Who Is Answering My Queries: Understanding and Characterizing Interception of the DNS Resolution Path

Baojun Liu, Chaoyi Lu, Haixin Duan, and Ying Liu, *Tsinghua University;*
Zhou Li, *IEEE member;* Shuang Hao, *University of Texas at Dallas;* Min Yang, *Fudan University*

https://www.usenix.org/conference/usenixsecurity18/presentation/liu-baojun

This paper is included in the Proceedings of the
27th USENIX Security Symposium.

August 15–17, 2018 • Baltimore, MD, USA

ISBN 978-1-939133-04-5

Open access to the Proceedings of the
27th USENIX Security Symposium
is sponsored by USENIX.

Who Is Answering My Queries: Understanding and Characterizing Interception of the DNS Resolution Path

Baojun Liu*, Chaoyi Lu*, Haixin Duan*, Ying Liu*✉, Zhou Li†, Shuang Hao‡ and Min Yang§

* Tsinghua University, † IEEE member,
‡ University of Texas at Dallas, § Fudan University

Abstract

DNS queries from end users are handled by recursive DNS servers for scalability. For convenience, Internet Service Providers (ISPs) assign recursive servers for their clients automatically when the clients choose the default network settings. But users should also have the flexibility to use their preferred recursive servers, like public DNS servers. This kind of trust, however, can be broken by the *hidden interception of the DNS resolution path* (which we term as `DNSIntercept`). Specifically, on-path devices could spoof the IP addresses of user-specified DNS servers and intercept the DNS queries surreptitiously, introducing privacy and security issues.

In this paper, we perform a large-scale analysis of on-path DNS interception and shed light on its scope and characteristics. We design novel approaches to detect DNS interception and leverage 148,478 residential and cellular IP addresses around the world for analysis. As a result, we find that 259 of the 3,047 ASes (8.5%) that we inspect exhibit DNS interception behavior, including large providers, such as China Mobile. Moreover, we find that the DNS servers of the ASes which intercept requests may use outdated vulnerable software (deprecated before 2009) and lack security-related functionality, such as handling DNSSEC requests. Our work highlights the issues around on-path DNS interception and provides new insights for addressing such issues.

1 Introduction

Domain Name System (DNS) provides a critical service for Internet applications by resolving human-readable names to numerical IP addresses. Almost every Internet connection requires a preceding address lookup. DNS failures, therefore, will seriously impact users' experience of using the Internet services. Previous studies have shown that rogue DNS resolvers [38, 42], DNS transparent proxies [41, 55] and unauthorized DNS root servers [27] can damage integrity and availability of Internet communication.

In this work, we study an emerging issue around DNS, the *hidden interception of the DNS resolution path* (`DNSIntercept`) by on-path devices, which is not yet thoroughly studied and well understood by previous works. DNS queries from clients are handled by recursive nameservers to improve performance and reduce traffic congestion across the Internet. By default configuration, users' recursive nameservers are pointed to the ones operated by ISPs. On the other hand, users should have the flexibility to choose their own DNS servers or public recursive nameservers, such as Google Public DNS `8.8.8.8` [12]. However, we find on-path devices intercept DNS queries sent to public DNS, and surreptitiously respond with DNS answers resolved by alternative recursive nameservers instead. The on-path devices *spoof* the IP addresses of the users' specified recursive nameservers in the DNS responses (e.g., replacing the resolver IP address with `8.8.8.8` of Google Public DNS), so users will not be able to notice that the DNS resolution path has been manipulated.

The purposes of DNS interception include displaying advertisements (e.g., through manipulation of `NXDOMAIN` responses [56]), collecting statistics, and blocking malware connections, to name a few. However, such practices can raise multiple concerns: (1) The interception is not authorized by users and is difficult to detect on the users' side, which leads to ethical concerns; (2) Users have higher risks to put the resolution trust to alternative recursive DNS servers, which often lack proper maintenance (e.g., equipped with outdated DNS software), compared to well-known public DNS servers; (3) Certain security-related functionalities are affected or even broken, e.g., some alternative DNS resolvers do not provide DNSSEC.

[*] Part of this work was done during Baojun Liu's research internship at Netlab of 360. Part of this work was done in the Joint Research Center by Tsinghua University and 360 Enterprise Security Group.

In this paper, we conduct a large-scale analysis of DNSIntercept. Our study investigates the magnitude of this problem, characterizes various aspects of DNS interception, and examines the impact on end users. Finally, we provide insights that could lead to mitigation.

Challenges. There are two main challenges that we face to systematically analyze DNSIntercept. The first is to acquire clients belonging to different Autonomous Systems (ASes) to perform a large-scale measurement, which also should allow fine-tuning on the measurement parameters. The measurement frameworks proposed by previous works, including advertising networks [33], HTTP proxy networks [19, 36, 37, 52], and Internet scanners [42, 48], cannot fulfill the conditions at the same time. Another challenge is to verify whether the DNS resolution is intercepted rather than reaching users' designated recursive nameservers. Since on-path devices are able to spoof the IP addresses in the DNS responses, it is difficult to sense the existence of DNS interception merely from the clients.

Our approach. To address these challenges, we devise a new measurement methodology and apply it to two different large-scale experiments, named Global analysis and China-wide analysis. For Global analysis, we use a residential proxy network based on TCP SOCKS (not HTTP) which provides 36,173 unique residential IP addresses across 173 countries. This allows us to understand DNSIntercept from the world-wide point of view. However, this proxy network only allows us to send DNS packets over TCP SOCKS. To learn more comprehensive characteristics, we collaborate with a leading security company which provides network debugging tool for millions of active *mobile users*. We obtain DNS traffic over both UDP and TCP from 112,305 IP addresses (across 356 ASes), mainly within China.

To verify interception of DNS traffic, we register a set of domains (e.g., OurDomain.TLD), and use the authoritative nameservers controlled by us to handle resolutions. Each client is instructed to send DNS packets to a list of public DNS servers and query nonce subdomains under our domain names, e.g., UUID.Google.OurDomain.TLD (where we use Google to indicate we send the DNS requests to Google Public DNS). Note that we do not change DNS configurations of clients, but send DNS requests directly to the public DNS servers. Since each subdomain UUID is non-existent, the resolution cannot be fulfilled by DNS cache at any level and must go through the DNS server hierarchy. On the authoritative nameserver operated by us, we record the IP addresses that query the subdomain names we monitor. By checking whether the IP address belongs to the originally requested public DNS service, we can learn whether the DNS resolution is intercepted by an alternative resolver. According to Alexa traffic ranking [57], we select three popular public DNS servers as the target of our study, including Google Public DNS [12], OpenDNS [22], Dynamic DNS [9]. In addition, we build a public DNS server by ourselves, named EDU DNS, and use it for comparison.

Our findings. In this work, we develop the following key findings.

- Among the 3,047 ASes that we investigate, DNS queries in 259 ASes (8.5%) are found to be intercepted, including large providers, such as China Mobile. In addition, 27.9% DNS requests over UDP from China to Google Public DNS are intercepted.
- Interception policies vary according to different types of DNS traffic. In particular, DNS queries over UDP and those for A-type records sent to well-known public DNS services are more likely to be intercepted.
- DNS servers used by interceptors may use outdated software, e.g., all 97 DNS servers that we identify install old BIND software which should be deprecated after 2009, and are vulnerable to attacks like DoS [6]. Moreover, 57% of the DNS servers do not accept DNSSEC requests.
- DNSIntercept provides limited performance improvement to end users. In fact, 15.37% of the UDP DNS traffic to public DNS services are even faster than the counterpart issued by alternative DNS servers.

Contributions. The contributions of our study are summarized below.

- Understanding: We systematically measure DNSIntercept, which spoofs the IP addresses of users' specified DNS servers to intercept DNS traffic surreptitiously.
- Methodology: We design novel approaches to conduct large-scale analysis to characterize DNS interception, through 148,478 residential and cellular IP addresses around the world.
- Findings: Hidden interception behaviors are found to exist in some famous ASes, including those belonging to large providers like China Mobile. Our results show that DNS servers used by interceptors typically have less security maintenance and are vulnerable to attacks, which can damage the integrity and availability of DNS resolution for end users.
- Checking tool: We release an online checking tool at http://whatismydnsresolver.com [25] to help Internet users detect DNSIntercept.

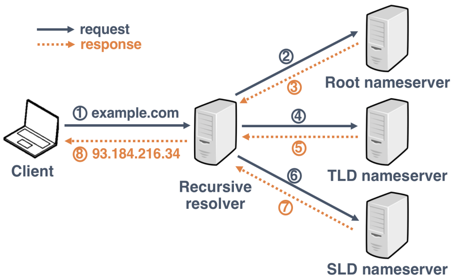


Figure 1: *Domain resolution process with a recursive resolver*

2 Threat Model and Mechanisms

In this section, we first give an overview of how domain names are translated into addresses using DNS. Then we introduce our threat model of DNSIntercept, with a taxonomy of interception paths according to our observation. Finally, we discuss the potential interceptors and their behaviors.

