Towards Tackling Privacy Disclosure Issues in Domain Name Service

Xuebiao Yuchi, Guanggang Geng, Zhiwei Yan, and Xiaodong Lee
National Engineering Laboratory for Naming and Addressing, China Internet Network Information Center, Beijing, China
{yuchixuebiao, gengguanggang, yanzhiwei, xl}@cnnic.cn

***Abstract*—Serving as the global Internet's phonebook, the Domain Name Service (DNS) helps to translate human-friendly domain names into machine-readable IP addresses, which makes DNS of great importance to the operation of the Internet and virtually relied on by today's almost all kinds of Internet-based activities. As such, people whoever want to go anywhere over the Internet will need to refer to the DNS first. Therefore, it has become an ideal way to conduct online privacy exploitations through the DNS due to people's pervasive usage of the Internet. However, the current DNS doesn't provide any countermeasure against this kind of exploitation, and thus risks severe privacy disclosure problems. In this paper, we give a comprehensive empirical analysis of DNS privacy disclosure problems by exploring potential privacy leaking paths in the DNS. Then we further identify and describe multiple criterions of validity systematically that are obligated when considering DNS privacy preservation. Finally, we propose a simple DNS privacy preserving solution with significant deployment potential in the current DNS, which can only lead to a moderate level of extra query latency perceived by end users.**

Keywords—Domain Name Service; privacy disclosure; privacy preserving.

I. INTRODUCTION

As a global hierarchically distributed directory service, the Domain Name Service (DNS) provides the translating functionality between human-friendly domain names and machine-readable IP addresses, which makes itself the most critical infrastructure component of the Internet and heavily relied by today's almost all kinds of Internet activities. Thereby, for a person who wants to go somewhere over the Internet will need to resort to the DNS first to get corresponding directional instructions. In this context, someone can learn a lot about people's almost all online source-target information from the DNS. However, as one of the oldest pieces of infrastructure used in the Internet, the DNS was initially designed as some kinds of open and public service of the Internet and didn't account for privacy with all its data transmitted in the clear. As a result, considerable and sensitive information that are valuable for user profile characterization could be easily exploited from the DNS.

In this paper, we focus our work on the privacy disclosure issues of the DNS. By exploring all possible privacy leaking paths in the DNS, we first present our empirical analysis of the privacy disclosure problems in the DNS. Then we introduce and detail multiple criterions of validity that are obligated for DNS privacy preservation. Finally, we propose a novel scheme to address the DNS privacy disclosure problems, and further validate its effectiveness and availability according to real world experiments. The rest of the paper is organized as follows. We discuss the related work in Section 2. Then we present our analysis of the DNS privacy disclosure problems and describe the criterions of validity that are essential for privacy preserving in Section 3. We propose our DNS privacy preserving solution and further validate it through experiments in Section 4. Finally, we discuss our work and conclude this paper in Section 5.

II. RELATED WORK

While the privacy issues have been studied intensively over the decades, the privacy issue of the DNS has attracted much less attention and has been largely ignored by the Internet research community. Due to the recent revelations on pervasive monitoring by nation-state surveillance [5], the need for a private DNS has become of great interest in recent years. Several different methods to address this growing problem have been proposed, with a large portion of them mainly taking ways of encrypting the DNS transmitted data, such as *DNS over TLS* [1]. However, their proposal can only provide the transmitted data encryption between recursive and authoritative DNS servers, while these DNS servers themselves still have the ability to spy on the data. Besides DNS data encryption, the IETF DPRIVE working group introduces a technique called "*qname minimization*" [2], where the recursors no longer needs to send the full query name, but only as much of the name as is necessary for making progress in the resolution process, to the upstream authoritative DNS servers. Yet again, the DNS privacy problems can only be solved partially by this solution. First, the full query name will be finally exposed in plaintext during the final step of the resolution process between the recursive and authoritative DNS servers. Second, the proposal doesn't address the privacy leaking problem at the recursor side who still have the ability to spy on the data.

