively.

IP addresses: Figure 10 and Figure 11 show the volume of the total number of IP addresses and their aggregated counterpart over /24 subnetwork addresses over time. As with the general growth trend with the number of queries, an interruption in the monotonic growth happens in 2010. However, we observe a consistent and persistent trend of growth for the number of IP addresses originating the .onion queries, as well as their diversity of location measured by their /24 association, which is also consistent with the previous findings over our A and J dataset.

3) Representation: An interesting question that is raised by our reliance on the A and J roots operated by Verisign is "how representative are both root nodes for the population of queries seen at all roots?". Understanding this representation would explain the size of the problem reported in this paper in the Tor system as a whole. Unfortunately, the DITL data set does not have traffic from all root servers except for the year of 2010, to which we limit our attention to answer the aforementioned question, despite some caveats.

Figure 12 shows the share of queries observed at every root for the year 2010 as a percent of the total number of queries (obtained from the statistics in Table 2). On this figure, we

TABLE VII
TOP GEOGRAPHICAL COUNTRIES AND REQUESTING "ONION". THE RESULTS USE THE COUNTRY RANKING IN TABLE 4

Country	Requests	(%)	2014	2013	2012
US	1,469,134	21.45	30.96	24.85	17.98
RU	302,222	4.41	5.34	5.74	8.02
DE	228,487	3.34	5.06	3.46	3.79
BR	256,195	3.74	3.93	8.03	3.44
CN	201,318	2.94	2.05	2.27	2.32
GB	320,550	4.68	4.14	3.44	3.52
KR	63,546	0.93	1.24	1.05	1.33
PL	136,693	2	1.34	2.03	1.72
CA	367,859	5.37	1.75	2.84	2.01
FR	245,159	3.58	1.83	1.59	2.42

make two observations. First, the distribution of requests over all root servers is not uniform, with a few servers answering the majority of queries (roots F, I, J, and K answer a combined total of 56% of the queries). Second, and in answering the aforementioned question concerning representation of A and J, we find that they answer 7.5% and 12.3%, respectively, with a combined total number of queries of 19.8%. If such ratios hold over time (an assumption that we were not able to verify for the lack of data, and is unlikely to hold given what we observed of the variability of queries over time), they put the earlier estimates for the total number of queries at the DNS root servers—for the same period of time corresponding to the timeframe where the data described in §II-A was collected—to 139.4 million queries (from the previous estimate of 69 million queries). This estimate gives an average query rate of about 840k queries per day. As a result, we conclude that the total number of .onion queries is substantial, and may potentially pose a high risk to a large number of users.

4) Geographical Distribution: Similar to the experiment shown in Table 4, we map the various sources originating traffic to their home countries. With respect to the index of countries shown in Table 4, we choose the top 10 countries, and compute the share of requests originated from them. Table 7 shows the number of queries originated from each given country in the top, along with their percent out of the total queries computed over the 7 years of DITL. We further add a per-country percentage of traffic share for the years 2014 and back until 2012. From this table, we make the following observations:

- Inconsistent representation: while part of the order of the countries is mostly consistent with the ranking provided by the Tor project on its use [66], we find that the ranking is inconsistent with the previous ranking established for the countries with traffic seen in the A and J root study.
- Inconsistent order: the order of countries as shown in Table 4, which highlights countries in a descending order, is not preserved in Table 7. This highlights dynamics of shares in the traffic, perhaps based on phenomena best seen in those countries through the usage of hidden services.

IV. ROOT CAUSES

Applications electing to use non-delegated TLDs as a namespace in which they seed their routing and resolution processes face scenarios in which possible DNS leakage may occur. While the security and privacy of users in some application that utilize such technique might not be affected, it is at stake with other applications. For example, Tor has been specifically designed to prevent .onion requests from leaking within the application into the global DNS infrastructure. However, it is clear from the measurements we presented so far that a significant volume of requests are being issued to the global DNS root servers. Whether they are initiated by users by mistake, caused by a misconfiguration in the underlying application such as Tor client, or resulted from prefetching web browsers, leaked DNS queries outside of the Tor network have a significant implication to individuals' privacy, and perhaps more importantly to their safety. To this end, understanding the causes of the leakage is of paramount importance and may help reducing the risk of leakage at the user side.

There are many plausible reasons or mechanisms in which .onion queries could be generated and observed in the global public DNS; however, the root cause of how and why these queries are being requested within the global DNS remains unclear and is indeed very difficult to pinpoint given the sophisticated and increasingly interdependent system that DNS is today. In the following, we outline some of those plausible root causes, including user error, browser prefetching, third party application or plug-ins, DNS suffix search lists, web crawlers, and malware. We also provide various case studies that highlight the potential of those root causes as a possible explanation for the observed .onion traffic in the public DNS infrastructure.