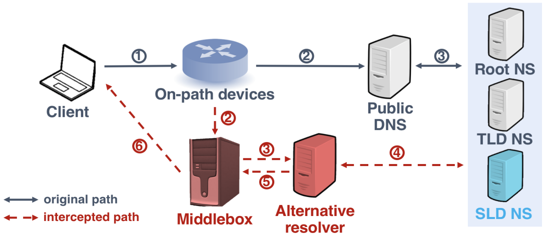


Figure 2: *Threat model*

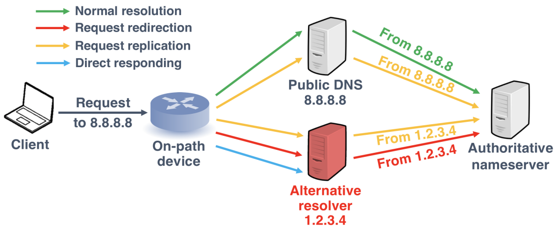


Figure 3: *Four DNS resolution paths (request shown only)*

2.1 Domain Resolution Process

DNS is a hierarchical naming system organized to handle domain resolutions at different levels. At the top of the hierarchy is DNS root which manages Top-Level Domains (TLD) resolutions. Second-Level Domains (SLD) are delegated to resolvers below DNS root. Consisting of labels from all domain levels, a fully qualified domain name (FQDN) specifies its exact location in the DNS hierarchy, from its lowest level to root. As a example, `www.example.com` is an FQDN, and its corresponding TLD and SLD are `com` and `example.com`.

When a client requests resolution of a domain, the resolution is typically executed by a recursive DNS resolver at first, which can be either assigned by ISP or specified by Internet users. Illustrated in Figure 1, the recursive resolver iteratively contacts root, TLD and SLD nameservers to resolve a domain name, and eventually returns the answer to the client. Therefore, intercepting DNS traffic to a recursive resolver directly affects the domain resolution process for users.

2.2 Threat Model

Figure 2 presents our threat model. We assume that users' DNS resolution requests are monitored by on-path devices. These on-path devices are able to intercept and selectively manipulate the route of DNS requests (e.g., by inspecting destinations and ports) which are sent to recursive resolvers like public DNS servers originally. The on-path devices either *redirect or replicate* the requests to alternative resolvers (typically, local DNS resolvers), which perform the standard resolution process. Finally, before responses are sent from alternative resolvers back to clients, the sources are *replaced* with addresses of the original resolvers. Therefore, from a client's perspective, DNS responses appear to come from the original DNS resolvers according to their source addresses, making the actual interception behaviors difficult to be discerned.

By default, in order to handle DNS requests, Internet users are assigned with local DNS resolvers by ISPs. In the mean time, users reserve the right to *specify their preferred recursive resolvers* to launch DNS requests (in particular, public DNS servers). However, our study shows that, for users using designated DNS servers, not only does DNSIntercept violate the will of users, but it also can bring in security issues.

Scope of study. We aim to measure and characterize DNSIntercept through large-scale data analysis. We focus on how DNS resolution paths between clients and *well-known public DNS resolvers* are tampered. Other types of network traffic manipulation mechanisms, such as BGP prefix hijacking [51] and unauthorized manipulation of DNS root servers [27], which have been systematically studied before, are not considered in our study.

Taxonomy of DNS resolution paths. In this study, we classify the mechanisms of DNS resolution into four categories, based on how the resolution path is constructed during the stage of the request. Except for *Normal resolution*, all the other three scenarios are regarded as DNSIntercept. Figure 3 presents the paths of DNS requests in the four DNS resolution mechanisms.

- *Normal resolution.* The resolution strictly follows the standard process. A DNS query sent by client *only* reaches the specified resolver, without being modified by any on-path device. The specified resolver performs the resolution by contacting authoritative nameservers if the resolution is not cached.
- *Request redirection.* The original DNS query sent to

user-specified resolver is dropped. In the meantime, an alternative resolver is used to perform the resolution. The specified resolver is completely removed from the resolution process.

- *Request replication.* The DNS query sent to user-specified resolver is not modified or blocked. However, the request is replicated by on-path devices, and handled by an alternative resolver at the same time. Consequently, the authoritative nameserver receives two identical requests from the user-specified resolver (i.e., *in-band* request) and the alternative resolver (i.e., *out-of-band* request [46]). When multiple responses are returned, typically the fastest one will be accepted by the client.
- *Direct responding.* Similar to *request redirection*, user's DNS request is redirected by the on-path device to an alternative resolver, without reaching the specified resolver. However, even for domains that are not cached, the alternative resolvers directly respond to the user without contacting any other nameservers.

2.3 Potential Interceptors

Anecdotally, on-path devices are mainly deployed by network operators like ISPs, in order to intercept DNS traffic [16]. However, the same kind of interception can be conducted by other parties, which are described below. We design our measurement methodology to minimize the chances of triggering interception unwanted to our study. Nevertheless, we acknowledge that other interceptors cannot be completely removed, due to the limitations of our methodology and vantage points.

- Censor and firewall. To block the access to certain websites (e.g., political and pornographic websites), censors and firewalls can manipulate DNS queries on their path and return fake responses. As studied by previous works [28], such DNS interception usually happens when the domain name contains sensitive keywords or matches a blacklist. We try to avoid triggering this type of interception by embedding a normal domain name in the DNS request.
- Malware and anti-virus (AV) software. For purposes like phishing, malware can change its host's configuration of DNS resolver and reroute DNS traffic to a rogue resolver [38]. On the other hand, AV software may intercept DNS queries as well, in order to prevent DNS requests of their clients from being hijacked. For example, Avast AV software provides this functionality by rerouting DNS requests from client machines to its own DNS server in an encrypted channel [3]. In both cases, the resolvers are likely to be directly controlled by operators behind malware and AV software, which are hosted by cloud providers or dedicated hosting services.
- Enterprise proxy. A large number of enterprises deploy network proxies to regulate the traffic between employees' devices and the Internet. Some proxies, like Cisco Umbrella intelligent proxy [5], are able to scrutinize DNS requests and determine whether the corresponding web visits are allowed. Similar to the AV setting, users are required to point their DNS resolvers to the proxy's resolver.

Since the mapping between the IP addresses of resolvers and their owners is unknown to us, alternative resolvers owned by parties other than ISPs, like AV and enterprise resolvers, can be included by our study. Straightforward classification using AS information is not always reliable. For example, an enterprise resolver might be mistakenly classified as an ISP resolver, if the enterprise rents a subnet of the ISP. We are currently developing the method to enable accurate resolver profiling to address this issue.

3 Methodology and Dataset

In this section, we describe the methodology and data collection of our study, which try to address the two major challenges described in Section 1. We begin by describing the high-level idea of our approach and the design requirements it needs to meet. Then, we elaborate the details of each component of our measurement framework and how we obtain *a large volume* of globally distributed vantage points. Finally, ethical concerns regarding our data collection are discussed.

3.1 Overview

We first illustrate our methodology of identifying `DNSIntercept`, which includes *Request redirection*, *Request replication* and *Direct responding*.

Approach. Detecting `DNSIntercept` is conceptually simple. Recalling *Normal resolution*, upon receiving a request from a client, a recursive resolver tries to contact the authoritative nameserver for an answer, if the result is not cached. However, as shown in Figure 3, when interception takes place, requests forwarded by alternative resolvers reach authoritative nameservers.

Therefore, our approach to identify interception contains the following steps. We (1) instruct a client to send a DNS request about one of our controlled domains to a public resolver *A*; (2) record its corresponding request at our authoritative nameservers, which originates from recursive resolver *B*; and (3) compare *A* with *B*. As a complementary step, we also (4) validate the response eventually received by the client.

Only when A matches B, the request is regarded as a *Normal resolution*. Otherwise, for each request sent by the client to public resolver A that gets a valid response, if (1) no corresponding request is captured by authoritative nameservers, we regard it as *Direct responding*; if (2) a single request not from resolver A is captured, we regard it as *Request redirection*; if (3) multiple identical requests from resolvers, one of them being A, are captured, we regard it as *Request replication*.

Design requirements. Our methodology should meet several requirements to obtain valid results.

Firstly, the queried domain name of each request from client should be different to avoid caching. Secondly, as we capture packets separately from clients and authoritative nameservers, we should be able to correlate a request from client with the one captured by our authoritative nameserver in the same resolution. As will be discussed in Section 3.2, the two issues are addressed by uniquely prefixing each requested domain name.

Thirdly, the clients in our study should be diverse, being able to send DNS packets directly to specified public resolvers, even when local DNS resolvers have been assigned by ISPs. Fourthly, aiming to study interception characteristics in depth, the vantage points are expected to issue diversified DNS requests (e.g., requests over different transport protocols and of different RR types). The measurement infrastructure used by previous works, including advertising networks [33], `HTTP` proxy networks [19, 36, 37, 52] and Internet scanners [42, 48], do not meet the requirements. How the two issues are addressed will be discussed in Section 3.3.

Finally, public DNS services are accessed by clients using anycast addresses (e.g., `8.8.8.8` of Google DNS). These addresses rarely match the unicast addresses (e.g., `74.125.41.0/24` of Google) when the requests are forwarded to our authoritative nameservers. We propose a novel method to identify the egress IPs of a public DNS service, as will be elaborated in Section 3.2.

3.2 Methodology

Before presenting our methodology, we first illustrate an interception model with possible elements that interceptors may consider. On this basis, we elaborate our methodology regarding how DNS requests are generated and how egress IPs of public DNS services are identified.