Zhao et al. [3] propose to ensure the DNS privacy by concealing the actual queries using noisy traffic. However, the privacy ensured by added queries is difficult to analyze and that the technique introduces noticeable additional latency and overhead, making it impractical in real world deployment. In order to fully solve the DNS privacy disclosure problems, some radical solutions propose to replace the current DNS with some alternative peer-to-peer name systems, such as *GNS* [4] and *Namecoin* [5]. For example, the *GNS* resolution process utilizes a *distributed hash table* (DHT) and peer-to-peer technologies to enable users to find out key-value mappings, which departs significantly from that of the current DNS. While promising, we do not expect that these radical

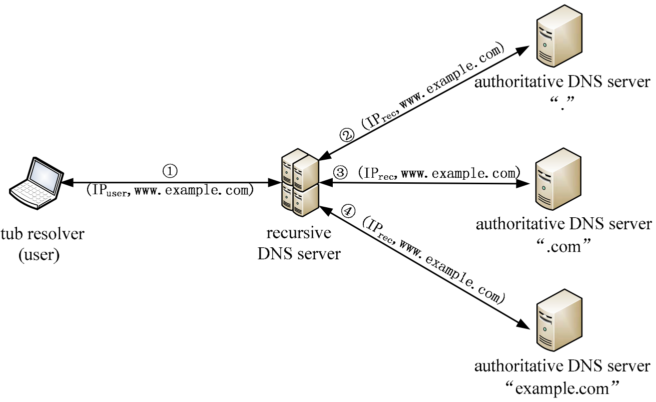
978-3-901882-89-0 @2017 IFIP

solutions could be widely adopted in the near future due to the need for a completely different DNS infrastructure and its high computational complexity which requires special hardware. In general, there still lacks of readily available, practical and effective solution for the DNS privacy preservation so far.

III. Problem description

A. DNS Overview

Let's first take an overview of how the DNS runs. Generally, the DNS infrastructure consists of three different types of components: stub resolvers (on behalf of users), recursive DNS server, and authoritative DNS servers (Fig. 1). Typically, when a user wants to establish an Internet connection with some remote resources, the stub resolver first needs to launch a DNS query for a corresponding domain name towards its configured recursor (recursor for short), which is commonly provided by the local ISP or a third party such as *Google Public DNS*. The recursor will then forward this query to these authoritative DNS servers iteratively until it receives an authoritative answer to this query (step 2 - 4). Finally, the recursor can reply the stub resolver with this answer, and also have this answer cached locally for a certain while in case of reuse. As illustrated in Fig. 1, each of the DNS queries will contain two major pieces of information, namely, the source IP of the originator who launches the query (namely, IP_{user} or IP_{rec}) and the targeted domain name that the user is looking for (namely, *www.example.cn*).



① (IP$_{user}$, www. example. com)
② (IP$_{rec}$, www. example. com)
③ (IP$_{rec}$, www. example. com)
④ (IP$_{rec}$, www. example. com)
stub resolver (user)
recursive DNS server
authoritative DNS server "."
authoritative DNS server ".com"
authoritative DNS server "example.com"

Fig. 1. The current DNS query process.

B. Risks Analysis

From the above description, we can infer once a user launches a DNS query for some specific domain name, the user's IP address and targeted domain name will be explicitly exposed to the DNS severs. Moreover, in the current DNS all querying behaviors are conducted via UDP in plaintext. In other words, the current DNS provides no privacy at all. As a result, the user's querying information can be exploited easily by these DNS servers and potential third-party eavesdroppers. In this section we explore the DNS privacy disclosure risks by three different scopes, namely, *recursors*, *authoritative DNS servers*, and *transmission channels*.

a) Recursors. Typically, there is no caching policy on the stub resolver's side, which means that all of the queries generated by the stub resolver would be sent to the recursor. In other words, the recursor can be able to observe and collect the user's all targeting information including each target's query volume. Since all users rely on the recursor for web surfing, so it's easy to do this kind of exploitation. The recursors could either utilize these data themselves, or they can pass the data to some third-party being part of a surveillance program like "*PRISM*" [6]. For example, some large hotels may use their controlled recursor to aggregate DNS data in order to extract information from their customers about what kind of website they always access while residing in their hotels. Moreover, the recursors can not only listen to DNS queries sent to it but also can actively drop, forge or manipulate DNS responses. For example, the recursor controlled by some ISP can simply block access to some particular website by dropping the user's all queries for its corresponding domain name.