A. User Error and Misconceptions

We have seen so far numerous global events that spurred additional query volume. One potential explanation associated with the surge in the volume of .onion domains in those times is users errors, in which users are not aware that the addresses of hidden services should be run on top of Tor (i.e., by first installing the Tor plug-in associated with the browser).

Validation: User Study: To validate whether the user error and misconception are a root cause of the leaked .onion domain names, we perform a user study with the proper institutional review board approval, and highlighting those findings are in line with the best practice recommendations provided in [11].

User Study: Settings: To understand whether this hypothesis for user error being a possible root cause for the leakage of .onion or not, we conduct the following study. Along with other domain names in other TLDs, we present .onion domain names to a set of users, and aim to answer the following questions: 1) how many of the users recognize the special nature of the pseudo-TLD and the domain names associated with it, and 2) any special considerations that the users would need to take into account when querying .onion pseudo domain names. For this study, we consider 27 subjects recruited from a graduate-level advanced computer security class in our college. The average of the subjects was 24.3 years, with the minimum of 22 years and a maximum of 34 years of age. 2 of the subjects were females, while 25 were males, and all of them identified a certain level of DNS and privacy enhancing technologies and

their operation: 16 students identified as very knowledgeable (viz. Tor's hidden services), 5 as knowledgeable, and the rest of the users identified themselves as familiar.

Results and Findings: Among the 27 subjects we tested in this user study, we found that only 8 subjects recognized the special purpose of the .onion pseudo-TLD. Of those 8 subjects, only 3 recognized how to use .onion pseudo domain names, having already used them in the past, while others were not able to provide a correct use idea. On the other hand, all of the subjects indicated their knowledge of Tor as an anonymizer.

B. Browser Prefetching

Web prefetching is widely used nowadays and is aimed at improving user experience [41]. In a typical web prefetching mechanism [36], web browsers proactively try to retrieve contents of links on a page so that a user who is likely to visit is served the contents from the web cache. In particular, such mechanisms are particularly useful and effective when the browsing behavior of users is predictable [27].

Similar to web prefetching is DNS prefetching [38], in which the browser proactively tries to resolve links (at the DNS level) posted on a page before they are visited and while the user is idle. In doing so, the links are requested directly when the user needs to visit them thus saving the DNS resolution round trip time. As of late 2013, both mobile and desktop web browsers support prefetching in both forms. For example, the Google Chrome, Internet Explorer, and Mozilla Firefox, which account collectively for a great majority of the usage share of web browsers, support the standard web and DNS prefetching [60].

With the lack of explicit rules in a suffix list that prevents browsers from prefetching names in the .onion pseudo TLD, DNS prefetching stands as one of the very important and potential root causes of the .onion leakage at the root DNS servers.

Validation: For validation, we use a crawl of the .com and .net domain names (more than 130 million domains) provided by Verisign, while limiting the crawl depth to 1 (i.e., the front page and pages pointing to it). For each website we crawl, we statically analyze the contents of the pages by searching for .onion suffixed strings in it. Among those domain names, we identified 33,257 domain names that actually have at least one .onion link in them. Given that the number of domain names is minuscule compared to the total number of possible domain names in both TLDs (i.e., only $\approx 0.026\%$), we explore further evidence for the potential of prefetching as a root cause by searching our root dataset for .onion strings observed in our crawl. We found that a large number of those domain names match: over 17% of the .onion strings in our A+J root dataset were also observed in our crawl, and their share of the queries were over 92%. This further highlights and supports the potential for prefetching as a root cause from data. We notice that many of those incidents of .onion domains on web pages in the .com and .net zone are hosting blogs, forums, and news outlets in which .onion strings are distributed, advertised, or just mentioned.

C. Malware

One of the important building blocks of today's malware families is their reliance on advanced mechanisms for communication between botmasters and infected hosts [54]. One of such advanced techniques utilizes DNS, by using domain names registered by the botmaster as a communication channel (those domain names are often algorithmically generated so that bots can generate and use them as well for communication) [8]–[10], [46], [51], [68]. However, there has been an ample body of work on detecting such domain names registered under delegated TLDs [10], [72], [73], thus thwarting their harm and limiting their use as a C2 channel.

