Digital Object Identifier: 10.1109/ACCESS.2014.2323299

URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6814824

Measurement of Globally Visible DNS Injection

Matthäus Wander, Christopher Boelmann, Lorenz Schwittmann, Torben Weis

***Abstract*—DNS injection is a censorship method for blocking access to blacklisted domain names. The method uses deep packet inspection on all DNS queries passing through the network and injects spoofed responses. Compared to other blocking mechanisms, DNS injection impacts uninvolved third-parties if their traffic is routed through a censored network. In this paper, we look for large deployments of DNS injection, measured from vantage points outside of the censored networks. DNS injection is known to be used in China since it leaked unintentionally into foreign networks. We find that DNS injection is also used in Iran and can be observed by sending DNS queries to Iranian networks. In mid 2013 the Iranian DNS filter was temporarily suspended for some names, which correlated with media coverage of political debates in Iran about blocking social media. Spoofed responses from China and Iran can be detected passively by the IP address returned. We propose an algorithm to obtain these addresses remotely. After testing 255,002 open resolvers outside of China, we determined that 6% are potentially affected by Chinese DNS injection when querying top-level domains outside of China. This is essentially the result of one top-level domain name server for which an anycast instance is hosted in China.**

***Index Terms*—Domain Name System, Internet, data security.**

I. Introduction

The Domain Name System (DNS) is a common target to facilitate Internet censorship. The Internet filter of the People's Republic of China, colloquially known as Great Firewall of China (GFW), returns bogus DNS responses for the purpose of blocking websites since at least 2002 [1]. DNS spoofing is used in conjunction with other filtering methods, e.g. inspection of HTTP traffic [2]. Lowe discovered in 2007 that DNS spoofing in China occurs on router-level, i.e. spoofed responses originate from intermediate hops on the path to the actual IP destination [3]. The principle of this man-in-the-middle attack is shown in Fig. 1. The spoofed response contains an IP address that diverts the user application to the wrong server or to an unreachable destination. The attacker could also spoof a negative response with a name resolution or server error instead to prevent access to the server.

This type of DNS spoofing was later coined as DNS injection [4] and is different from blocking domain names on recursive resolvers. Recursive resolvers are DNS servers, typically at the Internet Service Provider (ISP) premises, which handle the domain name resolution functionality for end hosts. An example for DNS filtering on recursive resolvers is the blocking of Twitter and Youtube in Turkey in March 2014. The blocking quickly turned out to be ineffective when Turkish Internet users started to bypass their ISP resolvers by using public resolvers like Google Public DNS and OpenDNS. Blocking domain names on recursive resolvers affects only the users of the resolver, whereas DNS injection affects all users whose traffic is routed through the censored network. DNS injection was brought to the attention of the DNS community by Ereche in 2010 when queries to an anycast instance of the I-root name server resulted in bogus responses [5].

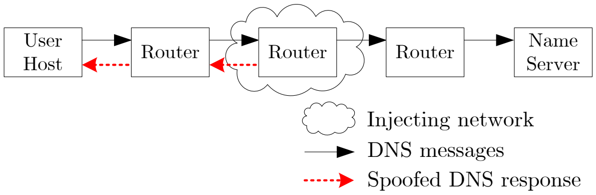

Fig. 1. Injection of spoofed DNS response.

The purpose of this paper is to find deployments of DNS injection and to understand their behavior and effect that they can have on third-parties. The findings are summarized and compared with previous studies in Section II. Section III describes an Internet-wide measurement to find deployments of DNS injection. Section IV contains an analysis of the granularity of the domain blacklist. In Section V, we propose an algorithm to obtain the list of bogus addresses returned by the DNS injection filters, which can be used to detect spoofed responses. Section VI contains an impact assessment of Chinese DNS injection on third-parties in other countries.

II. Findings

After probing a representative set of 14,479,104 public IPv4 addresses, 960,078 addresses (6.6%) indicated DNS injection in China and 11,414 (0.08%) in Iran. We did not observe evidence for large-scale DNS injection in other countries from our vantage points in AS680, AS24940 and AS24961. This may be due to our choice of domain names selected for probing or because different filtering methods are used, which are not observable outside of the affected networks.

A. DNS Injection in China

DNS injection is part of the Great Firewall of China. We found 404 domain names to be blocked by DNS injection but this is supposedly only a small part of the whole blacklist. Some domain names are blocked by the filter even when combined with any prefix or suffix. This will e.g. in case of filtering `facebook.com` result in an overblocking of `iamnotonfacebook.com`. Other than the query name, responses are spoofed regardless of the query content, e.g. querying for a SOA or TXT record type will trigger a futile spoofed response with an A resource record. Original DNS queries are not taken off the network which typically yields in multiple spoofed responses. It is not necessary for a successful

Distributed Systems Group, University of Duisburg-Essen, 47048 Duisburg, Germany. Email: matthaeus.wander@uni-due.de.

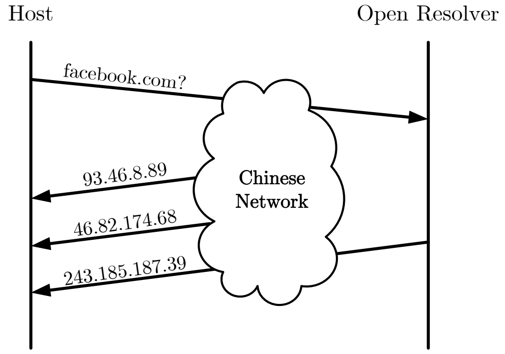

Fig. 2. Example DNS query into a Chinese network.

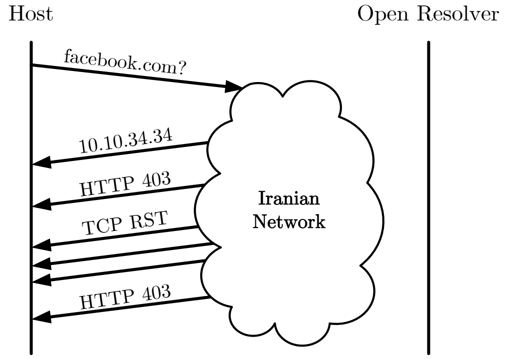

Fig. 3. Example DNS query into an Iranian network.

spoofing attack to suppress DNS messages. As the client host is topologically closer to the injecting device than to the destination name server, injected responses have a head start over genuine ones.