Interception model. On-path devices are deployed to inspect and manipulate DNS packets. We consider each DNS packet to be represented by a tuple of five fields:

$$\{\mathit{Src\ IP},\ \mathit{Dst\ IP},\ \mathit{Protocol},\ \mathit{RR\ Type},\ \mathit{Requested\ Domain}\}$$

Each field could decide how interception is actually carried out. So, to understand `DNSIntercept` in a comprehensive way, we need DNS packets with diversified field values. To this end, we construct a client pool with a large volume of source IPs (i.e., client IPs) distributed globally. Destination IPs point to our specified public DNS resolvers. Investigating all public resolvers would take a tremendous amount of time and resources, so we narrow down to three representative and widely-used public DNS services according to Alexa traffic ranking [57], including Google Public DNS [12], OpenDNS [22] and Dynamic DNS [9]. As a supplement, we also include a self-built public DNS service, named EDU DNS, to make comparisons. Transport protocol can be either `TCP` or `UDP`. As for resource record (RR), five kinds of security-related records are considered [43], including `A`, `AAAA`, `CNAME`, `MX` and `NS`. Lastly, we registered four domains exclusively for our study, spanning four TLDs including a new gTLD (`com`, `net`, `org` and `club`). We avoid any sensitive keyword in the domain names.

Generating DNS requests. In this study, we need to address the issue of the inconsistent source IPs between a request from client and its corresponding request(s) supposed to be launched by recursive resolvers. To this end, we devise a method to link those requests through unique domain prefix. The prefix includes a distinct UUID generated for each client (representing SrcIP) and a label of public DNS service which is supposed to handle the resolution (representing DstIP). By considering RR Type at the same time, we are able to identify DNS packets in the same resolution. For instance, when a client launches a DNS `A`-type request for `UUID.Google.OurDomain.TLD`, this request is supposed to be handled by Google Public DNS. Its corresponding request captured by authoritative nameservers should be `A`-type as well and match every label in the domain prefix.

Generating DNS responses. Under *Request replication* scenario, a client receives an in-band response and an out-of-band response. We want to classify these two cases but the regular response from the authoritative nameservers cannot tell such difference. As such, we need a reliable mechanism to link the response received by the client to that from our authoritative nameservers. Similar to the prior component, we encode a unique nonce in the response. In particular, our authoritative nameservers hash the timestamp, source address and requested domain name together, and derive a unique response from the hash string fitting to the record type. For instance, once receiving an `A`-type request, the response is an IPv4 address converted from the hash value (using the last 32 binary bits of the hash).

To notice, the response synthesized by this approach might point the client to unwanted servers. For example, the response IP could be used by botnet servers accidentally. We want to emphasize that no actual harm will be introduced to our vantage points, because clients' actions are no more than DNS lookups. There is *no follow-up connection* to the servers.

Resolvers are able to manipulate TTL value of a response based on what is returned from authoritative nameservers and their policies. We attempt to measure this scenario by selecting a random TTL value between 1 and 86400.

Identifying egress IPs of public DNS. Our next task is to identify whether a source IP contacting our authoritative nameservers belongs to a public DNS service, i.e., is an *egress IP*. From the client's point of view, *anycast address* is accessible, which essentially represents a proxy in front of a set of recursive resolvers. Such design is for load balancing. However, the unicast addresses of the affiliated resolvers, which are observed by our authoritative nameservers, typically do not match their anycast addresses. The ownership of the anycast addresses are usually not known to public audiences. As such, we need to infer the ownership.

Previous studies leveraged IP WHOIS data and information from public forums [37,49] to identify egress IPs, which are not sufficiently accurate when examined on our data. We propose a more reliable method leveraging DNS `PTR` and `SOA` record. Our method is based on an assumption that, instead of scattered IP addresses, a public DNS service tends to use addresses aggregated in several network prefixes (e.g., `/24` networks). Therefore, for ease of management, identity information of an IP address is usually embedded in `PTR` and `SOA` records by network administrators. We validate this assumption for the top 12 public DNS services according to Alexa traffic [57], from different vantage points in five ASes, and find *all* 12 DNS services embed identity information in either `PTR` (e.g., Norton ConnectSafe) or `SOA` records (e.g., Freenom), or both (e.g., OpenDNS). As an example, responses from reverse lookups of egress IPs of Google Public DNS are all `dns-admin.google.com`.

In practice, for an IP that contacts our authoritative nameservers, we first perform its *reverse DNS lookup*. Subsequently, we recursively request the `SOA` record of the responded domain name and build its `SOA` dependencies (5 iterations), which is similar to [43]. If particular SLDs (e.g., `opendns.com`) present in the dependency chain, we regard the address as an egress IP of the corresponding public DNS service. For instance, the `PTR` record of `45.76.11.166` (AS20473; Choopa, LLC) is `hivecast-234-usewr.as15135.net`. The `SOA` record of this domain name is `ns0.dynamicnetworkservices.net`, hence we regard `45.76.11.166` as an egress IP of Dynamic DNS.

Using this method, we are able to infer ownership of 85% addresses that contact our authoritative nameservers. Meanwhile, compared to IP WHOIS method, new egress ASes of public DNS services are discovered by our method. For instance, AS20473 (for Dynamic DNS) and AS30607 (for OpenDNS) are found to be egress ASes, yet they cannot be found with IP WHOIS or BGP information.

Discussion. As discussed in Section 2.3, our methodology may not be able to accurately distinguish whether an interception is caused by network operators or other interceptors. Secondly, by configuring fake `PTR` and `SOA` records for alternative resolvers, their egress IPs will not be correctly identified. However, those furtive changes should be observed from Passive DNS data, such as that managed by Farsight [10] and DNS Pai [15]. At present, we do not include Passive DNS data due to the access limit and consider to include it in our future work. Meanwhile, `PTR` records have been proved to be a reliable source to classify IP addresses in previous studies. As an example, [48] used `PTR` records to identify domains hosted on particular CDNs.

3.3 Vantage Points

Our study requires a large number of clients distributed globally. Besides, our clients should be able to send customized DNS requests about a domain to a specified public resolver. To this end, we first leverage a *residential proxy network* based on `TCP SOCKS` which allows us to directly send DNS packets from globally-distributed clients, to depict a global landscape of `DNSIntercept` (this phase is named *Global analysis*). This experiment, however, cannot reveal full characteristics of `DNSIntercept`, because the proxy network does not allow us to change every field of DNS request. Therefore, we design another experiment in which we cooperate with our industrial partner who develops security software installed by millions of active users. We implement a measurement script and integrate it to the software's network debugger module. When the change is delivered to the client, a consent is displayed and the script is not executed until the client acknowledges the change. As clients in this experiment are mainly from China, we named it *China-wide analysis*.

Global analysis. Proxy networks have been used by previous studies as measurement vantage points [37, 52]. However, DNS requests from clients under those proxy networks are only allowed to go to the pre-assigned local DNS resolvers, which doesn't satisfy our requirement. To address the issue, we leverage a `SOCKS` proxy network called *ProxyRack* [14], which allows us to send customized DNS requests to *any specified resolver* over `TCP`.

The network architecture of ProxyRack is shown in Figure 4. It interacts with our measurement client with a Super-proxy. When DNS packets are sent by our machine, they go to affiliated nodes and finally leave the network from diverse exit nodes. The packets are forwarded to the recursive resolvers which are supposed to contact

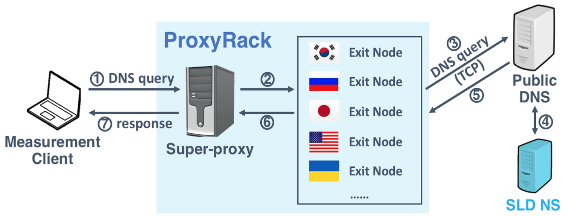


Figure 4: *Network architecture of ProxyRack*

our authoritative nameservers. Therefore our client pool is in fact composed of those exit nodes. ProxyRack has recruited more than 100K nodes [14], so we are able to send DNS requests from nodes distributed globally to public resolvers, and verify the responses, both by interacting with the Super-proxy. However, ProxyRack only accepts DNS requests over `TCP`, which is only used by a small fraction of DNS requests in the real-world settings. Therefore, we conduct the next experiment to measure interception over `UDP` and other factors.

China-wide analysis. We cooperate with an international security company who has developed mobile security software with millions of users. The software has been granted the permission to send arbitrary network requests when installed, so we are able to collect fine-grained DNS data.

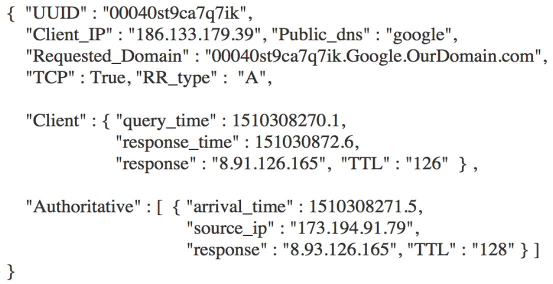
The major concerns of this experiment are around ethics and privacy, and we carefully address these concerns as briefly described below (more details are covered in Section 3.5). Firstly, the module where we implement our measurement script (sending and receiving DNS packets) comes with a consent, and the software has to be run manually with granted permission from users. Secondly, although sending diverse DNS requests from a client helps us comprehensively understand `DNSIntercept` characteristics, we try to avoid generating excessive traffic on user's devices. This choice limits the diversity of our DNS requests. Finally, our script only captures DNS packets of domains exclusively registered for this study, thus the data deemed private, like requests to social networks, is not collected.

Distribution of DNS packets. According to our inception model described earlier, to generate as diverse DNS packets as possible, we should launch DNS requests from a client under all four different SLDs, of all five RR types, over both `TCP` and `UDP`, and to all four public DNS services. However, we believe it is difficult due to ethical concerns and limitations of vantage points.

In the phase of *Global analysis*, ProxyRack only accepts `TCP` traffic. Meanwhile, the proxy network has a rate limit of submitting requests, so we have to be careful in crafting DNS requests. Therefore, from each client, we only request DNS `A` record, the most common RR type, of our `com` domain name using `TCP`-based lookups, to all four public DNS services.

