Extended Abstract: Understanding the Privacy Implications of ECS

Panagiotis Kintis[1], Yacin Nadji[1], David Dagon[1],
Michael Farrell[2], and Manos Antonakakis[3]

[1] Georgia Institute of Technology, School of Computer Science,
{kintis,yacin}@gatech.edu,dagon@m.sudo.sh
[2] Georgia Institute of Technology, Institute for Internet Security & Privacy,
michael.farrell@iisp.gatech.edu
[3] Georgia Institute of Technology, School of Electrical and Computer Engineering,
manos@gatech.edu

Abstract. The edns-client-subnet (ECS) is a new extension for the Domain Name System (DNS) that delivers a "faster Internet" with the help of client-specific DNS answers. Under ECS, recursive DNS servers (recursives) provide client network address information to upstream authorities, permitting topologically localized answers for content delivery networks (CDNs). This optimization, however, comes with a privacy penalty that has not yet been studied. Our analysis concludes that ECS makes DNS communications less private: the potential for mass surveillance is greater, and stealthy, highly targeted DNS poisoning attacks become possible.
Despite being an experimental extension, ECS is already deployed, and users are expected to "opt out" on their own. Yet, there are no available client-side tools to do so. We describe a configuration of an experimental recursive tool to reduce the privacy leak from ECS queries in order to immediately allow users to protect their privacy. We recommend the protocol change from "opt out" to "opt in", given the experimental nature of the extension and its privacy implications.

1 Introduction

In 2011, an experimental Internet draft [5], suggesting some extensions to the Domain Name System (DNS), was proposed to the Internet Engineering Task Force (IETF). The proposed changes enable more efficient content delivery services, especially when edge devices make use of remote, open recursive servers. These changes specifically call for the addition of the edns-client-subnet (ECS) extension, and as part of this effort, an initiative for "a faster Internet" [17] was announced to promote collaboration. The latest update of the ECS Internet draft, including the technical and implementation details, was published in November 2014 [7].

ECS has seen increased adoption and delivered on its promise of a faster Internet for end users [21]. This improvement in performance, however, was achieved by changing the data shared between recursive DNS servers and authoritative DNS servers, which we will henceforth refer to as *recursives* and *authoritatives*. Historically, the authoritative only received (1) the fully qualified domain name and (2) the IP address

of the recursive attempting to resolve the domain on behalf of the end user (located at the edge). The authoritative used the fully qualified domain name in order to provide an answer for the given DNS request. Optionally, the authoritative could provide an "optimized" answer based on the recursive's IP. In the past, this was a reasonable optimization because the recursive and client were more closely linked to each other. For example, the recursive was often provided by the client's Internet Service Provider (ISP). With the advent of large open recursives this is no longer a safe assumption.

By using ECS, the recursive can reveal a truncated portion of the client's IP address, such as the first three octets of the IP. This allows the authoritative to optimize answers for the client issuing the DNS request, rather than the client's recursive. This is particularly useful for content delivery networks (CDN). If a client uses a large open recursive, adding ECS provides better performance due to the improved localization in the authoritative's answers. Studies have shown that ECS not only decreases latency for end users but also has seen increased adoption worldwide [4], perhaps due to these benefits. While the performance improvements are clear, ECS allows anyone in the path between the authoritative and the recursive to surreptitiously read some bits of the client's IP address, which may raise privacy concerns.

Despite the IETF draft acknowledging that there might be privacy issues for clients using an ECS-speaking recursive, very little attention has been paid to identify and evaluate any possible privacy issues with the new extension. In fact, the IETF draft states that there should be a way for users to "opt out" of ECS. The draft also suggests that users should be able to specify how much of their IP address they wish to reveal to the remote authoritative. However, nearly **five years** after the ECS draft was proposed, there are still no client centric tools that empower users to control how much of their IP address is revealed. Thus, ECS is effectively opt-out in nature. In this study, we argue that ECS should be opt-in by default. As we discuss in detail, ECS could be "weaponized" for surveillance or targeted DNS poisoning attacks.

In summary, the study aims to increase the situational awareness around the use of ECS by making the following observations:

1. We describe the potential for novel surveillance and targeted cache poisoning attacks that are made possible by ECS. ECS adds an additional location where surveillance can be performed: between a client's recursive and the domain's authority. The targeted cache poisoning attack can selectively reroute users, down to the granularity of an IP address and be performed while making postmortem forensic analysis difficult.
2. We describe how to set up a custom version of Unbound to opt out of ECS that can be used by end users now who are concerned about their privacy. Prior to this, there was no way for users to opt out.