An example is shown in Fig. 2: a DNS query is replied with two injected responses and one response from the open resolver. The response from the resolver is not altered by the network. However, it is also a bogus response because the resolver itself received and cached a bogus response.

Bogus answer addresses for blocked domain names are returned randomly from a fixed set of IP addresses. An explanation for this behavior is that DNS injection may be used in conjunction with IP filtering: blacklisted domain names resolve to IP addresses that are filtered on border routers [1]. The IP address sets differ depending on the blacklisted domain name and can be gathered by repeatedly sending DNS queries to Chinese networks. Overall, we collected 33 bogus IP addresses, some of them returned for many different domain names. We asked a network operator who happens to own one of the IP addresses frequently returned by Chinese DNS injection whether there is any suspicious network traffic visible. They explained that the IP address is unused and traffic of about 150 packets per second is dropped at the network borders. This is significantly more than what is expected from Internet background radiation [6] and may be an indication that hosts affected by Chinese DNS injection attempt to connect to the IP address without success. A geolocation analysis of a short (not representative) packet trace shows packets originating from 51 countries and regions, including the U.S., Hong Kong, Pakistan and Taiwan.

Measurement of the query paths between 255,002 open resolvers outside of China and 1144 root and TLD name servers outside of China shows that 15,225 (6%) open resolvers are affected by DNS injection. The majority was only affected on a single path: `e.dns.kr` which is authoritative for `.kr` and the internationalized variant `.xn--3e0b707e`. One of the anycast instances of `e.dns.kr` is hosted in Beijing, China [7]. There is no evidence for DNS traffic with destinations outside of China to be affected by DNS injection on a larger scale. However, this may be specific to the root and TLD name servers. DNS traffic to second-level domain name servers may nevertheless transit through China and be affected by DNS injection.

B. DNS Injection in Iran

We found DNS injection to be used in Iran for blocking access to 14 domain names, including `facebook.com`, `plus.google.com`, `twitter.com` and `youtube.com`. The filtering of relatively few DNS names is in line with research by Halderman on Internet censorship from *inside* of an Iranian network [8]: the main filtering method in Iran is by blocking HTTP traffic. Characteristic for Iranian DNS injection is the use of the bogus answer IP address 10.10.34.34 and spoofing of TCP packets. Fig. 3 shows how a UDP-based DNS query provokes a bogus UDP-based DNS response, two TCP segments with HTTP error 403 (‘*forbidden*’) and FIN/ACK bits set and three TCP RST packets. Note that we did not send any TCP traffic to the Iranian open resolver, thus the TCP sequence and acknowledgement numbers do not match any existing connection.

Different from Chinese DNS injection, the Iranian DNS filter strictly expects certain query flags. Queries with e.g. the *authenticated data* (AD) bit set and the *recursion desired* (RD) bit clear will pass through the DNS filter without triggering a spoofed response. If the filter catches a query, it will not only inject a spoofed DNS response but also drop the query.

During our measurements we observed that the number of injected responses from Iranian networks decreased for `facebook.com` and `twitter.com` in mid 2013. This coincides with a report from The Guardian in July 2013 about the newly elected President of Iran Hassan Rouhani who spoke out to loosen filtering of social media [9]. However, a few weeks later the numbers were as before, indicating that the social media filters have been reinstalled again in September/October 2013.

C. Comparison with Earlier Studies

In a 2007 study Lowe et al. probed 1607 open resolvers in China whether their responses differ from resolvers in the U.S. [3]. Almost all Chinese resolvers consistently returned bogus responses for a list of 393 apparently censored domain names. The bogus responses referred to a set of 21 distinct answer IP addresses, which is a subset of the 33 IP addresses that we collected in our measurement.

An anonymous study in 2011/2012 found significant occurrence of Chinese DNS injection for .de and .kr but not for the root servers [4]. Their measurement method was to query 43,842 open resolvers outside of China for `domain.xy.RANDOM.tld`, where `domain.xy` is a blacklisted name, `RANDOM` a random string and `tld` the top-level domain under test. With this method, it is not possible to determine which name server of the tested TLD the query is sent to. To increase the likelihood of testing all name servers of a TLD, the test was repeated 200 times with different random strings, requiring 62,400 queries per resolver to test all 312 TLDs at that time. The method we used in our measurement is not per domain but per name server. As we can control which name server is being queried, 1155 queries per resolver suffice to test 317 TLDs and to identify which of the name servers is being affected. Thus, we can confirm that .kr was still affected in 2013 but only for one name server for which an anycast instance is located in China. The .de name servers were no longer affected, including `a.nic.de` which still comprises an anycast instance in Beijing.

Similar was found by Koch for .de in 2012 by utilizing 1762 RIPE Atlas measurement probes [10]: 339 probes (19%) were routed to an anycast instance in China, out of which 218 (12%) were affected by DNS injection. There was no indication of tampering for anycast locations of .de other than Beijing.

III. Probing for DNS Injectors

We actively probed the public IPv4 address space to identify networks that employ globally visible DNS injection. The method used in this section is to send DNS queries to a large amount of destinations and to determine whether the responses have been tampered with (Fig. 4). The destination hosts do not need to run a name server and even do not need to be online. As DNS injection works on router-level, we will receive a response if the DNS query is routed into or through a network that spoofs bogus DNS responses. Without DNS injection, we do not expect a DNS response. However, there is also a chance that the DNS query reaches a responsive name server or an open resolver. We thus need to identify whether a response is genuine or spoofed.

The measurement consists of two parts: first, a sparse probing to find filtered domain names, and second, a dense probing to analyze the extent of DNS injection. The rationale of this approach is to avoid unnecessary network load. For the second part, we sent DNS requests into every IPv4 /24 subnet for domain names, which showed evidence of DNS injection in the first part. We refrained from probing all IPv4 addresses because we do not expect substantially different results than from probing one IPv4 address from each /24 subnet.