Table 1: *Statistics of collected dataset*

Phase	# Request	# UUID	# IP	# Country	# AS
Global	1,652,953	476,153	36,173	173	2,691
China-wide	4,584,413	400,491	112,305	87	356



```
{ "UUID" : "00040st9ca7q7ik",
  "Client_IP" : "186.133.179.39", "Public_dns" : "google",
  "Requested_Domain" : "00040st9ca7q7ik.Google.OurDomain.com",
  "TCP" : True, "RR_type" :  "A",

  "Client" : { "query_time" : 1510308270.1,
               "response_time" : 151030872.6,
               "response" : "8.91.126.165",  "TTL" : "126"  } ,

  "Authoritative" : [  { "arrival_time" : 1510308271.5,
                         "source_ip" : "173.194.91.79",
                         "response" : "8.93.126.165", "TTL" : "128" } ]
}
```

Figure 5: *Format of collected data*

In the phase of *China-wide analysis*, while sending requests from a software client is more flexible and efficient, we ought to limit the quantity of our requests to avoid excessive traffic. Therefore, for each client, we consider two public DNS services, two TLDs, one transfer protocol which are all *randomly* selected, and all five RR types. In addition, we also send a single request to a client's assigned local DNS resolver.

3.4 Datasets

Table 1 summarizes our collected dataset in both phases. In total, we obtain DNS traffic from 148,478 distinct residential and cellular IP addresses globally.

Format of dataset. Through launching DNS requests from clients, monitoring DNS queries on authoritative nameservers and capturing DNS responses, we are able to "connect the dots" for each DNS resolution. To perform this correlation analysis, our collected data for each DNS request is stored in a `JSON` format shown in Figure 5. For each client, we capture each request and the corresponding response. At our authoritative nameservers, we collect the arrival time and source IP of the corresponding request(s), as well as the response returned.

Geo-distribution of clients. Leveraging ProxyRack and security software, we address the challenge of obtaining clients globally. Here we use the geo-distribution [20] of distinct IPs to give an evaluation of our clients. In *Global analysis*, our collected clients span more than 36K unique addresses in 173 countries. Figure 6 shows the geo-distribution and our clients cover the majority of countries in the world, with Korea, Russia, Japan and the US topping the list. In *China-wide analysis*, the clients we obtain are mostly from China, but still span 87 different countries.

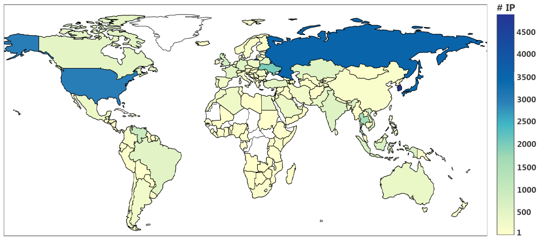


Figure 6: *Geo-distribution of proxy nodes*

3.5 Ethics

Our methodology could introduce a few ethical concerns. Here, we discuss them before presenting our results. Throughout this study, we take utmost care to protect users from side-effects that may be caused by our experiment.

In *Global analysis*, ProxyRack is *commercial*. We pay for the proxy plans and totally abide by their terms of service. More importantly, owners of exit nodes (i.e., our vantage points) have an agreement with ProxyRack that permits ProxyRack traffic to exit from their hosts. Therefore, launching DNS requests from ProxyRack adheres to the granted permission from owners of exit nodes.

In *China-wide analysis*, we implement our measurement script in a network debugger module of security software with millions of users. To avoid ethical concerns, this network debugger module comes with a *one-time consent* stating its procedure and data collected. Users reserve the right of choosing whether to install this security software and whether to run this module containing our measurement script manually. In addition, the user has the option to install the software without the measurement module.

Regarding our methodology, we carefully craft our DNS requests and limit their quantities to avoid excessive network traffic. Meanwhile, we only launch DNS lookups of domain names exclusively registered and used for this study on each client, without connecting to any host except for DNS resolvers.

Through said approaches, we believe we have minimized the threat to user's privacy and security in the experiments, as all operations are under granted permission from users, and we do not collect any data except for DNS resolutions under the limited scope.

4 TCP DNS Interception Analysis (Global)

To conduct a global measurement of DNSIntercept, we first leverage a *residential proxy network* based on TCP SOCKS. Here, we report our measurement results and analysis in the phase of *Global Analysis*, by showing its landscape and characteristics.

4.1 Scope and Magnitude

We first investigate the global landscape of DNSIntercept from three aspects. Firstly, using our methodology described in the previous section, we identify and classify interception by cross-matching resolver addresses. Secondly, we validate whether correct responses are eventually accepted by clients. Here we regard a response of an FQDN accepted by the client to be *correct*, only when its RR value is *identical* to the RR of the same FQDN which is responded by our authoritative nameserver; otherwise, the response is *incorrect*, which is tampered on its way back. Thirdly, specifically for *Request replication* scenario, interceptors may hope to use out-of-band DNS packets [46] (i.e., responses of replicated lookups) to replace in-band ones (i.e., responses of original lookups). To this end, replicated lookups are often made faster than original ones. Through our design of authoritative nameservers, we present how many in-band responses are eventually accepted by the client.

Table 2 summarizes our findings in *Global analysis*. All of three interception types are found in our dataset. In total, **198** (out of 2,691, 7.36%) client ASes witness intercepted traffic, in 158 of which queries to Google Public DNS are intercepted. The ratio of *Direct responding* is significantly low, since it is impossible for resolvers to correctly resolve a domain without contacting nameservers, and thus this behavior is distinguishable from clients. Moreover, we also find that compared to the less-known EDU DNS (0.45% packets intercepted), DNS traffic sent to renowned public DNS services are more likely to become victims of DNSIntercept (e.g., 0.66% for Google DNS).

As for responses accepted by clients, all except one are correct, suggesting major responses of intercepted queries are not tampered. The one incorrect response[1] is accepted by a client in AS36992 (EG, ETISALAT-MISR), which is caused by domain blocking. On the other hand, for *Request replication*, in-band responses accepted by clients are in the minority. Among 23 ASes where replicated queries are found, only clients in 2 of them (AS9198 JSC Kazakhtelecom, and AS31252 Star-Net Solutii SRL) receive in-band responses.

4.2 AS-Level Characteristics

As described in our landscape study, intercepted DNS requests are found in 198 client ASes, with different modes and ratio. We now analyze the AS-level characteristics of DNSIntercept, by focusing on the 158 ASes with intercepted requests to Google Public DNS.

[1] Response: `146.112.61.109`, its reverse lookup pointing to `hit-block.opendns.com`

Table 2: *Summary of interception (Global analysis). All DNS packets are over TCP. Under each type, ratio is used for correct answers and raw numbers are used for incorrect and in-band ones.*

Public DNS	# Request	Interception	Normal Resolution		Request Redirection		Request Replication			Direct Responding	# Problematic
		Ratio	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect	In-band	Incorrect	Client AS
Google DNS	391,042	0.66%	99.34%	0	0.41%	0	0.25%	0	8	0	158
OpenDNS	431,633	0.64%	99.36%	1	0.26%	0	0.38%	0	0	0	139
Dynamic DNS	407,632	0.53%	99.47%	0	0.29%	0	0.24%	0	0	0	116
EDU DNS	422,646	0.45%	99.55%	0	0.27%	0	0.18%	0	9	2	121

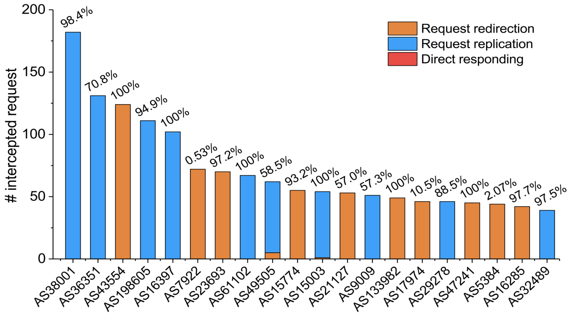


Figure 7: *Top 20 ASes with most intercepted requests to Google Public DNS. Ratio of intercepted requests over total requests to Google DNS is shown above for each AS.*

Types and ratio of interception. Figure 7 illustrates the quantity and types of intercepted requests to Google from each AS, as well as the ratio over its total requests to Google. We find that among the top 20 ASes, most of them only witness *one type* of interception, which indicates a unified policy of DNS traffic filtering within an AS. Both *Request redirection* and *Request replication* are found in top ASes.

Regarding interception ratio, we find that 82 (52%) of all 158 problematic ASes intercept more than 90% of DNS requests sent to Google, such as AS38001 and AS43554. By contrast, 50 (32%) ASes have an interception rate lower than 0.5 (e.g., AS17974). We speculate it to be a result of interception policies and deployment of on-path devices, which may cover only limited locations within the AS.

Country-level analysis. We further investigate the country distribution of the 158 ASes, and find they span 41 countries. Russia tops the list and accounts for 44 ASes (28%), followed by the US (15 ASes, 9%), Indonesia (8 ASes, 5%), Brazil and India (7 ASes each, 4%).

Targeted public DNS services. We find that in some ASes, only queries sent to specific public DNS services are intercepted. Table 3 shows the results of top 10 ASes with most intercepted requests to Google. While the majority of ASes do not, we find 2 ASes (AS43554 and AS15774) *exclusively intercept* traffic to Google DNS.