Validation: To this end, and to validate that malware is a root cause for the spikes in the leaked .onion, we analyze the data at hand. We observed numerous requests for .onion SLDs associated with the aforementioned malware families during our analysis (names of .onion SLDs are obtained from malware analysis and intelligence reports). While the root and original cause for observing such .onion strings is unclear, whether it is the result of a curious user attempting to resolve a .onion name used by malware, a browser prefetching a .onion pseudo-domain name on a webpage, or a malfunctioning piece of malware trying to connect to the C2 server using the address, the circumstantial evidence suggests that it is potentially a combination of a wide array of causes. For example, spikes of the malware .onion strings coincided with a large campaign launched by various of those malware families that was reported by various forums and media outlets. However, for the same reason of them being posted in those forums, such leaks might as well be the result of browser prefetching and not the actual malware activity.

V. Implications and Remedies

Given the widespread of .onion leakage at the root and other levels of the DNS resolution hierarchy, a next natural step would be to understand implications of such leakage on privacy and their remedies. In this section we discuss such implications (§V-A) and remedies (§V-B), focusing on recent developments in line of those remedies (§V-E), attempts to further manage the DNS namespace (§V-D), and more recent developments (§V-E).

A. Implications

1. Individual users' IPs and their resolution preserve locality information of the users issuing such information, and may considerably expose users to a high risk, depending on their location and the context of the queried hidden service.

2. Many of the queries issued to the root come from public recursive DNS servers that are responsible for a large number of queries aggregated from potentially multiple users, where the individual users' IP addresses are detached, thus the root does not see those address. However, this still puts the individual users at risk, although their individual IP addresses are not exposed. For example, the DNS queries observed at the root are likely the result of unencrypted traffic that an eavesdropper close by the user can listen to, and associate to the user. Furthermore, most public recursive service do not preclude the possibility of sharing users' traffic with a third party in their use agreements.

3. Whereas ISPs might be disincentivized from sharing the individual users information with third parties, eavesdropping while closer to the users may expose them. Furthermore, when

ISPs are a government entity (e.g., in Egypt and Turkey), their double function puts users at risk.

4. Unlike ISPs, open resolvers that do not serve a clear business agenda, do not have the business relationship with users, and might be willing to share such information with a third party, thus putting users at a great risk.

B. Remedies

1) Host-Level Remedies:

Browser: Given the nature of .onion, and other privacy or special purpose TLDs, special treatment—including blocking capabilities—should be enabled at the edge, including capabilities of blocking in the browser – when Tor is not being used in the first place. Furthermore, users are often time not exposed to low level details of connection failures with today's Tor distribution, and blocking may help mitigating the leakage of .onion queries when a Tor connection fails for one reason or another. Furthermore, as we have seen in this study, unaware users make it possible to observe some of the .onion traffic in the public DNS infrastructure, partly because they are not knowledgeable of the special nature of those domain names. Or even worse, some of the leaked .onion resolution in the public DNS is due to certain functionalities implemented in the browser to improve user experience, such as DNS prefetching. To this end, when enabled, blocking of .onion queries based on further intelligence in the browsers, whether it is by excluding .onion domains in a suffix list from any further active prefetching, or determine whether to allow resolution of .onion when provided by users only when Tor is used, could perhaps help remedies the leaked .onion queries.

Legacy Software: Even when measures are taken to reduce the amount of leaked information in the public DNS infrastructure by, for example, implementing a suffix list and disabling prefetched queries of .onion when Tor is not being used on the hosts, legacy software will still leak information for the same reason. It is widely noticed that legacy software consistute a large number of the software on the end hosts. To this end, measures to address legacy software and their contribution should be considered and implemented. Queries associated with privacy enhancing technologies such as Tor should be controlled as to prevent and notify users if public DNS leakage occurs due to those legacy programs. Such measures could be by implementing a host-level profiler of all DNS traffic generated by hosts.

Configurations: With misconfiguration that may result in exposing users behavior in private networks (such as Tor) to the public DNS infrastructure, measures should be implemented to automate configurations using best practices. Automatic system-level configuration of .onion resolution should be used. The Tor distribution should provide a system-level fix to local DNS configuration and not require users to configure this component manually, or even allow them to do so.

2) Network-Level Remedies: Along with the host-level remedies discussed earlier, there are also some remedies that could be implemented in the network, by equipping DNS resolvers and authoritative name servers to handle .onion strings differently, thus blocking the leakage.

DNS resolvers: Many of the queries can be blocked lower in the DNS hierarchy, and be prevented from propagation into the public DNS by deploying techniques such as negative caching [7]. For example, public recursive name servers most close to the users may help by not sending out queries to the root for TLDs that do not exist. Given the (almost) static nature of the TLDs, and the static nature of the TLDs of interest (such as .onion), operators of public DNS services may deploy effective mechanisms in achieving such goal.