A. Finding Filtered Domain Names

We composed a list of websites from a wide range of categories that could be censored, including file sharing, information freedom, human rights, online gambling, sexual content, controversial religious or political content, graphic violence and social media. For each of the 47 candidate domain names, we sent a DNS query from our vantage point in AS680 (DFN, Germany) to 422,228 IPv4 addresses, each in a different publicly announced BGP prefix.[1] This lead to 682,640 responses which comprised 9958 distinct answer IP addresses. After manually removing genuine, unaltered responses, we investigated the most frequent remaining addresses which appeared bogus. Part of them originate from censoring open resolvers which our probe queries hit by coincidence. Thus as a byproduct of our measurements, we confirm systematic censorship on ISP resolvers being used in Bulgaria, Colombia, Denmark, Indonesia, Singapore and Turkey. In these cases, the bogus answer IP address points to an ISP webserver with a notice that the website the user tried to access is blocked. The remaining addresses are an evidence of censorship by DNS injection, which is further analyzed in the following section.

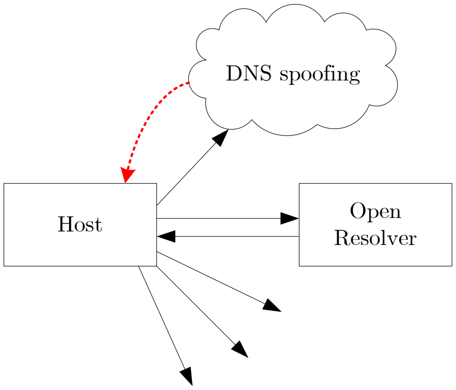

Fig. 4. Measurement for finding injecting networks.

B. Analyzing Extent of DNS Injection

To analyze the extent of DNS injection, we sent DNS queries from AS24961 (myLoc, Germany) to each IPv4 /24 subnet, except for subnets within globally unroutable networks like 10/8. To spread the load per destination network over time, we iterate through the IPv4 address space with an offset (1.0.0.99, 2.0.0.99, 3.0.0.99, etc.). It follows an analysis of measurement results for `facebook.com`. 14,479,104 DNS queries were sent and 1,960,297 responses received, out of which 99.0% were free from errors and included an A resource record with an IPv4 address.

At the time of this writing, the authoritative name servers return one public IPv4 address for `facebook.com` with no indication of DNS-based load balancing. However, the responses in our measurement data contain 284 distinct answer IPv4 addresses. 273 of them occurred only a few times (in total 559 times) and are of no further interest. The remaining eleven answer IPv4 addresses are shown in Table I. The only

[1] BGP routing table from APNIC [11].

TABLE I
ANSWER IP ADDRESSES FOR `facebook.com`.

Count	Answer IP address	CC	AS#	Organization
11,418	10.10.34.34	–	–	–
97,936	173.252.110.27	US	32934	Facebook
202,688	46.82.174.68	DE	3320	Deutsche Telekom
202,930	93.46.8.89	IT	12874	Fastweb
203,132	203.98.7.65	NZ	4768	TelstraClear
203,165	78.16.49.15	IE	2110	BT Ireland
203,257	8.7.198.45	US	3356	Level 3
203,751	59.24.3.173	KR	4766	Korea Telecom
203,756	37.61.54.158	AZ	28787	Baktelekom
203,795	243.185.187.39	–	–	–
204,163	159.106.121.75	US	–	US DoD NIC

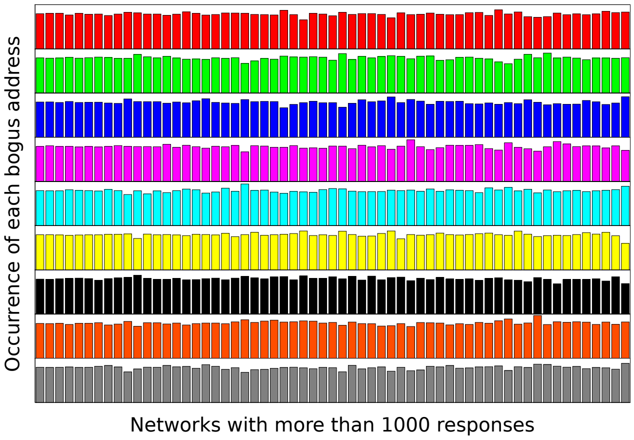

Fig. 5. Uniform distribution of bogus IP addresses in spoofed responses from Chinese networks.

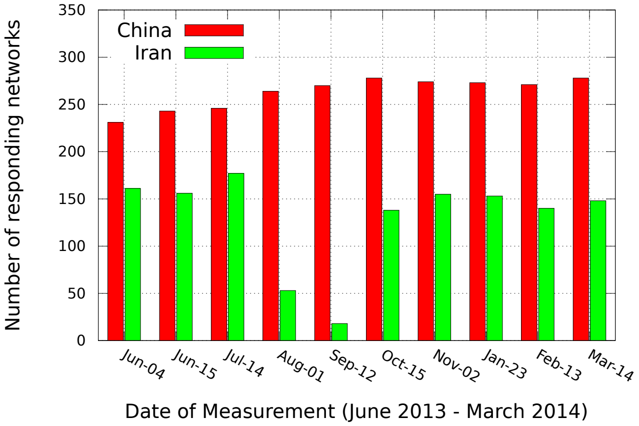

Fig. 6. Injected responses for `facebook.com` over time.

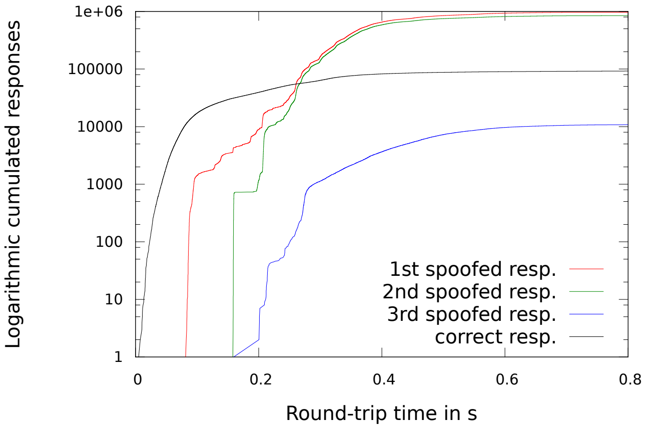

Fig. 7. Correct and spoofed responses for `facebook.com` by latency.

authentic answer IP address can be easily identified as it belongs to AS32934 (Facebook). Note that the number of bogus responses is larger than genuine ones due to the nature of DNS injection: to receive a genuine response the probe query needs to hit an open resolver whereas spoofed responses are always injected, even when the destination host is offline.