b) Authoritative DNS Servers. Note that the DNS queries received by authoritative DNS servers are originated from recursors, not from users. Therefore, user's source IP address can be hidden from the authoritative DNS servers, which gives the user some certain degree of privacy. However, this hiding does not always work. For instance, many of today's recursors actively uses "*edns-client-subnet*" mechanism [7] to enable themselves to tag queries sent to authoritative DNS servers with the user's IP address, so that they can receive optimized responses for this specific user instead of the one for this recursor. In this case, the authoritative DNS servers will know the exact IP addresses of the users which can lead to serious user privacy exposure. Moreover, the authoritative DNS server can also observe and collect all of the incoming DNS queries just like the recursor. Although the caching mechanism used by the recursor hides the exact volume of DNS queries sent to the authoritative DNS servers, the authoritative servers can still infer expected query volume theoretically based on the distribution of queries' arrivals.

c) Transmission Channels. Typically, the DNS traffic is not encrypted and could be easily observed or injected by eavesdroppers and attackers. As a result, if the user starts a HTTPS communication with a website, while the HTTP traffic is encrypted, the DNS traffic prior to it will not be. Therefore, we identify two main risks during the transmission channels, namely, passive eavesdropping and active *MITM* attack. In the first case, the eavesdropper does not compromise any DNS servers, but only eavesdrops on the transmitted DNS data passively in order to learn source-target related information. Currently, the plaintext DNS data can be easily eavesdropped via unprotected Ethernet and Wi-Fi networks by using many tools available. In the second case, attackers can actively inject forged packets into the DNS traffic to launch MITM attacks. These injections can fool the users and redirect the DNS traffic to a malicious DNS server leading to compromise of the user's privacy. Note that the best vantage point to do the above privacy exploitation is clearly between the stub resolvers and the recursors, since the DNS traffic is not limited by DNS caching.

C. Requirements

Given the above DNS privacy disclosure risk analysis, in this section we detail the criterions of validity that are required for DNS privacy preservation.

a) Effectiveness. Note that current DNS privacy preserving techniques such as DNS data encryption and "*qname minimization*" can only solve the DNS privacy disclosure problems partially. However, an ideal DNS privacy preserving solution should try to be effective enough to avoid all kinds of privacy leaking risks as described above. Most importantly, the user's source-target information should not be disclosed during the whole DNS query process. In this context, any of the DNS servers (including recursive and authoritative) or third-party observers should not keep the ability to observe or infer the linkage between the user's source IP address and his/her targeting domain name information from any part of the DNS traffic.

b) Usability. Some aggressive solutions aiming to fully solve the DNS privacy disclosure problems such as *Namecoin* would cause fundamental changes to the current DNS, thus cannot be widely adopted by the community. Therefore, an ideal DNS privacy preserving solution should also try to be compatible enough with current DNS by avoiding significant changes to the current DNS. In other words, the DNS privacy preserving solution should provide a standards-compliant and lightweight interface that could be accessed easily by both users and DNS servers. For example, since most of current DNS queries are transmitted in UDP, it is clearly the best choice to make a DNS privacy preserving solution under UDP protocol. Furthermore, the introduction of the DNS privacy preserving solution should not lead to significant delays to the DNS query process either and the additional workload added to the DNS servers (if any) should be low.

From the above description, we can imply that an ideal DNS privacy preserving solution should not only be effective enough to avoid all of these privacy leaking risks, but should also show high usability in the current DNS. Unfortunately, none of current solutions for DNS privacy preservation could meet these requirements simultaneously.

IV. Proposed solution

In practice, the introduction of any privacy enhancing technology will inevitably lead to potential challenge and cost to the DNS. As such, an ideal DNS privacy preserving solution should follow a lightweight design way to avoid adversely impacting the existing DNS infrastructure or the user base. In this section, we introduce our initial idea for effective DNS privacy preservation and further validate its usability through real-world simulations.