Alternative resolvers. When `DNSIntercept` takes place, alternative resolvers contact our authoritative nameservers. For each of top 10 ASes with most intercepted requests, Table 4 shows their alternative resolvers which handle the resolution. We can conclude that for top 10 ASes, alternative resolvers actually *locate in the same AS* as the clients.

Table 3: *Targeted public DNS services of top 10 ASes*

AS (Country)	Organization	Google	Others
AS38001 (SG)	NewMedia Express	✓	✓
AS36351 (US)	SoftLayer Technologies	✓	✓
AS43554 (UA)	Cifrovye Dispetcherskie	✓	
AS198605 (CZ)	AVAST Software	✓	✓
AS16397 (BR)	EQUINIX BRASIL SP	✓	✓
AS7922 (US)	Comcast Cable	✓	✓
AS23693 (ID)	PT. Telekomunikasi	✓	✓
AS61102 (IS)	Interhost Communication	✓	✓
AS49505 (RU)	Network of Selectel	✓	✓
AS15774 (RU)	TransTeleCom	✓	

Table 4: *Alternative DNS resolvers of top 10 ASes*

AS (Country)	Organization	Alternative resolvers
AS38001 (SG)	NewMedia Express	113.29.230.* (38001)
AS36351 (US)	SoftLayer Technologies	169.57.1.* (36351)
AS43554 (UA)	Cifrovye Dispetcherskie	178.209.65.* (43554)
AS198605 (CZ)	AVAST Software	77.234.42.* (198605)
AS16397 (BR)	EQUINIX BRASIL SP	177.47.27.* (16397)
AS7922 (US)	Comcast Cable	69.241.93.* (7922)
AS23693 (ID)	PT. Telekomunikasi	114.125.67.* (23693)
AS61102 (IS)	Interhost Communication	185.18.205.* (61102)
AS49505 (RU)	Network of Selectel	95.213.193.* (49505)
AS15774 (RU)	TransTeleCom	188.43.31.* (15774)

Traffic ranking of problematic ASes. We expect `DNSIntercept` tends to take place in ASes with lower reputation since such behavior should be furtive. However, by correlating problematic client ASes with their traffic ranking logged by CAIDA [2], our result shows that interception also exists in reputable ASes. Presented in Figure 8, problematic ASes span a diverse ranking. As an example, both *Request redirection* and *Request replication* are observed under AS3356, which is ranked the first according to CAIDA.

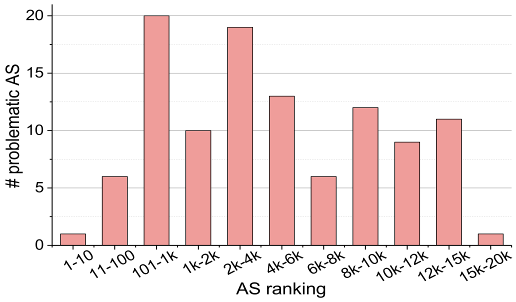


Figure 8: *Traffic ranking of problematic ASes*

Case study. AS7922 (ranked 22 according to CAIDA) belongs to Comcast Cable Communications, LLC, a renowned ISP based in the US. Among our 13,466 DNS requests sent from this AS to Google DNS, 72 (0.53%) are redirected, with alternative resolvers outside Google actually contacting our authoritative nameservers. The IP prefix of these alternative resolvers is `76.96.15.*` (also locating in AS7922), whose `PTR` record points to `hous-cns14.nlb.iah1.comcast.net`. We also find that clients of the 72 intercepted requests are grouped in several prefixes (e.g., `67.160.0.0/11`). As the interception ratio is low, we speculate that on-path devices conducting `DNSIntercept` are deployed only in limited sub-networks within this AS. Also, it is possible that interception devices are deployed by customers of Comcast, instead of AS-level network operators.

4.3 Summary of Findings

Our measurement findings in *Global analysis* are summarize below.

- `DNSIntercept` is found to exist in **198** ASes globally. For the public DNS services we investigate, up to 0.66% of DNS requests over `TCP` sent from the client are intercepted. Meanwhile, interception behaviors exist in both reputable ASes and those with a lower ranking.
- As for interception scenarios, *Request redirection* and *Request replication* are both found in top 20 ASes with most intercepted requests to Google DNS. *Direct responding* is rare, as it is more likely to be discovered by clients.
- For most of top 20 ASes, only one interception type is found within an AS, suggesting unified interception policies. Moreover, it is found that an interceptor can *exclusively* intercept DNS traffic sent to specific public DNS services (e.g., Google Public DNS). The concrete strategies differ among different interceptors. We also discover 82 ASes are intercepting more than 90% DNS traffic sent to Google Public DNS.

5 TCP/UDP DNS Interception Analysis (China-wide)

In order to learn more characteristics about `DNSIntercept`, we design another experiment called *China-wide analysis*. In this section, we first, on the whole, give an analysis on interception characteristics towards different kinds of DNS packets. Moreover, we also discuss issues regarding DNS lookup performance and response manipulation introduced by `DNSIntercept`. Finally, we discuss potential motivations of such interception behavior.

5.1 Interception Characteristics

In our experiment setup, we launch DNS packets with diverse field values from our clients to public DNS services. On the whole, by comparing the interception ratio of packets of different field values, we first investigate what kinds of packets are more likely to be intercepted. Table 5 presents our summary of results in this phase.

Transport protocol. Compared to those over `TCP`, DNS requests over `UDP` from clients are more likely to be intercepted. For instance, **27.9%** DNS requests sent to Google Public DNS over `UDP` are redirected or replicated, the ratio being only **7.3%** when it is through `TCP`. In fact, most of DNS requests in the real world are over `UDP`, and intercepting `UDP` traffic is technically easier. Therefore, it is reasonable for `UDP` traffic to be primarily intercepted.

Targeted public DNS services. `DNSIntercept` targets DNS traffic sent to not only renowned public DNS services but also less prevalent ones. Similar to our findings of *Global analysis*, the interception ratio for renowned public resolvers is significantly higher. For instance, 27.9% `UDP`-based DNS packets sent to Google are intercepted, the ratio being 9.8% for our in-house EDU DNS.

DNS RR Types. We find that `A`-type requests are slightly preferred to be intercepted, possibly because it's the most common RR type. Meanwhile, we notice in Table 5 that for *Request replication*, clients receive *no in-band responses* of `CNAME`, `NS` or `MX`-type requests. We speculate that on-path devices, while replicating requests, *block* responses of the three RR types from public DNS services, reiterating the unethical nature of the interception behavior.

TLD of requested domain. Due to the extra time overhead introduced by inspecting requested domain names, it is unlikely that on-path devices specify certain domains and merely intercept requests of them. Shown in Table 6, the ratio of intercepted DNS requests does not change much for domains under different TLDs.

Case Study. In total, we find **61** ASes out of 356 (17.13%) are problematic. In Table 7, we list the top five ASes from which most DNS requests (292K in total) are sent by the client. As our clients are mainly from China, the top 5 ASes belong to three largest Chinese ISPs. We find that ASes of China Mobile have *significantly higher interception ratio* than ASes of other Chinese ISPs. Regarding alternative resolvers, they are mostly locating in the same AS as their clients. However, we find that they may also locate in a different AS of the same ISP (e.g., AS56046 in Table 7).

Table 5: *Summary of interception (China-wide analysis)*

Public DNS	RR Type	Normal Resolution				Request Redirection				Request Replication						Direct Responding	
		Correct		Incorrect		Correct		Incorrect		Correct				Incorrect		Incorrect	
		UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	In / Out	TCP	In / Out	UDP	TCP	UDP	TCP
	Total	72.1%	92.7%	0	1	22.3%	7.2%	5	2	5.6%		0.2%		2	0	21	4
Google UDP:556,081 TCP:463,066	A	69.0%	92.4%	0	1	23.9%	7.4%	2	2	7.1%	2,191/5,860	0.2%	195/10	2	0	15	0
	AAAA	73.8%	92.6%	0	0	22.3%	7.3%	1	0	3.8%	1,126/3,130	0.2%	147/6	0	0	4	0
	CNAME	71.2%	92.5%	0	0	22.9%	7.3%	0	0	5.9%	0/6,589	0.2%	0/142	0	0	1	1
	NS	71.4%	92.5%	0	0	22.9%	7.3%	0	0	5.7%	0/6,393	0.2%	0/147	0	0	1	1
	MX	75.2%	93.3%	0	0	19.2%	6.5%	2	0	5.6%	0/6,595	0.2%	0/145	0	0	0	2
	Total	87.4%	99.1%	0	0	7.8%	0.7%	7	0	4.8%		0.2%		0	0	27	7
OpenDNS UDP:589,933 TCP:441,199	A	84.9%	98.9%	0	0	8.3%	0.7%	2	0	6.8%	2,901/5,327	0.4%	362/22	0	0	13	6
	AAAA	89.9%	99.1%	0	0	7.3%	0.7%	3	0	2.8%	1,593/1,709	0.2%	197/17	0	0	6	0
	CNAME	87.2%	99.1%	0	0	7.8%	0.7%	0	0	5.0%	0/5,952	0.2%	0//208	0	0	3	0
	NS	87.5%	99.2%	0	0	7.6%	0.7%	0	0	4.9%	0/5,888	0.2%	0/153	0	0	2	1
	MX	87.5%	99.2%	0	0	7.8%	0.7%	2	0	4.8%	0/5,122	0.2%	0/139	0	0	3	0
	Total	83.9%	97.7%	6	0	9.7%	1.9%	5	0	6.3%		0.4%		0	0	16	6
Dyn DNS UDP:461,263 TCP:164,582	A	83.5%	98.0%	4	0	8.8%	1.5%	0	0	7.7%	2,499/5,760	0.4%	89/94	0	0	13	5
	AAAA	88.6%	98.2%	0	0	8.3%	1.5%	3	0	3.1%	1,455/1,817	0.3%	38/80	0	0	2	0
	CNAME	85.8%	98.2%	0	0	8.7%	1.6%	0	0	5.5%	0/5,927	0.3%	0/114	0	0	0	0
	NS	74.9%	89.6%	1	0	15.2%	9.2%	0	0	9.8%	0/5,930	1.1%	0/79	0	0	1	0
	MX	82.8%	97.8%	1	0	10.0%	1.9%	2	0	7.2%	0/5,709	0.3%	0/87	0	0	0	1
	Total	90.2%	98.9%	5	0	6.3%	0.9%	3	0	3.5%		0.2%		0	0	21	6
EDU DNS UDP:701,128 TCP:409,019	A	88.0%	98.8%	5	0	7.0%	1.0%	0	0	5.0%	5,430/1,542	0.2%	143/20	0	0	8	2
	AAAA	91.6%	98.9%	0	0	6.2%	0.9%	3	0	2.2%	2,597/459	0.2%	114/19	0	0	1	1
	CNAME	90.0%	98.9%	0	0	6.5%	1.0%	0	0	3.5%	0/4,864	0.2%	0/126	0	0	4	1
	NS	90.1%	98.9%	0	0	6.4%	1.0%	0	0	3.5%	0/4,884	0.2%	0/132	0	0	4	2
	MX	91.1%	98.9%	0	0	5.6%	0.9%	0	0	3.4%	0/4,667	0.2%	0/139	0	0	4	0