There is a remarkably large occurrence of nine answer IP addresses with an almost equal distribution of around 200k responses. A geolocation analysis[2] reveals that 99.9% of the queried IP addresses returning one of these bogus answers are located in mainland China. This indicates that Chinese DNS injection filters return a random address for `facebook.com` from a set of nine fixed IPv4 addresses. The network owners of the bogus addresses do not seem to be related to the network owners who send the spoofed DNS responses. None of the nine addresses host a publicly reachable webserver, and two addresses (159.106.121.75 and 243.185.187.39) are not even routable in the default-free zone. Fig. 5 shows the distribution of the nine bogus addresses, grouped by large AS networks for which we received more than 1000 spoofed responses. The uniform distribution indicates that the selection of the bogus answer address in the DNS response does not depend on the destination address that the query has been sent to.

Another remarkable answer IP address is 10.10.34.34, a private address as per RFC 1918 [13]. 99.9% of the responders of this answer IP address are located in Iran. According to Anderson the address 10.10.34.34 is used by Iranian ISPs to display a webpage with filter notice [14]. When we repeated the probing measurement at different dates, we observed a significant drop of spoofed Iranian responses for a certain period while the Chinese results remained stable. Fig. 6 shows the number of queried AS networks for which we received spoofed responses. The Guardian reported on July 2, 2013 that the newly elected President of Iran Hassan Rouhani spoke out to loosen filtering of social media like Facebook [9]. Our measurements reflect that the DNS filter for `facebook.com` has been lifted by Iranian networks shortly after that. However, a few weeks later the DNS injection filter was active again. On October 7 the Iranian Ministry of Communications and Information Technology announced that removing filters for social media is under consideration and the public will be notified once a decision has been made [15]. As of March 2014, the filter was still in place for `facebook.com` according to our measurements. We interprete the temporary suspension of the DNS filter for Facebook in Iran as an effect of political uncertainty about whether social media should be blocked or not.

A large fraction of the 1.9M responses are duplicate responses which claim to originate from 1,087,945 distinct IPv4

[2] Using GeoLite from MaxMind for geolocation and AS data [12].

addresses. This is common for Chinese DNS injection from which we typically received two spoofed responses. Fig. 7 shows the round-trip times of correct and spoofed responses (y-scale is logarithmic and cumulative). Correct responses originate from open resolvers which are found worldwide. The latency increases (log-)linearly because there is no particular accumulation of latencies to be observed from our vantage point. On the contrary, the latency of spoofed responses increases stepwise, indicating a different injecting network with a typical latency band for each step.

Probing other domains like `twitter.com` (filtered in China and Iran), `www.minghui.org` and `www.strongvpn.com` (filtered in China) shows similar results as above for `facebook.com`. The temporary decrease of spoofed Iranian responses could be observed for `twitter.com` as well. From Chinese DNS filters, we received up to 300k spoofed responses with invalid UDP checksum for some domain names. Considering these faulty responses in the overall statistics, each filtered domain name caused the same amount of spoofed responses. There was no evidence for inconsistent domain blacklists among filtering Chinese networks. The set of bogus addresses returned varies depending on the domain name, which is discussed further in Section V.

IV. Blacklist Testing

Another measurement was performed to determine domain names blacklisted in China and Iran and the granularity of the corresponding domain name filters. The method used here is to send queries for different candidate domain names into Chinese and Iranian networks known to spoof DNS responses from the previous section. We extracted the candidate domain names from the Alexa list of the 1,000,000 most visited websites [16]. The list contains actually less than 1M domain names because some sites are distinguished by URL path but use the same domain name. Combined with a few names from other sources this led to 999,935 unique candidate domain names.

For each domain name `$NAME`, we tested five variants with a prefix or a suffix, as shown in Table II. `$RANDOM` is an alphanumeric string, which we verified to be not blacklisted. To minimize the effect of packet loss each variant was queried 10 times with different Chinese IP addresses and 10 times with Iranian IP addresses. For each queried destination address, we ensured that it does respond to the blacklisted name `facebook.com` but does not run an open resolver.

The measurement resulted in 1024 domains for China and 14 domains for Iran that are blacklisted in at least one of the five requested domain name variants. Some of these blocked domains are the result of overblocking, where e.g. `user1.appspot.com` and `user2.appspot.com` are blocked because `$RANDOM.appspot.com` is blacklisted. After manual testing of the common suffixes, we reduced the results to 404 domain names blacklisted in China and 14 domain names blacklisted in Iran (Table II). The results for China indicate that blocked domain names are in most cases also blocked when prepended with `www.` or a random subdomain. 21 domain names were blocked only if prepended with `www.` but not without it, e.g. `www.nytimes.com`. Furthermore, about half of the names are blocked if random prefixes are prepended without a separating dot, which results e.g. in case of `facebook.com` in an overblocking of `iamnotonfacebook.com`. The Iranian DNS blacklist is less extensive, in terms of total blacklisted names and the use of prefix or suffix wildcards.

TABLE II
Granularity of blacklist.

Query name	China	Iran
1. `$NAME`	383	14
2. `www.$NAME`	384	14
3. `$RANDOM.$NAME`	368	1
4. `$RANDOM$NAME`	167	1
5. `$NAME.$RANDOM`	146	1
Total	404	14

Overall, the number of blacklisted domain names that we have found is not as large as expected. One possible reason may be the use of DNS injection as supplemental blocking method, whereas the primary blocking is via a keyword-based deep packet inspection of HTTP traffic. Maintaining a blacklist of keywords for blocking unwanted content would be more effective than maintaining a blacklist of domain names. Another reason may be the list of popular domains worldwide, which may be missing blocked web sites with a regional audience.

V. Obtaining Bogus Addresses

As shown in Section III, the DNS injection filters deployed return DNS responses with distinct IP addresses in the A resource record. With the list of bogus IP addresses it is easy to detect DNS responses spoofed by DNS injection. Iranian networks return always the same bogus IP address but Chinese networks return an address randomly chosen from a predefined set of bogus IP addresses. In this section, we present an efficient method to obtain the set of bogus IP addresses for one domain name.