Authoritative name servers: We notice that not much that is not done today can be additionally performed at the authoritative name servers to address the leakage problem of .onion strings. This is particular true given that the leakage is already observed in the networks connecting the stub resolver (user) with the authoritative. However, to reduce the attack surface associated with .onion leakage, authoritative servers should not attempt to resolve .onion strings, and should always return negative resolution results. Once measures are performed at the recursive side, less queries of .onion will be exposed to the authoritative servers, which could remedy this global leakage. Notice that such remedies might be operationally subtle, especially in light of the various implications discussed in §V-A. Nonetheless, we include them here for the complete treatment of the subject.

C. Comparison of Remedies

In the following we compare the different remedies presented in this section for their potential in addressing the .onion leakage. We compare those remedies based on their anticipated effectiveness in blocking leakage, the privacy they ensure by such effectiveness, and the amount of effort and level of difficulty required for implementing them in the existing domain name infrastructure.

Privacy: We notice that a system that implements remedies at the host-level would ensure the highest level of privacy among all suggested solutions, since no queries would be leaked to the public DNS infrastructure upon successful implementation and enforcement of remedies at the client side. However, we also notice that such remedy is complex by nature, since it requires fix to the problem in multiple types of operating systems, browsers, etc., including addressing issues with legacy software. Second comes remedies at the recursive side, which would expose .onion queries to individual recursive, which would identify the individual use of clients of .onion, but would prevent the rest of the DNS entities and infrastructure from knowing what is being queried if remedies are implemented at the recursive. We believe that remedies at higher levels of the infrastructure, e.g., root, do not facilitate privacy, although they are easy to implement.

Complexity: As noted earlier, client-side remedies, including browser-level blocking, can be very complex, since they require addressing the problem in multiple instances of operating systems and browsers, and for large numbers of users (all potential users on the Internet). The number of resolvers is relatively smaller than the number of users, making the problem less complex with resolver-level remedies. However, the problem is still nontrivial given the diversity of the software-base of DNS servers and their versions (including legacy

TABLE VIII
COMPARISON OF THE VARIOUS LEVELS OF REMEDIES

Remedy	Privacy	Complexity
Host-level	More privacy	More complexity
Recursive-level	Less privacy	Less complexity
Authoritative-level	No privacy	Least complexity

software) that would require modifications and update. Finally, the authoritative-level remedies provide the least complex remedy. However, they do not ensure any levels of privacy, with all links between the stub and authoritative being exposed (unless negative caching is aggressively employed).

In conclusion, Table 8 provides a comparison between the various levels of remedies summing up this evaluation. We note that an effective remedy might use multiple of those remedies.

D. Namespace Management

Focus within the Internet Engineering community has recently increased on ways for applications to properly use non-delegated domains. A recent Internet draft describes several special-use domain names of peer-to-peer name systems and is seeking approval from the Internet Engineering Steering Group (IESG) [30]. Discussions about the proposal on the DNS operators mailing list have brought forth other generic solutions such as proposed .alt alternative TLD in which applications would safe anchor namespace under it [70]. Blurred lines of authority, privacy and security makes such a namespace problem difficult to solve and appease all parties.

E. Recent Developments

After the release of our preliminary results in [69], Appelbaum and Muffett [11] led drafting an RFC to address the special nature of the .onion pseudo-TLD and associated strings, and using some of the recommendations in our study. In their view, they propose that .onion should be registered as a special case TLD, users should be made aware of such special nature of the TLD and strings associated with it, applications must recognize the special use of .onion strings, name resolution APIs must respond to .onion strings and their queries by resolving them according to Tor specifications, resolvers that are not part of Tor and its operation should not attempt to resolve .onion strings, while authoritative name servers should respond with NXDOMAIN response (which is the case today). While most of their recommendations are identical to our study in [69], which precedes their work, their novel recommendations, in general, imply collaboration of various entities in the DNS ecosystem with the Tor system for safe resolution, which requires major changes in the existing infrastructure.

Taking our recommendations into account, the most recent release of unbound (developed by NLnet Labs and sponsored by Verisign) in February 2016 addresses leakage of .onion by developing a fix to the problem and blocking .onion queries in the DNS resolution hierarchy at the recursive resolver level. However, we emphasize that unbound is only one among many distributions of DNS servers that also need to address this critical issue. Furthermore, and based on our previous analysis, blocking .onion queries at the root only prevent the root from observing .onion queries, but does not prevent third party resolvers from observing the leakage of .onion and profiling users. To this end, fixes that include addressing root causes (e.g., browser, legacy software, etc) at the host-level perhaps should be considered. Finally, our previous work [47] analyzed the privacy implications of blocking of the leaked DNS queries as a method of improving the privacy of users.