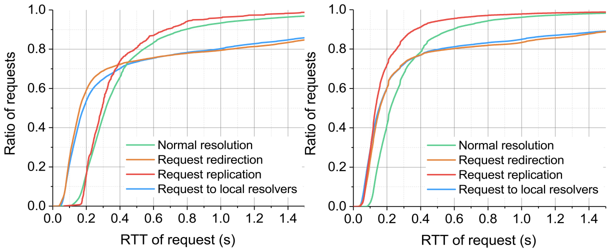
Table 6: *Interception ratio of domains under different TLDs*

TLD	# Request	Normal	Redirection	Replication
com	945,954	83.60%	14.80%	1.50%
net	947,532	83.40%	15.10%	1.50%
org	954,221	83.60%	14.90%	1.50%
club	948,707	83.60%	14.90%	1.50%

Table 7: *Top 5 ASes with most DNS requests*

AS	Organization	Redirection	Replication	Alternative resolvers
AS4134	China Telecom	5.19%	0.26%	116.9.94.* (4134)
AS4837	China Unicom	4.59%	0.51%	202.99.96.* (4837)
AS9808	China Mobile	32.49%	8.85%	112.25.12.* (9808)
AS56040	China Mobile	45.09%	0.04%	120.196.165.* (56040)
AS56041	China Mobile	23.42%	0.09%	112.25.12.* (56046[1])

[1] AS56046 also belongs to China Mobile.



(a) TCP (b) UDP

Figure 9: *ECDF of RTT of DNS requests*

5.2 Performance of DNS Lookups

As claimed by one large ISP [24], DNSIntercept is designed for improving the performance of DNS lookups, and we would like to investigate whether this is true. We regard RTT (round-trip-time), the interval between sending request and receiving answer measured by client, as the indicator of DNS lookup performance. Both timestamps can be recorded by clients in our study.

Figure 9 presents the ECDF of RTT of DNS requests. We find that performance impacts introduced by each type of interception are different. As for *Request replication*, when DNS requests are sent over UDP by clients, performance improvement does exist, with more requests of shorter RTT compared to *Normal resolution*. However, over TCP, due to the cost of establishing connections, the improvement is less obvious. On the other hand, for *Request redirection*, the performance improvement is uncertain. Taking TCP as an example, while 70% witness better performance, 30% requests have longer RTT than those not intercepted. While recalling that redirected requests are mostly handled by local resolvers (previously illustrated in Table 4, that alternative resolvers locate in the same AS as clients), it also shows that little extra time overhead is introduced by the on-path devices to redirect requests. As a result, the hidden interception behavior is hard to be noticed by Internet users.

Specifically for *Request replication*, taking replicated requests to Google DNS as examples, we calculate the difference of *arrival time* between in-band and corresponding out-of-band requests, at authoritative nameservers. In total, out-of-band requests of 14,590 resolutions (84.63%, of 17,239 replicated requests) arrive at

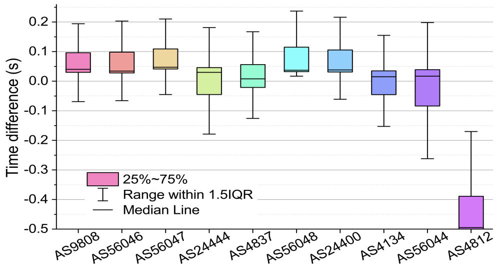


Figure 10: *Arrival time difference of replicated requests. Top 10 ASes with most replicated requests to Google Public DNS is shown. If positive, in-band request from Google arrives slower than the replicated one from alternative resolvers.*

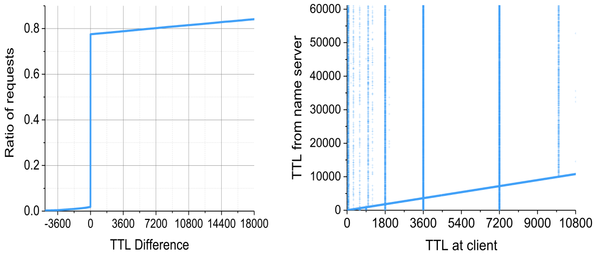
our authoritative nameservers *faster* than in-band ones. Zooming into ASes, Figure 10 presents the top 10 ASes with most replicated requests to Google. While replicated requests from most ASes arrive faster, in AS4812 (China Telecom Group), *all* out-of-band requests lag behind. We suppose that it might be caused by the implementation of network devices in this AS, or the out-of-band requests following different and longer route.

Summary. Through RTT of DNS lookups, we discover that *Request replication* improves the performance of DNS lookups, especially for requests over UDP, making out-of-band responses more likely to be accepted by clients. Observing from authoritative nameservers, 84.63% replicated requests to Google DNS arrive faster. However, *Request redirection* brings uncertain impact according to our findings.

5.3 Manipulation of Responses

By comparing responses generated by our authoritative nameservers and those accepted by clients, we find cases where responses are tampered on the way back. We focus on TTL and DNS record values of a response and elaborate on how they are manipulated.

TTL Value. As illustrated in Section 3.2, TTL values returned by our authoritative nameservers are randomly selected from 1 to 86400. However, for our clients, we find that about 20% of the TTL values are replaced, mostly with a smaller value, as shown in Figure 11(a). By scattering each request onto Figure 11(b), we find that there are preferred values for modified TTL, such as 1800, 3600 and 7200.



(a) ECDF (b) Scatter

Figure 11: *TTL value of DNS responses. (a) presents the ECDF of TTL difference; a positive value suggests TTL at client is smaller than TTL from authoritative nameserver. In (b), each dot represents a single response.*

Table 8: *Classification of tampered DNS responses*

Classification	#	Tampered Responses	Client AS
Gateway	54	`192.168.32.1`	AS4134, CN, ChinaTelecom
		`10.231.240.77`	AS4134, CN, ChinaTelecom
Monetization	10	`39.130.151.30`	AS9808, CN, GD Mobile
		`117.102.104.28`	AS17451, ID, BIZNET
Misconfiguration	26	mx1.norelay.stc.com.sa	AS25019, SA, Saudi NET
		`::218.207.212.91`	AS9808, CN, GD Mobile
Others	54	`fe80::1`	AS4837, CN, ChinaUnicom

DNS record values. Though small in quantity, we do observe cases where clients accept answers with tampered DNS records (including A, AAAA and MX), shown in Table 8. For A and AAAA records which occupy a majority, besides being replaced with private addresses (possibly being traffic gateway), we observe DNS hijacking for *illicit traffic monetization*. As an example, 8 responses from Google Public DNS are tampered in AS9808 (Guangdong Mobile), pointing to a web portal which promotes an APP of China Mobile. The corresponding clients are located in the same AS. For MX records, possibly due to configuration errors, we observe mail servers of a Saudi Arabian ISP show up in the responses to a client in AS25019 (Saudi Telecom Company JSC).

5.4 Motivations of Interception

In this section, we investigate the motivations of DNSIntercept. We first survey the devices, vendors and software platforms that provide DNSIntercept capability, by querying search engines and browsing technical forums. In the end, we find that three well-known router manufacturers (Cisco [4], Panabit [23], and Shenxingzhe [8]), three companies (ZDNS [26], Haomiao [7], Ericsson [21]) and one software platform (DNS traffic redirecting system of Xinfeng [24]) support DNSIntercept. Meanwhile, several detailed technical approaches to intercepting DNS traffic have been published in [7, 21, 23, 24, 26]. As an example, China Mobile proposed an approach, which can replicate out-of-band DNS requests at backbone networks and respond to clients with local DNS resolvers, which is similar to *Request replication*. The above publications mention several possible motivations of DNSIntercept, and we now discuss them based on our measurement results.