After sending DNS queries repeatedly to Chinese networks behind the GFW, we collect n bogus IP addresses from r spoofed responses. The challenge is to estimate whether we have obtained the complete set of bogus addresses. Assuming that the whole set consisted of $n + 1$ IP addresses, the probability p to miss one of the IP addresses in one response would be

$$p = 1 - \frac{1}{n+1}$$

and p^r in r responses. If the whole set consisted of $n + 2$ or more IP addresses, then the probability to miss one of them would be even lower. We continue sending DNS queries until the probability to have missed a bogus IP address is below a predefined threshold t. The minimum amount of responses r required to reach the threshold t is thus:

$$r = \log_p(t)$$

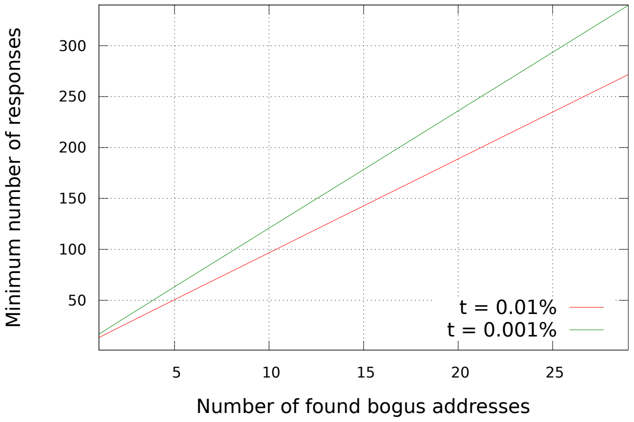

Fig. 8. Number of required responses for given thresholds.

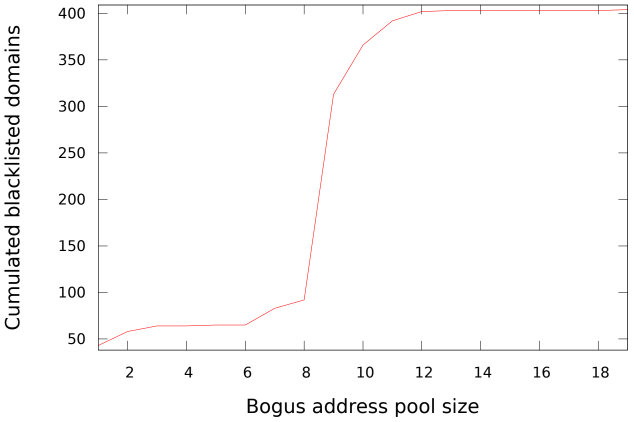

Fig. 9. Number of bogus addresses per blacklisted domain name.

We stop sending further queries once r responses have been received because the set of obtained IP addresses is complete at that point with a probability of $1 - t$.

A. Analysis

The minimum number of required responses r increases with each additional bogus IP address discovered. In Fig. 8 two examples are shown for probability thresholds $t = 0.01\%$ and $t = 0.001\%$. The y-axis shows the minimum number of responses r for a given number of known bogus addresses n on the x-axis.

The underlying assumption in the calculation is a uniform distribution of bogus addresses in spoofed responses. As shown in Table I, this is true in case of `facebook.com` but the distribution is unknown for other blocked domain names. We now discuss the impact when the underlying assumption is not met for an example with a heavily askew distribution. We determined the actual distribution of bogus addresses for `twitter.com` with the large-scale probing method from Section III. The probing yielded a cluster of eight equally distributed addresses, each occurring on average 76,495 times (±185), and one address occurring 1,011,034 times. To calculate the worst case probability to miss one of these IP addresses with the above algorithm, we assume to have obtained eight bogus addresses in total and missed the least frequent one with 76,306 occurrences. For an anticipated $t = 0.001\%$ and with $n = 8$, the algorithm will send queries until $r = 98$ spoofed responses have been received. However, t is only valid for uniform address distributions. The actual probability to miss the least frequent address in this example is:

$$t' = \left(1 - \frac{76,306}{1,622,998}\right)^{98} \approx 0.89\%$$

While the actual failure probability is significantly larger than $t = 0.001\%$, it is a decent probability given that only 98 DNS messages needed to be sent. Determining the address distribution for blacklisted domain names is a high cost operation requiring significantly more than 98 DNS messages per domain name. As we have observed different address distributions for `twitter.com` in earlier measurements, it is also a volatile information. It is thus a reasonable heuristic to assume a uniform address distribution and to apply a safety margin to the threshold probability t to account for deviations.

B. Measurement Result

We applied the above measurement method with $t = 0.01\%$ to 404 domain names blocked in China. To disperse the network load, each query is sent to a different destination IP address. The list of destinations originates from the probing measurement in Section III-B and the list of domain names from the blacklist measurement in Section IV. If the query times out e.g. due to random packet loss, it is resent to another IP address. As we are solely interested in DNS spoofing from routers, we need to ensure that we do not process responses from open resolvers. For each responding address, we thus send five DNS queries for domain names known to be not blocked. If we get a response, then there is an open resolver and the results will be void for this destination address.

The number of bogus IP addresses per domain name ranges from 1 address after 14 responses to 19 addresses after 180 responses. As shown in Fig. 9, most domain names return a set of 9 bogus addresses. While a few bogus addresses are returned exclusively for one domain name, most are used for several blocked names. The most commonly returned addresses are the nine bogus addresses that are also used for `facebook.com` (Table I). For example, 59.24.3.173 was returned for 257 domains. For one domain name the GFW returns a CNAME record (alias name) instead of an IP address, whereas the CNAME target does not seem to be blocked. In total, we obtained a set of 33 bogus IPv4 addresses, which can be used to detect spoofed responses from DNS injection. An example usage scenario would be a web browser extension checking the resolved names against the list of well-known bogus addresses. With this method, it is possible to find further names of the domain blacklist if the names are returning one of the known bogus addresses.

VI. Impact on Resolvers

In this section we analyze the impact of DNS injecting networks on unrelated resolvers from other networks. The basic idea is to query open resolvers worldwide for blacklisted domain names and check if the response is poisoned by DNS injection. We use a measurement method which utilizes a technique from the King tool [17] to send DNS queries between the open resolver and an arbitrary destination address. By using all authoritative name servers of a domain as destination addresses, we can exhaust all query paths between a resolver and a domain under test.