VI. RELATED WORK

With the exception of our preliminary study in [69], there has been no prior work on measuring and understanding the leakage of .onion in the DNS infrastructure in a systematic way. The exceptions for such systematic study which is lacking from the literature include anecdotes reported in news stories, as seen for example in [28]. However, broadly related to our study are various lines of research that highlight the use of hidden services, their deanonymization, Tor use characterization, and remedies to DNS leakage. In the following we review a sample of those works.

Hidden Services

There has been several works in the literature on measuring, understanding, and attacking the Tor's hidden services. Kown *et al.* [40] proposed a passive attack on hidden services that utilizes circuit fingerprinting. Owen [49] proposed to denonymize hidden services using global attack capabilities. Hopper looks at the challenges of protecting hidden services from aggressive usage by malware [31], Biryukov *et al.* [16] analyze contents popularity of hidden services based on their prior work of detection and deanonymization of hidden services in [15].

DNS Leakage

Recommendations for addressing DNS leakage of .onion strings have been made by Appelbaum and Muffett [11]. DNS leakage as a side channel to undermine security of cloud services is explored by Ristenpart *et al.* [52]. Rose and Nakassis [53] proposed mechanisms for minimizing information leakage in DNS. Similar ideas of minimization, but applied at the query-name level, are discussed by Bortzmeyer [18]. Bortzmeyer [19] proposed QNAME minimisation to decrease exposure to the authoritative name server. to protect the DNS query and response interaction between a DNS client and a DNS resolver. Thomas suggested blocking lists (in the browser) for addressing DNS leakage [67]. Simpson investigated how search lists affect DNS leakage [58]. Chen *et al.* [23] addressed the Web Proxy Auto-Discovery (WPAD) name collision attack from the unintentional leakage of internal WPAD DNS queries into the public DNS namespace.

DNS Profiling for Attribution

DNS leakage (intentional or unintentional) has been intensively used in the past for profiling end hosts, and sometimes

for detecting malicious activities and actors. Jones *et al.* [35] presented techniques for detecting unauthorized DNS root servers on the Internet using primarily endpoint-based measurements. Jiang *et al.* [33] identified DNS radiation and constructed failure graphs for malware detection. Luo *et al.* [43] utilized a similar concept by leveraging client-side DNS failure for malware detection. Xu *et al.* [71] proposed to use DNS for large-scale command and control. The use of DNS for identifying fast-flux domain names has been explored by Perdisci *et al.* [50]. DNS for botnet takedown has been explored by Nadji *et al.* [48]. Passive DNS analysis for malware detection has been explored by Bilge *et al.* [13], [14], among other works.

VII. Conclusion and Future Work

In this paper we introduced the first in-depth study of the .onion DNS requests at both the A and J root name servers, and from the day in the life of the Internet (DITL) data sets. We identify the prevalence and scale of .onion leakage in the public DNS infrastructure, and examined the unique characteristics of .onion requests longitudinally as well as the dynamics of requests received from a geographical and network location for unique SLDs. We found that increased traffic spikes within the global DNS for .onion requests corresponded with external global events, highlighting the potential human and ecosystem factor in those leakages (i.e., user error and DNS prefetching). While the root cause of these leaked DNS queries remains unknown with high certainty, particularly as to what is the contribution of each cause, our investigation unveiled plausible explanation for some of this leakage supported by various case studies.

Our future work will continue this line of work at multiple fronts. First, we will continue the examination of leaked DNS queries to the root by extending our study to other non-delegated TLDs such as i2p and .exit. Second, we plan to further dissect the impact of global events and the role of malware in the leakage, potentially towards swatting their risk, and investigate the potential privacy consequences of the leakage under the various leakage causes. Third, we will explore the potential of analytically exploring the cost and effectiveness of the various remedies with more concrete deployment scenarios, which have been out of the scope of this study. Finally, we will analytically explore how partial blocking of .onion in the DNS infrastructure affects privacy.

Acknowledgement and Disclaimers

The authors would like to thank M. Thomas for his help, D. McPherson and E. Osterweil for their feedback, and A. R. Kang for proofreading earlier versions.

References

[1] (Oct. 2013). *ELI5: What Exactly is the 'Deep Web'. Reddit*. [Online]. Available: http://bit.ly/1l7hLbz

[2] Huffington Post UK. (Oct. 2013). *FBI Arrest 'Silk Road' Owner Ross William Ulbricht, Shut Down Tor's Most Notorious Black Market*, http://huff.to/1fu0tA7

[3] E. Osterweil, M. Thomas, A. Simpson, and D. McPherson, "New gTLD security, stability, resiliency update: Exploratory consumer impact analysis," Verisign Labs, Reston, Virigina, USA, Tech. Rep. 1130008, Aug. 2013, pp. 1–28. [Online]. Available: http://bit.ly/QB6ntp

[4] ICANN. (2014). *New Generic Top-Level Domains*. [Online]. Available: http://newgtlds.icann.org/en/

[5] (Mar. 2014). *People who Have Visited the 'Deep Web' What was it Like and why did you do it?* [Online]. Available: http://bit.ly/ROuupk

[6] L. Abrams. (Feb. 2014). *CryptorBit and HowDecrypt Information Guide and FAQ. Bleepingcomputer*. [Online]. Available: http://bit.ly/1eoKEjh

[7] M. Andrews, *Negative Caching of DNS Queries (DNS NCACHE)*. document RFC 2308, 1998.