2 Background

In the following sections, we will discuss ECS and the fundamental technologies on which it relies. Since ECS is simply an extension on top of existing DNS infrastructure, we will start with a short discussion of DNS in Section 2.1. This will be followed by a more thorough discussion of the changes introduced by ECS in Section 2.2.

2.1 DNS Basics

The Domain Name System (DNS) [15] is a fundamental service that enables ease of use of the Internet. Its primary goal, is to translate human readable text (domain names) to IP addresses, like `example.com` to `93.184.216.34`. In order for this process to happen, a series of eight steps must take place. In step one a stub resolver or client (i.e. web browser, application, etc), submits a DNS resolution request to a DNS recursive server, referred to as *recursive* from this point on. The recursive server server will look for the domain name to IP address mapping in its cache memory; if found there, the latter will inform the stub of the IP address and the process will end. In the case where the IP address(es) is not available in the cache memory, the recursive will ask one of the 13 root servers for the IP address to which the domain name points, on behalf of the stub resolver (step two). The root server will reply with the IP address of the Top Level Domain server (or TLD) of the domain name for which it was queried (step three). During step four, the recursive will submit a query to that TLD for the domain server. The TLD will respond with the IP address of the Authoritative Server (referred to as *authoritative* from now on) and complete step five. Lastly, during steps six and seven, the recursive will communicate with the authoritative, asking for the IP address of the domain name and the latter will reply with it, in the simple case. Finally, step eight concludes the process, during which the recursive informs the stub resolver the IP address of the domain name it looked up.

This process can be divided in two communication phases: (1) the first one is between the stub resolver and the recursive, also known as *below the recursive*; and (2) the second is between the recursive and the servers queried in the DNS hierarchy, commonly referred to as *above the recursive*. The next section describes how the adoption of ECS affects the communication above the recursive.

2.2 Evolution of DNS with ECS

The previous section discussed the domain name resolution process. The adoption of ECS does not change that process, however, the information exchanged between recursives and authoritatives does change. Prior to ECS, only communication below the recursive contained information about the client performing a DNS query. Thus, any communication with the authoritative happened above the recursive, and the authoritative received no information about the client responsible for a particular request. ECS, embeds a truncated portion of the client's IP address, referred to as the *source netmask*, into communication above the recursive. According to the ECS RFC [7], the source netmask should be determined using the most detailed network information available to the recursive, but by default, it will include the first three octets of a client's IP address. The authoritative's reply will contain a *scope netmask* that may guide a recursive's future choice of source netmask. The scope netmask indicates the authoritative's desired source netmask length and should indicate the minimum source netmask required to return an optimal answer, with respect to network performance.

These changes were prompted due to the introduction and growing use of large, open recursives [16,11,10]. Traditionally, recursives were strongly tied to a client's network, and therefore, they served as a reasonable proxy for a client's location.

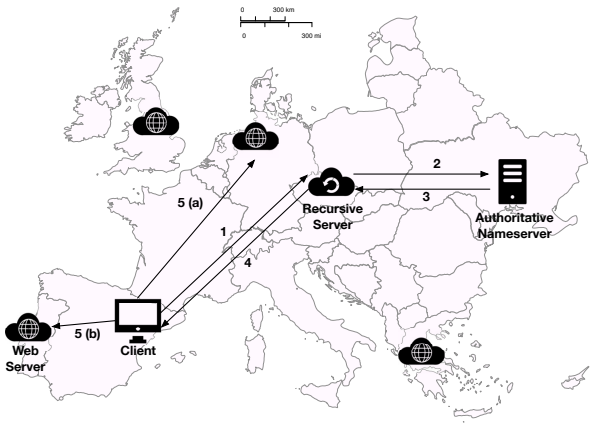

Fig. 1: The image shows the DNS resolution process. The last step of the resolution request (5), is split in two cases. The first case shows how the resolution would take place without using ECS, in step 5(a), whereas the second one shows the different reply when utilizing the client subnet, in step 5(b).

Public recursives, however, need not be related to a client's network or be in close geographic proximity to the client.