A. Method

The objective is to send a DNS query between a resolver and an IPv4 destination address, e.g. 192.0.2.1. We set up a domain name—here referred to as `example.net`—with delegations in the form of:

```
   exp1  IN   NS  ns.exp1.example.net.
ns.exp1  IN    A  192.0.2.1
ns.exp1  IN AAAA  2001:DB8::DE1E:6A7E
```

When a resolver attempts to look up a name like `domain.xy.exp1.example.net`, it will follow the delegation chain as indicated in Fig. 10. The solid black lines represent DNS messages sent over the network and the dashed green line delegations referring to a subzone on another name server. The name server authoritative for `example.net` refers the resolver to 192.0.2.1, to which the resolver will subsequently send the query for `domain.xy.exp1.example.net`. As the destination is not configured as authoritative name server for the queried name, we expect a SERVFAIL error response from the resolver. If `domain.xy` or any other part of the query name is blacklisted by DNS injection, then the resolver might receive one or more spoofed responses and return a bogus answer IP address to us. By setting up several delegations to different measurement destinations (experiment 1, 2, ...), we can test several paths between the resolver and arbitrary measurement destinations for occurrence of DNS injection.

The measurement method requires `domain.xy` to be blacklisted with any random suffix. As demonstrated in Section IV, this prerequisite does not apply to all domain names blacklisted by DNS injection, including `facebook.com`. We chose `www.minghui.org` as domain name to analyze the impact of Chinese DNS injection.

As shown above, each delegation also contains an AAAA record with an IPv6 address. This is not the IPv6 address of the measurement destination but instead is the address of our name server, pointing to itself. The reason for this seemingly useless delegation is the behavior of IPv6-capable resolvers: when the IPv4 measurement destination returns an error, the resolver attempts to look up the AAAA record of `ns.exp1.example.net` to find a working IPv6 name server. This causes additional AAAA queries which the resolver sends to the root, `.net` and our name servers in order to resolve `ns.exp1.example.net`. By putting a cacheable glue AAAA record into the delegation, an IPv6-capable resolver will send one additional IPv6 query to our name server, but none to the root and `.net` name servers, reducing the overall network load.

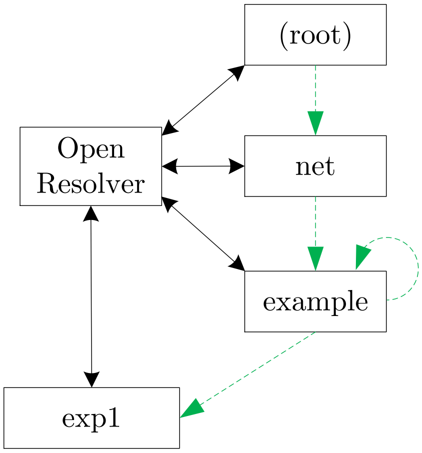

Fig. 10. Domain delegations set up (dashed lines) and DNS messages sent (solid lines) for impact measurement.

B. Measurement Result

As measurement destinations we composed a list of 1155 name servers that were authoritative for the DNS root zone or any of the 317 TLDs in July 2013. Each of the 1155 IPv4 addresses represents a separate experiment in the measurement. In this analysis, we omit 11 name servers that are authoritative for the three Chinese TLDs `.cn`, `.xn--fiqs8s`, `.xn--fiqz9s` because we study the effect of Chinese DNS injection on foreign third-parties.

As vantage points, we randomly selected 997,021 open resolvers[3] to test them with all measurement destinations. 709,446 open resolvers timed out repeatedly during the measurement, which lasted for several hours to keep the traffic profile low. This was to be expected: open resolvers are often connected via dynamic links with ephemeral availability. Fig. 11 shows the decreasing number of responding resolvers. Most timeouts occur at the first experiment, i.e. the open resolver was offline when the measurement has started. The graph decreases smoothly, except for some irregular drops which are a side effect of unreliable resolver implementations. TLD servers have usually a high availability but individual servers of smaller TLDs can show occasional downtimes. The common resolver behavior is to return SERVFAIL when the authoritative name servers have timed out. Some open resolvers, however, in this case do not return any response to the query sender and thus time out themselves, despite our timeout intervals of 30 up to 150 seconds. As the downtimes of the TLD servers vary over time, the timeouting open resolvers add up at different experiment numbers.

Besides timeouts, 17,138 open resolvers did respond but failed to resolve two well-known domain names correctly. Out of 270,437 open resolvers with complete and usable measurement results, 15,435 were located in mainland China

[3] List of open resolvers provided by OpenResolverProject.org [18].

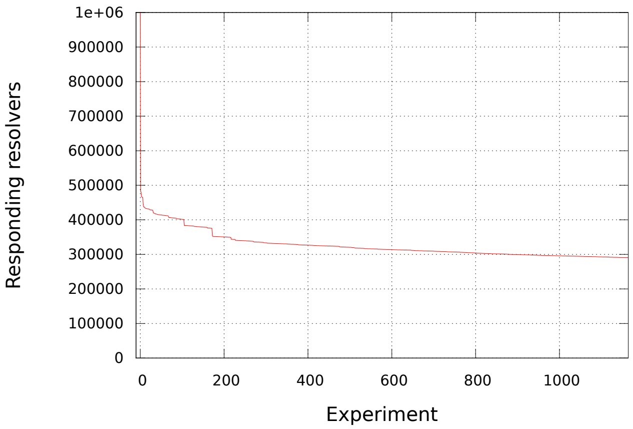

Fig. 11. Decreasing resolver availability.

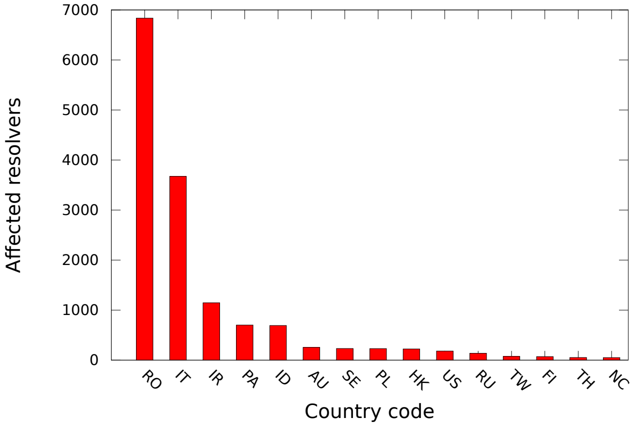

Fig. 12. Location of tested resolvers affected by DNS injection.