[8] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for DNS," in *Proc. USENIX Secur. Symp.*, 2010, pp. 273–290.

[9] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, II, and D. Dagon, "Detecting malware domains at the upper DNS hierarchy," in *Proc. USENIX Secur. Symp.*, 2011, pp. 1–16.

[10] M. Antonakakis *et al.*, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in *Proc. USENIX Secur. Symp.*, 2012, pp. 491–506.

[11] J. Appelbaum and A. Muffett, *The Onion Special-Use Domain Name*, document RFC 7686, Oct. 2015.

[12] J. Biggs. (Nov. 2013). *Silk Road 2.0 Rises Again. TechCrunch*. [Online]. Available: http://tcrn.ch/QB5HnQ

[13] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding malicious domains using passive DNS analysis," in *Proc. NDSS*, 2011, pp. 1–17.

[14] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "EXPOSURE: A passive DNS analysis service to detect and report malicious domains," *ACM Trans. Inf. Syst. Secur. TISSEC*, vol. 16, no. 4, p. 14, 2014.

[15] A. Biryukov, I. Pustogarov, and R. Weinmann, "Trawling for Tor hidden services: Detection, measurement, deanonymization," in *Proc. IEEE SP*, May 2013, pp. 80–94.

[16] A. Biryukov, I. Pustogarov, and R.-P. Weinmann, "Content and popularity analysis of Tor hidden services," in *Proc. ICDCS*, 2014, pp. 188–193.

[17] (Jul. 2013). *Bitcoin Wiki, Toratm*. [Online]. Available: https://en.bitcoin.it/wiki/TorATM

[18] S. Bortzmeyer, *DNS Privacy Considerations*, document IETF RFC 7626, 2015.

[19] S. Bortzmeyer, *DNS Query Name Minimisation to Improve Privacy*. document IETF RFC 7816, 2016.

[20] D. Brown, "Resilient botnet command and control with Tor," in *Proc. DEFCON*, 2010, pp. 1–30.

[21] L. Chapin and M. McFadden, *Reserved top Level Domain Names*, document RFC 2606, 2011. [Online]. Available: http://bit.ly/1nIQ5cS

[22] D. G. Chapman. *Some Properties of the Hypergeometric Distribution With Applications to Zoological Sample Censuses*. Los Angeles, CA, USA: Univ. California, 1951.

[23] Q. A. Chen, E. Osterweil, M. Thomas, and Z. M. Mao, "MitM attack by name collision: Cause analysis and vulnerability assessment in the new gTLD era," in *Proc. IEEE SP*, May 2016, pp. 675–690.

[24] D. Dagon, N. Provos, C. P. Lee, and W. Lee, "Corrupted DNS resolution paths: The rise of a malicious resolution authority," in *Proc. NDSS*, 2008, pp. 1–34.

[25] *Digital Envoy, Digital Element Services*, accessed on Jul. 2, 2017. [Online]. Available: http://www.digitalenvoy.net/

[26] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *Proc. USENIX Secur. Symp.*, 2004, pp. 303–320.

[27] L. Fan, P. Cao, W. Lin, and Q. Jacobson, "Web prefetching between low-bandwidth clients and proxies: Potential and performance," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 27, no. 1, pp. 178–187, 1999.

[28] R. Garcia. (Oct. 2008). *Preventing Tor DNS Leaks*, accessed on Jul. 2, 2017. [Online]. Available: https://trac.torproject.org/projects/tor/wiki/doc/Preventing_Tor_DNS_Leaks

[29] R. Garcia. (2014). *Preventing Tor DNS leaks, The Tor Project*. [Online]. Available: http://bit.ly/1royLtU

[30] C. Grothoff, M. Wachs, H. Wolf, and J. Appelbaum, "Special-use domain names of peer-to-peer name systems," *IETF Internet Draft*, pp. 1–21, Nov. 2013.

[31] N. Hopper, "Challenges in protecting Tor hidden services from botnet abuse," in *Proc. FC*, Mar. 2014, pp. 316–325.