Figure 1 shows an example of the problems that widespread use of public recursives can cause and how ECS can help. In the figure, a client in Spain connects to a public recursive in Czech Republic to resolve the domain name for a web server. Steps (1) to (4) are part of every resolution process, when a domain name is not stored in a recursive's cache, as described in Section 2.1. In this particular case, when the authority notices the IP address of the recursive being in Czech Republic, it will assume that the client is in the same geographical region and therefore reply with an IP address for the web service close to that. Provided that the web service is using a CDN or load balancing techniques to increase efficiency for both the provider and the client, this geographical assumption can actually be detrimental. As shown in step 5(a), the client will connect to a web server located in Germany, which is far less efficient than the one in Portugal. When ECS is used, the authority will be able to identify the geographic location of the client since the client's subnet is shared. Thus the authority will make an informed decision about the reply it will provide. In step 5(b), we can see that now the client is provided with an IP address of the web service located in Portugal and connects to that one instead.

3 Surveillance and Selective Cache Poisoning

Spurious ECS "speakers" enable Internet miscreants to potentially perform discreet and powerful surveillance and targeted cache poisoning attacks. The targeted poisoning can selectively reroute users, even down to the granularity of a specific IP address, to hosts under their control. It is important to note these attacks only require visibility of the path between a recursive and authoritative DNS server.

3.1 Surveillance

State sponsored surveillance has seen increased coverage in the press [9], with nation states using a myriad of techniques to monitor users. In light of such revelations, it is more important than ever to evaluate the privacy impact new technologies may have on the public – even if negative consequences are unintended. ECS provides another means for nation states or network operators to monitor individuals or groups on the Internet.

ECS simply makes surveillance easier. First, the introduction of ECS allows client information to be collected from a different vantage point: where an adversary is located between a client's recursive and the domain's authoritative. Second, since surveillance is done based on the domain name of the target server rather than its IP address, surveillance can be more fine-grained in instances where hosting is shared.

Prior to ECS, collecting DNS traffic above the recursive only revealed the IP address of the recursive; nothing was revealed about the user responsible for the original request. However, this is no longer true for ECS enabled domains. As discussed in Section 2.2, ECS allows a truncated version of the user's IP address to be embedded in a DNS request; this allows user-level surveillance to be approximated *above the recursive*, increasing the usefulness of DNS for surveillance of individual users on the Internet.

This change allows surveillance to be performed if the spying party is located in the path between the user's recursive and the authoritative of the domain name the user is querying (steps 2 and 3 in Figure 1). This surveillance is less informative but more specific than IP-based surveillance that would occur between the user and the application server (steps 1 and 4 in Figure 1). With ECS surveillance, only a portion of the user's IP address will be revealed, however, this will often allow the organization the user is connecting from to be identified. One benefit, however, is the specificity offered by performing surveillance on the server's domain name, rather than its IP address.

In a shared hosting environment, one IP address can host many distinct services separated by the domain name they use. For example, popular HTTP server software allows multiple websites to be hosted differentiated only by the domain name used to resolve the server's IP address. In instances where ECS enabled surveillance is performed, this can be catered specifically to the domain name used rather than the server's IP such that fewer packets have to be analyzed. For example, multiple blogs each using a distinct domain can resolve to one IP address. In the ECS surveillance case, monitoring a specific blog is easier. Furthermore, if used in concert with existing IP-based surveillance between the client and the server, an exact client IP match can be unified with more specific ECS enabled match by the server's domain name. Even more unsettling is the possibility of selective cache poisoning.

3.2 Selective Cache Poisoning

In the traditional context, a DNS Cache Poisoning attack, aims to insert false domain name to IP address mapping pairs in a recursive's cache memory. As explained in Section 2.1 the recursive server will store a response it receives from an authoritative for as long as it is instructed; this information is in the TTL field of the response. A fundamental problem in DNS is that the recursive cannot be sure that the response it received was actually from the authoritative or a rogue entity that submitted a response faster. To mitigate this problem randomization has been introduced in the DNS

packets. This makes it harder for an adversary to correctly craft a packet that matches the response expected by the recursive. Countermeasures include the Transaction ID field, source port randomization [15] and 0x20 [8]. DNSSEC [2] is probably the best defense against cache poisoning, since the authoritative can use public/private key encryption to certify its identity to the recursive. It is worth noting that, besides DNSSEC, any other attempt to increase the packet entropy and make it harder for the attacker to succeed is only plaintext information within the packet itself. Anyone with access to the packet is able to construct a response, exactly as the authoritative would.