TABLE III
MOST FREQUENTLY AFFECTED NETWORKS.

Affected	CC	AS#	Organization
6826	RO	9050	Romtelecom
3455	IT	3269	Telecom Italia
701	PA	11556	C&W Panama
263	IR	12880	ITC Iran
231	SE	3301	TeliaSonera
166	IR	48159	TIC Iran

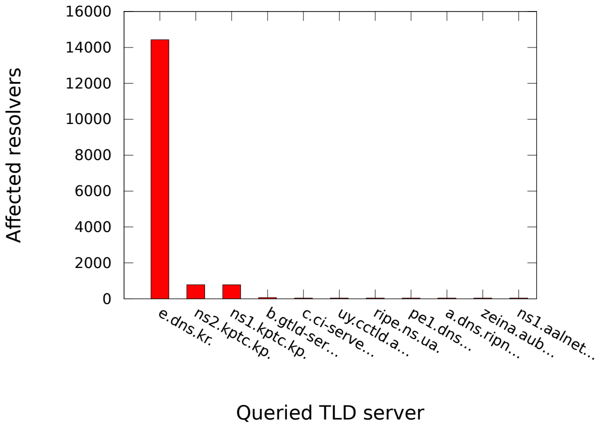

Fig. 13. Most frequently affected name servers.

and as expected all of them were affected by DNS injection. We consider an open resolver as affected if it returns a spoofed response in at least one experiment. For further analysis, we consider the remaining 255,002 open resolvers outside of China which should not be affected by a foreign censorship filter. The tested resolvers were located worldwide in 188 countries or regions. 15,225 resolvers (6.0%) from 79 countries were affected by Chinese DNS injection, which is shown in Fig. 12. In absolute numbers, most affected resolvers were located in Romania (91% affected of all tested Romanian resolvers) and Italy (53% affected of all tested Italian resolvers). Table III shows that for most countries the positive hits are essentially caused by one national network. This means a national network routed traffic to one of the measurement destinations through China whereas other networks—even those in geographical proximity—chose other routes.

The majority of spoofed responses were injected on behalf of one particular measurement destination. This can be seen in Fig. 13: 14,431 resolvers (5.7%) received spoofed responses when sending queries to `e.dns.kr`. There are six authoritative name servers for the South Korean TLDs `.kr` and `.xn--3e0b707e` but `e.dns.kr` is the only one which was affected by Chinese DNS injection. According to the operator KRNIC [7], `e.dns.kr` is using anycast addressing (AS23596) with locations in Daejeon (South Korea), Beijing (China), São Paulo (Brazil) and Seoul (South Korea). Given the absence of injection for the other five name servers which are all hosted outside of China, this suggests that DNS injection for `.kr` occurs only on paths to the anycast instance in Beijing.

DNS injection does not occur on all routes through mainland China, which can be seen in the results for the North Korean TLD `.kp`. The two name servers for `.kp` are both located in the same network (AS131279, Star JV). Public BGP data[4] suggests that this network uses one Chinese upstream provider (AS4837, CNC Group) and no anycast routing. In earlier studies AS4837 has been shown as censored network [4] [20]. Despite being routed through a Chinese network, only a minor portion of 794 (0.3%) resolvers were affected by DNS injection. As explained by Wright [21], the technical administration of Internet filtering in China is decentralized and thus leads to heterogeneous filter configurations.

Apart from the three South Korean and North Korean name servers, the remaining 1141 measurement destinations did not show significant traces of DNS injection. There were around 30 resolvers affected per each name server but exemplary analysis shows that these resolvers are suffering from injection regardless of the measurement destination. These resolvers are located in random networks, suggesting that this is not a network issue but rather a host-specific issue. A possible explanation is a malware similar to DNSChanger which forwards DNS queries to a destination in mainland China and thus triggers spoofed responses by the GFW unintentionally.

[4] Using BGPlay from RIPE NCC [19].

C. Impact and Protection

In the above measurement, we have been using the domain name `www.minghui.org` to trigger spoofed responses. In practice, the name servers for `.kr` and `.kp` are never queried for this domain name. The measurement results imply that 6% of the open resolvers worldwide suffer from Chinese DNS injection when resolving names below `.kr` or `.kp`, e.g. `epochtimes.co.kr`. The DNS community is well-aware of Chinese DNS injection and the adverse effect it can have on third-parties. Earlier studies showed significant occurrence of Chinese DNS injection for `.de` [4] [10]. In our mid 2013 measurement, there were no traces of DNS injection for `.de`, though one of the anycast instances of `a.nic.de` was hosted in Beijing at that time (via AS24151, CNNIC). This suggests that operators of global anycast name server networks can confine the effect with careful routing configuration.

Cryptographic integrity mechanisms like DNSSEC remediate the negative effects of DNS injection on third-parties, if fully deployed on client and server side. 26,170 (10%) of the tested open resolvers showed indication of DNSSEC validation by rejecting a crafted domain name with invalid DNSSEC signature. However, 18,092 (7%) of them relied on the validating Google Public DNS service. If those resolvers do not validate the responses additionally by themselves, they may be still subject to DNS injection on the path to the Google DNS servers. We thus recommend operating system vendors and home router vendors to consider DNSSEC integration into the local DNS resolver.

Without cryptographic integrity validation, the Hold-On method by Duan may help to detect DNS spoofing attacks [22]. With Hold-On, a resolver waits for multiple responses and discards apparently bogus responses by their IP TTL and round-trip time. The method is not as effective as DNSSEC and may produce false-negative results but provides opportunistic spoofing detection with a pure client-side deployment. Combined with the list of bogus addresses obtained with our method in Section V, the Hold-On method could help to mitigate the effect of DNS injection while DNSSEC is not universally deployed.

VII. Conclusion

In this paper, we studied the effects of Internet censorship by DNS injection from vantage points outside of the censoring networks. The measurement methods we have used are shown in Table IV. Our measurement data collected for this paper is available for public download [23]. We found Iranian networks as source for DNS injection visible from the Internet and compare their behavior with Chinese DNS injection. Unlike the Chinese DNS filter, the Iranian DNS filter takes unwanted DNS queries off the network. Bypassing the DNS filter from inside of an Iranian network thus requires to hide the DNS query, e.g. with a VPN or Tor. Chinese DNS injection can be observed for a much larger part of the public IPv4 address space, simply due to the fact that the number and size of Chinese networks is larger.