A. Details

Generally, the recursor plays as an agent role between the users and the authoritative DNS servers, and thus has the ability to naturally access to all DNS query data involved in the whole DNS query process. Therefore, we believe that the key point for an effective DNS privacy preserving solution is to eliminate the recursor's ability of accessing to all DNS query data. In this context, the main idea of our proposed method for DNS privacy preservation is to introduce a new type of DNS servers called "*privacy preserving server*" into the DNS query process. Just like the recursors, the privacy preserving servers play as an agent role between the users and the authoritative DNS servers. As illustrated in Fig. 2, when the stub resolver wants to query a name "*www.example.com*", it first converts this name into an encrypted one (like "*e5sdn49imw*") by using the public key provided by its predefined privacy preserving server (such as "*privacy.cn*"). Then the stub resolver launches a DNS query for a new combined name instead towards the recursor (namely, "*e5sdn49imw.privacy.cn*"). When the privacy preserving server receives this DNS query from the recursor, it will decrypt this combined name by using the local private key, and response the corresponding reply from the authoritative DNS servers to the recursor, in an encrypted way. Finally, the stub resolver will receive the corresponding reply from the recursor and decrypted it by using the local public key.

From the above description, we can notice that none of these three types of DNS servers (or other third-party eavesdroppers) would keep the ability to observe user's origin-target information from any part of the DNS query process, and thus the user's privacy can be preserved very well. Meanwhile, all DNS data here can be transmitted by standards-compliant DNS packets (in UDP), and the whole DNS query process can be implemented by existing DNS protocols without any changes to the current recursive or authoritative. In a word, our solution can be expected to have remarkable effectiveness and usability in the current DNS.

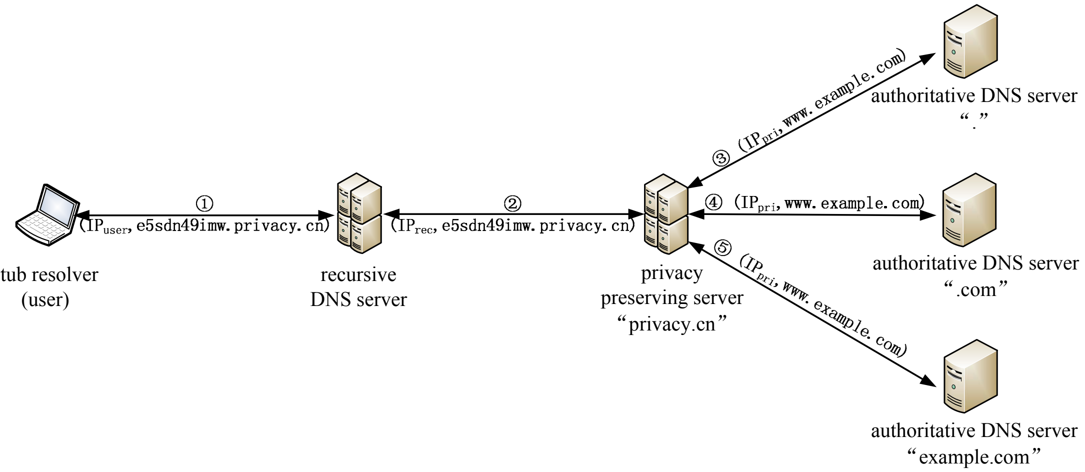


Fig. 2. DNS query process with privacy preservation.

B. Validation

Note that the proposed DNS privacy preserving solution does not cause any changes or additional workload to the current recursive or authoritative DNS servers. Nevertheless, the user's query latency can be increased inevitably due to the introducing of privacy preserving servers. Therefore, for practical concern, this kind of increase in user's query latency should be kept in a moderate level. Therefore, we need to analyze this kind of increase in user's query latency quantitatively through real world simulations, to further validate our solution's potential applicable prospective.

Our simulated testbed is built up within a local network (1 *Gbps*) and ready-made desktop PCs (*Intel Core i7-4710 octal cores, 2 GB RAM, Ubuntu 16.04.1*). We use network emulator *netem* to simulate real-world latencies between different components. The privacy preserving server is configured to be authoritative for all DNS queries generated by the stub resolvers. Since the query latencies between the privacy preserving server and the authoritative DNS servers will not be affected by our proposed solution, our focus here is simply to measure and compare the query latencies between the user and the privacy preserving server, with the traditional ones between the user and the recursor. We generate encrypted DNS queries from multiple stub resolvers by using locally deployed load generators towards a single privacy preserving server, and calculate the average query latency between the stub resolvers and the privacy preserving server which can be traced by the load generators.