ECS information carried in the DNS packet when a resolution request takes place allows the authoritative to approximate with increased accuracy the geographic location of the entity that initiated the recursive procedure. This is also true for every other entity that is able to monitor the traffic between the recursive and the authoritative servers. For instance, a third party, can *tap the wire* and start collecting information about the clients performing resolution requests. Accessing the *question* packet means that any arbitrary response can be constructed, which will be accepted by the recursive server. Of course this is not something new and any network administrator between a recursive and an authoritative would be able to do it, but the consequence would have been to redirect all traffic around the world to a different IP address than the real one.

Using the client subnet within the ECS-enabled DNS packets, a network administrator could be motivated to change a response and make a recursive server "think" that a domain name points to a different IP address. In this case, a network administrator would do this if she wanted to impact a specific IP address, subnet or geographic location. The current implementation of ECS not only supports such behavior (it is in fact the reason ECS was created), but also understands the difference in caching for the affected subnet. For example, someone might be interested in manipulating only hosts in 10.0.0.0/24. Crafting a response for a resolution request that contains this subnet mask in the payload will force the recursive to cache this domain name to IP address mapping. This will happen for future client DNS lookup requests where the IP address is within that network. This network can be arbitrarily small, even targeting a specific IP address, i.e., 10.0.0.0/32. Every other client will be served with a different mapping pair. A truly stealthy adversary could set the time-to-live (TTL) of the DNS packet to zero to leave the minimum possible forensic trail.

Figure 2 shows how this scenario is possible to occur, when the adversary has access to the network traffic between the recursive and the authoritative server. The adversary will need to be able to process the incoming UDP DNS packets to the network and construct a reply faster than the authoritative. To prove the practicality of the attack, we used our ECS enabled domain name and targeted all IP addresses in a network we have machines located in (***.251.0.0/16). Using a simple network packet sniffer we were able to parse the DNS packets, extract the transaction ID, source port and requested domain fields, which are randomized by most recursive servers, and craft a custom response for requests with subnet masks within the aforementioned network. We used Google Public DNS (8.8.8.8) as the ECS-enabled recursive server, for the requests we submitted to be resolved for our clients. We have made a video[4] that demonstrates the feasibility of this attack, where a request is first sent from an IP address outside of the network and then another one with source IP address that

[4] `https://youtu.be/U1ehqjGwETc`

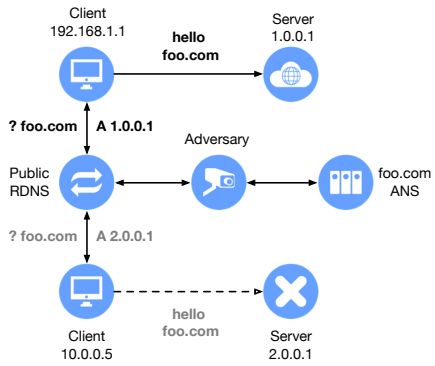

Fig. 2: An adversary monitoring the traffic between the recursive and the authoritative is able to selectively poison the cache of the recursive, even without ever capturing traffic from the clients initiating the requests.

matches our filters. The video shows the network topology on which the attack took place and is similar to Figure 2. In the first case, the machine is not targeted and the poisoning attack does not occur. In the second case, however, the IP address is changed using a VPN service, making the filters trigger and the RDNS replies with the injected IP address 1.3.3.7. The output of both the packet sniffer that performs the injection and the *tcpdump* tool running on the authority is shown during the resolution requests.

Things can be more serious when one considers that the edns-client-subnet draft does not specify whether the DNS requests to the root and Top Level DNS (TLD) servers should carry the client's subnet mask or not. It explicitly states that the recursive servers *MAY* be configured to not send the ECS option to them, but it is not enough to ensure security. In a case where the recursive shares the exact same packet with the root and TLD servers, then the adversary does not need to be between the recursive and the authoritative exclusively. All an attacker needs is the recursion path to include a root or TLD server within the network she is able to monitor. Thus, any DNS query that the adversary is in position to identify can result to a selective cache poisoning attack.