[32] Interisle Consulting Group, LLC. (2013). *Name Collision in the DNS, ICANN*. [Online]. Available: http://bit.ly/1iQVj5F

[33] N. Jiang, J. Cao, Y. Jin, L. E. Li, and Z.-L. Zhang, "Identifying suspicious activities through DNS failure graph analysis," in *Proc. IEEE ICNP*, Oct. 2010, pp. 144–153.

[34] W. John, S. Tafvelin, and T. Olovsson, "Trends and differences in connection-behavior within classes of Internet backbone traffic," in *Proc. PAM*, 2008, pp. 192–201.

[35] B. Jones, N. Feamster, V. Paxson, N. Weaver, and M. Allman, "Detecting DNS root manipulation," in *Proc. PAM*, 2016, pp. 276–288.

[36] R. P. Klemm, "WebCompanion: A friendly client-side Web prefetching agent," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 4, pp. 577–594, Jul./Aug. 1999.

[37] E. Kovacs. (Feb. 2014). *Backdoor.AndroidOS.Torec.A: First Tor-Based Trojan for Android*. [Online]. Available: http://bit.ly/1pte18L

[38] S. Krishnan and F. Monrose, "DNS prefetching and its privacy implications: When good things go bad," in *Proc. IUSENIX Conf. Large-Scale Exploits Emergent Threats: Botnets, Spyware, Worms, More*, 2010, p. 10.

[39] W. Kumari and P. Hoffman, "Securely distributing the DNS root," *IETF Internet Draft*, pp. 1–9, Jul. 2014.

[40] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas, "Circuit fingerprinting attacks: Passive deanonymization of tor hidden services," in *Proc. USENIX Secur. Symp.*, 2015, pp. 287–302.

[41] T. Leighton, "Improving performance on the Internet," *Commun. ACM*, vol. 52, no. 2, pp. 44–51, 2009.

[42] Z. Liu *et al.*, "Two days in the life of the DNS anycast root servers," in *Proc. PAM*, 2007, pp. 125–134.

[43] P. Luo *et al.*, "Leveraging client-side DNS failure patterns to identify malicious behaviors," in *Proc. IEEE*, Sep. 2015, pp. 406–414.

[44] D. McCoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker, "Shining light in dark places: Understanding the TOR network," in *Proc. PETs*, 2008, pp. 63–76.

[45] P. Mockapetris and K. J. Dunlap, "Development of the domain name system," in *Proc. ACM SIGCOMM*, 1988, p. 1.

[46] A. Mohaisen and O. Alrawi, "Unveiling ZEUS: Automated classification of malware samples," in *Proc. WWW*, 2013, pp. 829–832.

[47] A. Mohaisen, A. R. Kang, and K. Ren, "Does query blocking improve DNS privacy?" in *Proc. WISA*, 2016, pp. 1–4.

[48] Y. Nadji, R. Perdisci, and M. Antonakakis, "Still beheading hydras: Botnet takedowns then and now," *IEEE Trans. Depend. Sec. Comput.*, to be published, doi: 10.1109/TDSC.2015.2496176.

[49] G. Owen, "Hidden services and deanonymisation media.ccc.de," in *Proc. Chaos Commun. Congr.*, 2015, p. 1.

[50] R. Perdisci, I. Corona, and G. Giacinto, "Early detection of malicious flux networks via large-scale passive DNS traffic analysis," *IEEE Trans. Depend. Sec. Comput.*, vol. 9, no. 5, pp. 714–726, Sep. 2012.

[51] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient behavior-based tracking of malware-control domains in large ISP networks," in *Proc. IEEE/IFIP DSN*, Jun. 2015, pp. 403–414.

[52] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds," in *Proc. ACM CCS*, 2009, pp. 199–212.

[53] S. Rose and A. Nakassis, "Minimizing information leakage in the DNS," *IEEE Netw.*, vol. 22, no. 2, pp. 22–25, Apr. 2008.

[54] A. Sanatinia and G. Noubir, "OnionBots: Subverting privacy infrastructure for Cyber Attacks," in *Proc. IEEE/IFIP DSN*, Jun. 2015, pp. 69–80.

[55] N. Scaife, H. Carter, and P. Traynor, "OnionDNS: A seizure-resistant top-level domain," in *Proc. IEEE CNS*, Sep. 2015, pp. 379–387.

[56] M. J. Schwartz. (Dec. 2013). *Chewbacca Malware Taps TOR Network Dark Reading*. [Online]. Available: http://ubm.io/1nrFfKY

[57] I. Security. (Nov. 2010). *Invalid top Level Domain Queries at the Root Level of the Domain Name System*. [Online]. Available: http://bit.ly/1mDxRJO

[58] A. Simpson, "Detecting search lists in authoritative DNS," in *Proc. WPNC*, Mar. 2014, pp. 1–11.