We also validate our solution's performance in scalability by ranging the query rate from 1*k*~5*k* queries per second gradually. In practice, query latency may be one of the major concerns when the users are considering privacy preserving technologies for their DNS query process. Exaggerated query latency will definitely weaken the user's willing to adopt this solution. Simulation results show that the introduction of our privacy preserving servers into the DNS query process can only lead to moderate level of additional query latency to the users which is well kept in a moderate level (Fig. 3). Therefore, our propose solution is acceptable for the users and worthwhile for practical deployment.

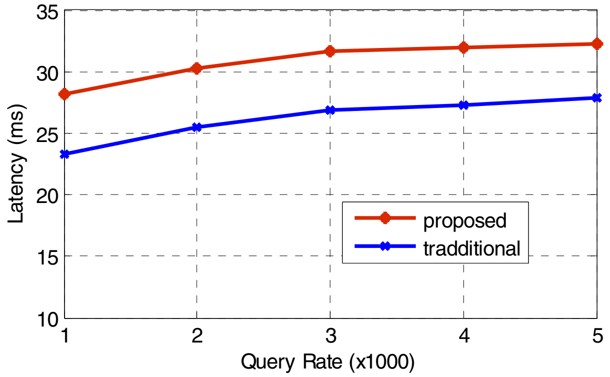


Fig. 3. Query latency comparison between the traditional DNS query process and the proposed one.

V. Conclusion and Future Work

While the DNS research community begins to be aware of the privacy issues in the DNS, it is virtually impossible to make significant progress quickly because of the need for compatibility with existing infrastructure. For example, the deployment rate of the security enhancing technology DNSSEC among world-wide DNS servers is still extremely low, even though it has been introduced for many years. In fact, even some tiny modification in the DNS might lead to serious negative impact to the DNS, and also somebody's business model or national interests.

Due to effectiveness or usability issues, previous work on DNS privacy preservation has not resulted in readily deployment into the DNS so far. In this paper, we first analyze the whole DNS query process and the privacy disclosure problems during every single step of the query, then we describe the requirements that an effective and usable DNS privacy preserving technology should meet. We further propose a simple DNS privacy preserving solution which can only lead to a moderate level of additional query latency to the users demonstrating significant applicable prospect in the current DNS infrastructure. We hope our work could be an initial step to address the DNS privacy preserving issues in a more effective and usable way. In the next step, we are aiming to seek for our solution's large scale deployment in the real world. We also propose to integrate our solution with DNSSEC technology to provide a more comprehensive one for authenticity, integrity and privacy protection of the DNS.

Acknowledgment

This work is partially supported by China's Key R&D Program (No. 2016YFC0801100) and Around Five Top Priorities of "One-Three-Five" Strategic Planning, Computer Network Information Center, Chinese Academy of Sciences (No. CNIC_PY-1402).

References

[1] L. Zhu, Z. Hu, J. Heidemann, D. Wessels, A. Mankin, and N. Somaiya, "Connection-oriented DNS to Improve Privacy and Security," in Proceedings of 2015 IEEE Symposium on Security and Privacy (S&P), pp. 171-186, 2015.

[2] IETF DNS Private Exchange (DPRIVE) Working Group, https://datatracker.ietf.org/wg/dprive/documents/, 2016.

[3] F. Zhao, Y. Hori, K. Sakurai, "Analysis of Privacy Disclosure in DNS Query," in Proceedings of International Conference on Multimedia and Ubiquitous Engineering, pp. 952-957, 2007.

[4] M. Wachs, M. Schanzenbach, and Christian Grotho, "A Censorship-Resistant, Privacy-enhancing and Fully Decentralized Name System," in Proceedings of the 13th International Conference on Cryptology and Network Security, pp. 127-142, 2014.

[5] Namecoin, https://en.wikipedia.org/wiki/Namecoin, 2016.

[6] B. Schneier, "NSA Targets the Privacy-Conscious for Surveillance," https://www.schneier.com/blog/archives/2014/07/nsa_targets_pri.html, 2014.

[7] C. Contavalli, W. van der Gaast, D. C. Lawrence, and Warren Kumari, "Client Subnet in DNS Queries," https://datatracker.ietf.org/doc/rfc7871/, 2016.