4 Remedies

ECS enables potentially devastating attacks, but these shortcomings can be alleviated easily while maintaining the known performance improvements of ECS. First, ECS should be opt-in by default rather than opt-out to protect potential victims of surveillance and cache poisoning. Second, despite the fact that the RFC for ECS allows clients to specify the number of bits in their source netmask, there is currently no method for users to do this; this issue is exacerbated by the fact that ECS is already deployed. To this end, we provide instructions for configuring a personal recursive to specify a source netmask of 0 bits, effectively opting out of ECS entirely.

```
$ svn co http://unbound.nlnetlabs.nl/svn/branches/edns-subnet/
$ cd edns-subnet && ./configure --enable-subnet && make && make install
$ cat > /etc/unbound/unbound.conf <<EOF
server:
    verbosity: 1
    interface: ${INTERFACE}
    outgoing-interface: ${OUTBOUND_INTERFACE}
    use-caps-for-id: yes
    access-control: 127.0.0.0/8 allow # Allow local access only
    module-config: "subnetcache validator iterator"
    client-subnet-opcode: 8
    max-client-subnet-ipv6: 0
    max-client-subnet-ipv4: 0
forward-zone:
    name: "."
    forward-addr: ${RECURSIVE_IP_PRIMARY}
    forward-addr: ${RECURSIVE_IP_SECONDARY} # If needed
EOF
$ /usr/local/sbin/unbound -c /etc/unbound/unbound.conf
```

Fig. 3: Installation and configuration for `Unbound` recursive software to send scope-0 by default.

Opt-in vs. Opt-out Many of the privacy issues that stem from ECS are due to the rapid deployment of ECS and the assumption that all users want to enable ECS by default. We argue that ECS should be disabled by default and users and networks should opt in to the service, rather than have it enabled by default and force users to opt out. In instances where performance is key, ECS is clearly beneficial, but ECS has the potential to be abused to infringe on users' privacy.

Tools to Opt Out While the RFC suggests users can adjust the source netmask setting to cater to their privacy needs, this is not possible with standard tools. To assist privacy-conscious users we have provided instructions to compile and configure the ECS version of the `Unbound` recursive in Figure 3. This configuration will forward connections to the recursive specified while sending scope-0.

5 Related Work

DNS is already known to be able to interfere with a user's privacy. Krishnan et al. [14] have shown how DNS prefetching can leak information regarding users' activity online, to a degree that information regarding web searches can be inferred by simply logging the resolution requests a web browser is making. Zhao et al. [22] performed a deep analysis on each step of a domain name resolution process, showing information that can be inferred from users' private data by only looking at public data. They also propose a simple range query scheme that can be used to protect the user. In the same context, Guha and Francis [12] describe an attack against the DNS by passively monitoring DNS-related traffic. This attack can provide a variety of information about a user that includes location, habits, and commute patterns. Lastly, an Internet Draft by Bortzmeyer [3] attempts to enumerate DNS-only attacks and their privacy implications. These are aggregated into six different categories, and the authors concluded their work with several security considerations on the matter. Some work

has extended this to ECS, identifying cases where details of the infrastructure can be uncovered [21] identify more accurate geographic locations [6,18], but the implications for surveillance and poisoning have not been studied so far.

DNS cache poisoning attacks are well understood outside of the context of ECS. In [20], Stewart Joe describes two types of attacks and provides a brief history of the DNS cache poisoning evolution. In 2008, Kaminsky [13] demonstrated a new version that was able to cache poison DNS recursives much more efficiently and overcame all known countermeasures at the time. New controls and mitigation techniques were suggested and deployed: (1) Anax is a system able to identify poisoned records in the cache of a recursive resolver [1]; and (2) WSEC DNS [19] utilizes random subdomain strings for entropy WSEC DNS [19], utilizes random subdomain strings for entropy increase and poison resistance for the packets exchange between DNS servers. Lastly, 0x20 [8] proposed to randomize the case of the question a recursive submits to other DNS servers. This is effective since the protocol is case-insensitive, further increasing the number of attempts needed to successfully poison the cache of a recursive.

6 Conclusions

In this work we discussed ways that ECS can be used to help augment surveillance, and enable extremely targeted cache poisoning attacks against DNS. The latter is especially concerning. Even though ECS has not been officially standardized, it has seen increased adoption over the last several years. Therefore, the unintended consequences introduced by ECS represent *current* threats to Internet users and should be addressed sooner rather than later. To this end, we acknowledge the benefits that ECS provides, but we propose that it should be Opt-In instead of Opt-Out. We also propose a patch to the popular `Unbound` recursive DNS server that helps users opt-out of using ECS. However, in order to have broader impact, popular public recursives should provide their own mechanisms for disabling ECS, and they should make ECS usage opt-in only.

Acknowledgments. This material is based upon work supported in part by the US Department of Commerce under grant no. 2106DEK and Sandia National Laboratories grant no. 2106DMU. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US Department of Commerce nor Sandia National Laboratories.