Improving DNS security. Vendors claim that through DNSIntercept, DNS requests are handled by *trusted* local DNS servers rather than *untrusted* ones outside the local network, hence are less likely to be hijacked [24, 26]. However, first of all, for clients who trust public DNS services and designate them to handle DNS requests, `DNSIntercept` certainly brings ethical issues and violates the trust relationship between users and their preferred DNS resolvers. Besides, our measurement results show that the interception ratio of public DNS services, which are of good reputation and security deployment, is *significantly higher* than that of less-known public services. This conclusion conflicts with improving DNS security using `DNSIntercept`, since out-of-band public DNS services are not treated equally as untrusted resolvers. What's worse, while rare, we do observe hijacking behaviors for profit (e.g., traffic monetization).

Improving performance of DNS lookups. Another claimed motivation of `DNSIntercept` is to improve the performance of DNS lookups and user experience. As discussed in Section 5.2, we find that *Request replication* does shorten the RTT of DNS lookups, while the influence of *Request redirection* is uncertain. However, in practice, for top 5 ASes shown in Table 7, the ratio of *Request redirection*, which brings uncertain rather than probable improvement of performance, is *significantly higher*. Therefore, `DNSIntercept` only brings limited improvement to DNS lookup performance.

Reducing financial settlement. ISPs, especially those of a small scale, would like to reduce their cost of traffic exchange among networks. *Request redirection* satisfies the need of reducing out-of-band traffic, thus is witnessed in some ASes as shown in Table 7. Therefore, we suppose the financial issue to be a major motivation of `DNSIntercept`. After an offline meeting with the DNS management team of one large Chinese ISP, this motivation is confirmed.

5.5 Summary of Findings

To sum up, we develop the following findings in phase *China-wide analysis*.

- On the whole, DNS packets over `UDP` are preferred for `DNSIntercept`. Taking packets sent to Google Public DNS as examples, **27.9%** `UDP`-based packets are intercepted, the ratio being only 7.3% over `TCP`. Moreover, `A`-type requests have slightly higher interception ratio, while different requested domain names introduce a minor difference.
- Interception behaviors are found in **61** ASes. We find that China Mobile, one of the largest Chinese ISPs, has intercepted significantly more DNS traffic than other ISPs. *Request redirection* is preferred, in order to conduct `DNSIntercept`.
- As for the performance of DNS lookup, in general, *Request replication* shortens the RTT of a DNS request. As for *Request redirection*, an uncertain effect is brought to RTT of DNS requests.
- We speculate the motivations of `DNSIntercept` include *reducing financial settlement* and *improving performance of DNS lookups*, instead of improving DNS security.

6 Threats

With good reputation and availability, well-known public DNS services are widely trusted by Internet users and applications. Unfortunately, our study shows that the trust can be violated by `DNSIntercept`. We further discuss the potential threats and security concerns introduced by `DNSIntercept`.

Ethics and privacy. `DNSIntercept` is difficult to detect at client side, thus Internet users might not realize their traffic is intercepted. Firstly, when DNS requests from clients are handled by alternative resolvers, previous studies have proved it is possible to illegally monetize from traffic [36, 56]. Secondly, as it is difficult for Internet users to detect `DNSIntercept` merely from clients, public DNS resolvers can be wrongly blamed when undesired results (e.g., advertisement sites or even malware) are returned [36]. Finally, it is possible for intercepted DNS requests to be snooped by untrusted third parties, leading to the leak of privacy data. Therefore, we believe `DNSIntercept` potentially brings ethical and privacy risks to Internet users.

DNS security practices. While popular public DNS servers are often deployed with full DNSSEC support and up-to-date DNS software, a number of nameservers and resolvers in the wild are still using outdated or even deprecated DNS software, which may be vulnerable to known attacks [42, 54], and DNSSEC deployment on resolvers is still poor. We provide a cursory view of security practices of 1,166 alternative DNS resolvers that contact our authoritative nameservers; 205 of them are open to the public. Although these resolvers might not be broadly representative, they still provide us with an opportunity to understand DNS security practices. Among the 205 public alternative resolvers, only 88 (43%) *accept DNSSEC requests*; those actually validating DNSSEC requests could be less. After fingerprinting the DNS software deployed on the resolvers using fpdns [11], we find 97 (47%) are running BIND. Unfortunately, the fingerprint shows that *all* 97 servers use versions earlier than 9.4.0, which ought to be deprecated *before 2009*. Therefore, according to the public vulnerability repository [6], all of them are vulnerable to known attacks like DoS.

DNS functionalities. Besides DNSSEC, other functionalities of DNS can be affected by `DNSIntercept`, if alternative resolvers do not provide the related support. An example is EDNS Client Subnet (ECS) request, which allows a DNS query to include the address where it originates, thus different responses can be returned according to the location of clients. However, by checking the 205 alternative resolvers that are open, we find that only 45 (22%) accept ECS requests.

7 Mitigation Discussion

At present, almost all DNS packets are sent unencrypted, which makes them vulnerable to snooping and manipulation. This problem has already been noticed by the DNS community, and RFC7858 [39], which describes the specification of DNS over Transport Layer Security (TLS), is released to address this problem. Unfortunately, the deployment of DNS over TLS is sophisticated and needs changes from the client side. As such, the wide deployment of this initiative could take a long time.

Based on our observation, we developed an online checking tool [25] to help Internet users detect `DNSIntercept`. This tool works with the help of the authoritative nameservers operated by ourselves. A user visiting our checking website will issue a DNS request to our domain, and the request is captured by our authoritative nameserver. By comparing the resolvers that contact our nameservers to their designated ones, Internet users are able to identify `DNSIntercept`. Currently, we are still perfecting this website, aiming at providing more information of `DNSIntercept` for Internet users. However, current solutions and mitigations are far from enough. The security community needs to propose new solutions that can address the issues around `DNSIntercept`.

8 Related Work

Rogue DNS resolvers. Adversaries can build DNS resolvers which return rogue responses for DNS lookups, which can arbitrarily manipulate traffic from users. Previous studies showed that motivations include malware distribution, censorship, and ad injection [38, 42]. In this paper, we study another type of DNS traffic manipulation.

Transparent DNS proxies. Transparent DNS proxies could manipulate DNS traffic that goes through. Firstly, network operators could monetize from through redirecting DNS-lookup error traffic to advertisements [55, 56]. Similarly, Chung *et al.* leveraged the residential proxy network to study violations of end-to-end transparency on local DNS servers, their results showing 4.8% `NXDOMAIN` responses are rewritten with ad server addresses [36]. Furthermore, previous studies presented that 18% DNS sessions of cellular network go through transparent DNS proxies [53] and time-to-live values (TTL) are treated differently [49]. In addition, technical blogs have reported that it is possible for Internet Service Providers to hijack DNS traffic using DNS transparent proxies [1, 13, 17, 18]. By contrast, our study focuses on the on-path hidden interception behavior, instead of rogue resolvers or DNS proxies.

Internet censorship. The DNS protocol lacks authentication and integrity check, hence DNS traffic manipulation has become a prevalent mechanism of censorship, blocking users from accessing certain websites. Significant efforts have been devoted to studying the whats, hows, and whys of censorship in both global and country-specific views. Results showed many countries have deployed DNS censorship capabilities, include China, Pakistan, Egypt, Iran and Syria [28, 29, 30, 31, 32, 44, 45, 58]. Also, from a global view, Pearce *et al.* discovered widespread DNS manipulation [48], and Scott *et al.* found DNS hijacking in 117 countries [50]. By contrast, the domain names used in our study are exclusively registered and used, and we avoid any sensitive keyword. Therefore, our study does not overlap with censorship mechanism.

Other manipulation of Internet resources. Moreover, researches have discovered other ways to manipulate DNS traffic, including abusing the DNS namespace (i.e., "Name Collision" [34, 35]), exploiting configuration errors and hardware issues (typosquatting [47] and bit-squatting [54]), and "Ghost domains" [40]. As the closest work to ours, Allman *et al.* presented how to detect unauthorized DNS root servers [27]. However, only one type of traffic manipulation was considered, with only limited cases being discovered. Our study serves as a complement to these existing works in understanding the security issues in DNS ecosystem.

Compared to previous researches, our work gives a systematic and large-scale research on `DNSIntercept`, a class of DNS behavior that has not yet been well-studied, and highlights issues around security, privacy, and performance.

9 Conclusions

In this paper, we present a large-scale study on `DNSIntercept`, which brings to light security, privacy and performance issues around it. We develop a suite of techniques to detect this kind of hidden behavior, leveraging two unique platforms with numerous vantage points. Based on our dataset, we find that `DNSIntercept` exists in some ASes and networks. In addition, interception characteristics as well as motivations of `DNSIntercept` are further analyzed. Our results indicate that the hidden `DNSIntercept` can potentially introduce new threat in the DNS eco-system, and new solutions are needed to address the threat.

Acknowledgments

We thank our talented team member, Zihao Jin, for his valuable work on data processing. We thank professor Vern Paxson for his detailed guidelines and insightful comments. We thank Xiaofeng Wang, Sumayah Alrwais, Sadia Afroz, Michael C Tschantz and Xianghang Mi for their valuable discussion. We thank Fengpei Li, Zaifeng Zhang, Jinjin Liang, Zhou Pang, Jianjun Chen, Yiming Zhang and Jia Zhang for their feedback and help. We also thank our shepherd Nick Nikiforakis and all anonymous reviewers for their helpful suggestions and comments to improve the paper.

This work was supported by the National Key Basic Research Program (grant 2017YFB0803202), the National Natural Science Foundation of China (grant 61772307, 61472215, U1636204), and CERNET Innovation Project NGII20160403. The Fudan author is supported in part by the National Natural Science Foundation of China (grant U1636204), the National Program on Key Basic Research (grant 2015CB358800).