Iranian and Chinese DNS censorship can be detected via the bogus IPv4 addresses that are returned in spoofed DNS responses. The Iranian filter returns always the same address, while the Chinese filter returns a random address out of a set of addresses. We presented an efficient method to obtain the set of bogus addresses from the Chinese DNS filter. These addresses can be used for passive opportunistic detection of DNS censorship, e.g. in a DNS resolver or in a web browser.

Using open resolvers worldwide, we reappraised the impact of Chinese DNS injection on foreign TLD name servers. 6% of the resolvers outside of China received spoofed DNS responses, which was mostly the result of an anycast name server instance of `.kr` being hosted in China. There was no evidence for DNS spoofing with TLD name servers located outside of China on a larger scale.

In conclusion, care should be taken when name servers are supposed to be hosted in China or Iran. Even if the Internet access provider does not spoof responses, one of the upstream providers might do. DNSSEC can be used to cope with the unwanted effects of DNS injection on unrelated third-parties and other types of DNS spoofing.

Acknowledgment

The authors would like to thank Nils Kopal, Marius Liebeton, Arno Wacker for providing network infrastructure and Jared Mauch for sharing data from OpenResolverProject.org. This work was supported by Deutsche Forschungsgemeinschaft.

References

[1] J. Zittrain and B. Edelman, “Internet filtering in China,” *Internet Computing, IEEE*, vol. 7, no. 2, pp. 70–77, 2003.

[2] R. Clayton, S. J. Murdoch, and R. N. M. Watson, “Ignoring the great firewall of China,” in *In 6th Workshop on Privacy Enhancing Technologies*, 2006.

[3] G. Lowe, P. Winters, and M. L. Marcus, “The Great DNS Wall of China,” 2007. [Online]. Available: http://cs.nyu.edu/ pcw216/work/nds/final.pdf

[4] Anonymous, “The Collateral Damage of Internet Censorship by DNS Injection,” *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, pp. 21–27, 2012. [Online]. Available: http://doi.acm.org/10.1145/2317307.2317311

[5] M. V. Ereche, “Odd behaviour on one node in I root-server,” dns-oarc.net [dns-operations] mailing list, Mar 2010. [Online]. Available: https://lists.dns-oarc.net/pipermail/dns-operations/2010-March/005260.html

[6] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston, “Internet Background Radiation Revisited,” in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’10. New York, NY, USA: ACM, 2010, pp. 62–74. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879149

[7] KISA KRNIC, “DNS Name Server.” [Online]. Available: http://krnic.or.kr/jsp/english/dns/nameServer.jsp

[8] S. Aryan, H. Aryan, and J. A. Halderman, “Internet Censorship in Iran: A First Look,” in *Free and Open Communications on the Internet*. Washington, DC, USA: USENIX Association, 2013.

[9] S. K. Dehghan, “Iran’s president signals softer line on web censorship and Islamic dress code,” *The Guardian*, Jul. 2013. [Online]. Available: http://www.theguardian.com/world/2013/jul/02/iran-president-hassan-rouhani-progressive-views

[10] P. Koch, “Using RIPE Atlas: A DENIC Case Study.” [Online]. Available: https://labs.ripe.net/Members/pk/denic-case-study-using-ripe-atlas

[11] P. Smith, “BGP Routing Table Analysis,” 2013. [Online]. Available: http://thyme.apnic.net/

[12] MaxMind, “GeoLite data,” 2013. [Online]. Available: http://www.maxmind.com

[13] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, “Address Allocation for Private Internets,” RFC 1918 (Best Current Practice), Internet Engineering Task Force, Feb. 1996, updated by RFC 6761. [Online]. Available: http://www.ietf.org/rfc/rfc1918.txt

TABLE IV
OVERVIEW OF MEASUREMENT METHODS USED.

Section	Purpose	Method	Input	Output
III	Find censored networks	Probe random networks	47 candidates domains	1M affected IP addresses 4 censored domains
IV	Find blacklisted names	Probe censored networks	1M affected IP addresses 1M candidate domains	404 censored domains (CN) 14 censored domains (IR)
V	Obtain bogus addresses	Probe censored networks	1M affected IP destinations 404 censored domains	33 bogus IP addresses (CN)
VI	Determine affected networks	Probe open resolvers	1M open resolvers 1155 name servers	15,225 affected resolvers

[14] C. Anderson, “The Hidden Internet of Iran: Private Address Allocations on a National Network,” *CoRR*, vol. abs/1209.6398, 2012.

[15] Ministry of Communications and Information Technology, Oct. 2013, announcement in Persian language. [Online]. Available: https://www.ict.gov.ir/fa/news/8741

[16] Alexa – The Web Information Company, “The Top 1 000 000 Sites on the web.” 2014. [Online]. Available: http://www.alexa.com

[17] K. P. Gummadi, S. Saroiu, and S. D. Gribble, “King: Estimating Latency between Arbitrary Internet End Hosts,” in *SIGCOMM Internet Measurement Workshop*, 2002.

[18] J. Mauch, “Open resolver project,” 2013. [Online]. Available: http://openresolverproject.org

[19] RIPE NCC, “BGPlay,” 2013. [Online]. Available: https://stat.ripe.net/widget/bgplay

[20] X. Xu, Z. Mao, and J. Halderman, “Internet censorship in china: Where does the filtering occur?” in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, N. Spring and G. Riley, Eds. Springer Berlin Heidelberg, 2011, vol. 6579, pp. 133–142.

[21] J. Wright, “Regional variation in chinese internet filtering,” *Information, Communication & Society*, vol. 17, no. 1, pp. 121–141, 2014.

[22] H. Duan, N. Weaver, Z. Zhao, M. Hu, J. Liang, J. Jiang, K. Li, and V. Paxson, “Hold-On: Protecting Against DNS Packet Injection,” in *Securing and Trusting Internet Names (SATIN)*, 2012.

[23] M. Wander, C. Boelmann, L. Schwittmann, and T. Weis, measurement data used for this paper. [Online]. Available: http://dnssec.vs.uni-due.de/dnsinjection/