References

1. Antonakakis, M., Dagon, D., Luo, X., Perdisci, R., Lee, W., Bellmor, J.: A Centralized Monitoring Infrastructure for Improving DNS Security. In: Recent Advances in Intrusion Detection. pp. 18–37. Springer (2010)
2. Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: DNS Security Introduction and Requirements. RFC 4033 (Proposed Standard) (Mar 2005), `http://www.ietf.org/rfc/rfc4033.txt`, updated by RFCs 6014, 6840
3. Bortzmeyer, S.: DNS Privacy Considerations. `https://tools.ietf.org/id/draft-bortzmeyer-dnsop-dns-privacy-02.txt` (April 2014)

4. Calder, M., Fan, X., Hu, Z., Katz-Bassett, E., Heidemann, J., Govindan, R.: Mapping the Expansion of Google's Serving Infrastructure. In: Proceedings of the 2013 Conference on Internet Measurement Conference. pp. 313–326. IMC '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2504730.2504754
5. Contavalli, C., Gaast, W.V.D., Leach, S., Rodden, D.: Client Subnet in DNS Requests (draft-vandergaast-edns-client-subnet-00) (2011), https://www.ietf.org/archive/id/draft-vandergaast-edns-client-subnet-00.txt
6. Contavalli, C., Leach, S., Lewis, E., Gaast, W.V.D.: Client subnet in DNS requests (2013)
7. Contavalli, C., Leach, S., Lewis, E., Gaast, W.V.D.: Client Subnet in DNS Requests (draft-vandergaast-edns-client-subnet-02) (2014), https://datatracker.ietf.org/doc/draft-ietf-dnsop-edns-client-subnet/
8. Dagon, D., Antonakakis, M., Vixie, P., Jinmei, T., Lee, W.: Increased DNS forgery resistance through 0x20-bit encoding: security via leet queries. In: Proceedings of the 15th ACM conference on Computer and communications security. pp. 211–222. ACM (2008)
9. Electronic Frontier Foundation: Mass Surveillance Technologies. https://www.eff.org/issues/mass-surveillance-technologies (2015)
10. Federrath, H., Fuchs, K.P., Herrmann, D., Piosecny, C.: Privacy-preserving DNS: analysis of broadcast, range queries and mix-based protection methods. In: Computer Security–ESORICS 2011, pp. 665–683. Springer (2011)
11. Google: Introduction to Google Public DNS. https://developers.google.com/speed/public-dns/docs/intro, https://developers.google.com/speed/public-dns/docs/intro, accessed: 2015-04-07
12. Guha, S., Francis, P.: Identity trail: Covert surveillance using DNS. In: Privacy Enhancing Technologies. pp. 153–166. Springer (2007)
13. Kaminsky, D.: Black ops 2008: It's the end of the cache as we know it. Black Hat USA (2008)
14. Krishnan, S., Monrose, F.: DNS prefetching and its privacy implications: When good things go bad. In: Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more. pp. 10–10. USENIX Association (2010)
15. Mockapetris, P.: Domain names - implementation and specification. RFC 1035 (INTERNET STANDARD) (Nov 1987), http://www.ietf.org/rfc/rfc1035.txt
16. OpenDNS: The OpenDNS Global Network Delivers a Secure Connection Every Time. Everywhere. http://info.opendns.com/rs/opendns/images/TD-Umbrella-Delivery-Platform.pdf (2010), http://info.opendns.com/rs/opendns/images/TD-Umbrella-Delivery-Platform.pdf
17. OpenDNS: A Faster Internet: http://www.afasterinternet.com (2011), http://www.afasterinternet.com
18. Otto, J.S., Sánchez, M.A., Rula, J.P., Bustamante, F.E.: Content delivery and the natural evolution of DNS: remote DNS trends, performance issues and alternative solutions. In: Proceedings of the 2012 ACM conference on Internet measurement conference. pp. 523–536. ACM (2012)
19. Perdisci, R., Antonakakis, M., Luo, X., Lee, W.: WSEC DNS: Protecting recursive DNS resolvers from poisoning attacks. In: Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on. pp. 3–12. IEEE (2009)
20. Stewart, J.: DNS cache poisoning–the next generation (2003)
21. Streibelt, F., Böttger, J., Chatzis, N., Smaragdakis, G., Feldmann, A.: Exploring EDNS-client-subnet adopters in your free time. In: Proceedings of the 2013 conference on Internet measurement conference. pp. 305–312. ACM (2013)
22. Zhao, F., Hori, Y., Sakurai, K.: Analysis of Privacy Disclosure in DNS Query. In: Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on. pp. 952–957. IEEE (2007)