Any views, opinions, findings, recommendations, or conclusions contained or expressed herein are those of the authors, and do not necessarily reflect the position, official policies or endorsements, either expressed or implied, of the Government of China or Qihoo 360.

References

[1] 22 networks with transparent dns proxies. https://help.dnsfilter.com/article/22-networks-with-transparent-dns-proxies.

[2] As rank: A ranking of the largest autonomous systems (as) in the internet. http://as-rank.caida.org.

[3] Avast secure dns. https://help.avast.com/en/av_abs/10/etc_tools_secure_dns_overview.html.

[4] Cisco: Dns configuration guide. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_dns/configuration/12-4t/dns-12-4t-book/dns-config-dns.html.

[5] Cisco umbrella intelligent prox. https://learn-umbrella.cisco.com/feature-briefs/intelligent-proxy.

[6] Cve-2015-5477: An error in handling tkey queries can cause named to exit with a require assertion failure. https://nvd.nist.gov/vuln/detail/CVE-2015-5477.

[7] Dns traffic clear refreshment system. http://www.xpspeed.net/product4.html.

[8] Dns traffic router of shenxingzhe. http://bbs.dwcache.com/t/31.

[9] Dynamic dns. https://dyn.com/dns/.

[10] Farsight passive dns. https://www.farsightsecurity.com/solutions/dnsdb.

[11] fpdns https://github.com/kirei/fpdns.

[12] Google public dns. https://dns.google.com/.

[13] How to find out if your internet service provider is doing transparent dns proxy. https://www.cactusvpn.com/tutorials/how-to-find-out-if-your-internet-service-provider-is-doing-transparent-dns-proxy/.

[14] Http and socks proxies. https://www.proxyrack.com/.

[15] Introduction of dns pai project. http://www.dnspai.com.

[16] Is your isp hijacking your dns traffic? https://labs.ripe.net/Members/babak_farrokhi/is-your-isp-hijacking-your-dns-traffic.

[17] Is your isp hijacking your dns traffic. https://labs.ripe.net/Members/babak_farrokhi/is-your-isp-hijacking-your-dns-traffic.

[18] Isp doing transparent dns proxy. https://www.smartydns.com/support/isp-doing-transparent-dns-proxy/.

[19] Luminati: Residental proxy service for businesses. https://luminati.io.

[20] Maxmind: Ip geolocation. https://www.maxmind.com/en/home.

[21] A method to conduct dns traffic redirecting in telecommunication system. https://patentimages.storage.googleapis.com/cc/b2/65/6272013c07765e/CN103181146A.pdf.

[22] Open dns. https://www.opendns.com/.

[23] Panabit intelligent dns system. http://www.panabit.com/html/solution/trade/service/2014/1216/94.html.

[24] The practice of dns control based on out-of-band responder mechanism. http://www.ttm.com.cn/article/2016/1000-1247/1000-1247-1-1-00064.shtml.

[25] What is my dns resolver? http://whatismydnsresolver.com.

[26] Zdns: solutions for campus network services. http://free.eol.cn/edu_net/edudown/2017luntan/zdns.pdf.

[27] ALLMAN, M. Detecting dns root manipulation. In *Passive and Active Measurement: 17th International Conference, PAM 2016, Heraklion, Greece, March 31-April 1, 2016. Proceedings* (2016), vol. 9631, Springer, p. 276.

[28] ANONYMOUS. The collateral damage of internet censorship by dns injection. *ACM SIGCOMM CCR 42*, 3 (2012).

[29] ANONYMOUS. Towards a comprehensive picture of the great firewalls dns censorship. In *FOCI* (2014).

[30] ARYAN, S., ARYAN, H., AND HALDERMAN, J. A. Internet censorship in iran: A first look. In *FOCI* (2013).

[31] BAILEY, M., AND LABOVITZ, C. Censorship and co-option of the internet infrastructure. *Ann Arbor 1001* (2011), 48104.

[32] CHAABANE, A., CHEN, T., CUNCHE, M., DE CRISTOFARO, E., FRIEDMAN, A., AND KAAFAR, M. A. Censorship in the wild: Analyzing internet filtering in syria. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (2014), ACM, pp. 285–298.

[33] CHEN, J., JIANG, J., DUAN, H., WEAVER, N., WAN, T., AND PAXSON, V. Host of troubles: Multiple host ambiguities in http implementations. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), ACM, pp. 1516–1527.

[34] CHEN, Q. A., OSTERWEIL, E., THOMAS, M., AND MAO, Z. M. Mitm attack by name collision: Cause analysis and vulnerability assessment in the new gtld era. In *Security and Privacy (SP), 2016 IEEE Symposium on* (2016), IEEE, pp. 675–690.

[35] Chen, Q. A., Thomas, M., Osterweil, E., Cao, Y., You, J., and Mao, Z. M. Client-side name collision vulnerability in the new gtld era: A systematic study. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), ACM, pp. 941–956.

[36] Chung, T., Choffnes, D., and Mislove, A. Tunneling for transparency: A large-scale analysis of end-to-end violations in the internet. In *Proceedings of the 2016 ACM on Internet Measurement Conference* (2016), ACM, pp. 199–213.

[37] Chung, T., van Rijswijk-Deij, R., Chandrasekaran, B., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., and Wilson, C. A longitudinal, end-to-end view of the dnssec ecosystem. In *USENIX Security* (2017).

[38] Dagon, D., Provos, N., Lee, C. P., and Lee, W. Corrupted dns resolution paths: The rise of a malicious resolution authority. In *NDSS* (2008).

[39] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and Hoffman, P. Specification for dns over transport layer security (tls). Tech. rep., 2016.

[40] Jiang, J., Liang, J., Li, K., Li, J., Duan, H., and Wu, J. Ghost domain names: Revoked yet still resolvable. In *Network and Distributed System Security Symposium* (2012).

[41] Kreibich, C., Weaver, N., Nechaev, B., and Paxson, V. Netalyzr: illuminating the edge network. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement* (2010), ACM, pp. 246–259.

[42] Kührer, M., Hupperich, T., Bushart, J., Rossow, C., and Holz, T. Going wild: Large-scale classification of open dns resolvers. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference* (2015), ACM, pp. 355–368.

[43] Liu, D., Hao, S., and Wang, H. All your dns records point to us: Understanding the security threats of dangling dns records. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), ACM, pp. 1414–1425.

[44] Lowe, G., Winters, P., and Marcus, M. L. The great dns wall of china. *MS, New York University 21* (2007).

[45] Nabi, Z. The anatomy of web censorship in pakistan. In *FOCI* (2013).

[46] Nakibly, G., Schcolnik, J., and Rubin, Y. Website-targeted false content injection by network operators. In *USENIX Security Symposium* (2016), pp. 227–244.

[47] Nikiforakis, N., Van Acker, S., Meert, W., Desmet, L., Piessens, F., and Joosen, W. Bitsquatting: Exploiting bit-flips for fun, or profit? In *WWW, 2013*.

[48] Pearce, P., Jones, B., Li, F., Ensafi, R., Feamster, N., Weaver, N., and Paxson, V. Global measurement of dns manipulation. In *26th USENIX Security Symposium* (2017), USENIX Association.

[49] Schomp, K., Callahan, T., Rabinovich, M., and Allman, M. On measuring the client-side dns infrastructure. In *Proceedings of the 2013 conference on Internet measurement conference* (2013), ACM, pp. 77–90.

[50] Scott, W., Anderson, T. E., Kohno, T., and Krishnamurthy, A. Satellite: Joint analysis of cdns and network-level interference. In *USENIX Annual Technical Conference* (2016), pp. 195–208.

[51] Shi, X., Xiang, Y., Wang, Z., Yin, X., and Wu, J. Detecting prefix hijackings in the internet with argus. In *Proceedings of the 2012 ACM conference on Internet measurement conference* (2012), ACM, pp. 15–28.

[52] Tyson, G., Huang, S., Cuadrado, F., Castro, I., Perta, V. C., Sathiaseelan, A., and Uhlig, S. Exploring http header manipulation in-the-wild. In *Proceedings of the 26th International Conference on World Wide Web* (2017), International World Wide Web Conferences Steering Committee, pp. 451–458.

[53] Vallina-Rodriguez, N., Sundaresan, S., Kreibich, C., Weaver, N., and Paxson, V. Beyond the radio: Illuminating the higher layers of mobile networks. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (2015), ACM, pp. 375–387.

[54] Vissers, T., Barron, T., Van Goethem, T., Joosen, W., and Nikiforakis, N. The wolf of name street: Hijacking domains through their nameservers. In *Proceedings of the 2017 ACM SIGSAC Con ference on Computer and Communications Security* (2017), ACM, pp. 957–970.

[55] Weaver, N., Kreibich, C., Nechaev, B., and Paxson, V. Implications of netalyzrs dns measurements. In *Proceedings of the First Workshop on Securing and Trusting Internet Names (SATIN), Teddington, United Kingdom* (2011).

[56] Weaver, N., Kreibich, C., and Paxson, V. Redirecting dns for ads and profit. In *FOCI* (2011).

[57] Wikipedia. Public recursive name server. https://en.wikipedia.org/wiki/Public_recursive_name_server.

[58] Xu, X., Mao, Z. M., and Halderman, J. A. Internet censorship in china: Where does the filtering occur? In *International Conference on Passive and Active Network Measurement* (2011), Springer, pp. 133–142.