[59] M. Smtih. (Nov. 2013). *Cryptolocker Crooks Charge 10 Bitcoins for Second-Chance Decryption Service. Network World*. [Online]. Available: http://bit.ly/ROxhPd

[60] S. Soulders. (Nov. 2013). *Prebrowsing*. [Online]. Available: http://www.stevesouders.com/blog/2013/11/07/prebrowsing/

[61] Y. Sun *et al.*, "RAPTOR: Routing attacks on privacy in TOR," in *Proc. USENIX Secur. Symp.*, 2015, pp. 271–286.

[62] D. Tarakanov. (Dec. 2013). *The Inevitable Move—64-bit Zeus has Come Enhanced With TOR SecureList*. [Online]. Available: http://bit.ly/1mIuAeR

[63] The TOR Project. (2014). *TOR Metrics Portal: Users The TOR Project*. [Online]. Available: http://bit.ly/1hrHqGp

[64] The TOR Project. (2014). *TOR: Overview*. [Online]. Available: http://bit.ly/1dZ2zvZ

[65] The TOR Project. (Feb. 2016). *TOR Metrics—Top-10 Countries by Directly Connecting Users*. [Online]. Available: https://metrics.torproject.org/userstats-relay-table.html

[66] The TOR Project. (Mar. 2016). *The Users of TOR*. [Online]. Available: https://bit.ly/1ud2CKh

[67] M. Thomas, Y. Labrou, and A. Simpson, "The effectiveness of block lists in preventing collisions," in *Proc. WPNC*, 2014, pp. 1–10.

[68] M. Thomas and A. Mohaisen, "Kindred domains: Detecting and clustering botnet domains using DNS traffic," in *Proc. WWW*, 2014, pp. 707–712.

[69] M. Thomas and A. Mohaisen, "Measuring the leakage of onion at the root: A measurement of tor's.onion pseudo-TLD in the global domain name system," in *Proc. WPES*, 2014, pp. 173–180.

[70] P. Wouters. (Dec. 2013). *DNS Operation Mailing List. DNSOP*. [Online]. Available: http://bit.ly/1roTIXw

[71] K. Xu, P. Butler, S. Saha, and D. Yao, "DNS for massive-scale command and control," *IEEE Trans. Dependable Secure Computing*, vol. 10, no. 3, pp. 143–153, May 2013.

[72] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *Proc. ACM SIGCOMM IMC*, 2010, pp. 48–61.

[73] 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 Trans. Netw.*, vol. 20, no. 5, pp. 1663–1677, Oct. 2012.

[74] K. Zournas. (Dec. 2013). *Pirate Bay Relocates to Thepiratebay AC*. [Online]. Available: http://bit.ly/1iQNEEz

Aziz Mohaisen (M'05–SM'15) received the Ph.D. degree from the University of Minnesota in 2012. He is currently an Associate Professor with the Department of Computer Science and the Department of Electrical and Computer Engineering, University of Central Florida. His research interests are in the areas of systems, security, privacy, and measurements. His research work has been supported by the NSF, AFOSR, and AFRL, and featured in MIT Technology Review, the New Scientist, Science Daily, The Verge, Deep Dot Web, and Slate, among others. He is a member of the ACM. He received the Doctoral Dissertation Fellowship from the University of Minnesota (2011), the Best Poster Award at the IEEE CNS (2013), the Summer Faculty Fellowship from the U.S. AFOSR (2016), the Best Student Paper Award at ICDCS (2017), and the Best Paper Award at WISA (2014), among other honors.

Kui Ren (M'07–SM'11–F'16) received the Ph.D. degree from the Worcester Polytechnic Institute. He is currently a Professor of computer science and engineering and the Director of the UbiSeC Lab, The State University of New York at Buffalo. He has authored 150 peer-reviewed journal and conference papers. His current research interest spans cloud and outsourcing security, wireless and wearable systems security, and mobile sensing and crowdsourcing. His research has been supported by the NSF, DoE, AFRL, MSR, and Amazon. He is a Distinguished Lecturer of the IEEE, a member of the ACM, and a past board member of the Internet Privacy Task Force, State of Illinois. He was a recipient of SEAS Senior Researcher of the Year in 2015, the Sigma Xi/IIT Research Excellence Award in 2012, and the NSF CAREER Award in 2011. He received several best paper awards including the IEEE ICNP 2011. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE WIRELESS COMMUNICATIONS, the IEEE INTERNET OF THINGS JOURNAL, and the IEEE TRANSACTIONS ON SMART GRID.