On the Performance and Analysis of DNS Security Extensions*

Reza Curtmola, Aniello Del Sorbo, and Giuseppe Ateniese

Information Security Institute and Department of Computer Science,
Johns Hopkins University, Baltimore, MD 21218, USA
{crix, anidel, ateniese}@cs.jhu.edu

Abstract. The Domain Name System (DNS) is an essential component of the critical infrastructure of the Internet. The role of DNS is vital, as it is involved in virtually every Internet transaction. It is sometimes remarked that DNS works well as it is now and any changes to it may disrupt its functionality and add complexity. However, due to its importance, an insecure DNS is unacceptable for current and future networks. The astonishing simplicity of mounting an attack against the DNS and the damaging potential of such an attack should convince practitioners and system administrators to employ a secure version of DNS. However, security comes with a cost. In this paper, we examine the performance of two proposals for secure DNS and we discuss the advantages and disadvantages of both. In particular, we analyze the impact that security measures have on the performance of DNS. While it is clear that adding security will lower DNS performance, our results show that the impact of security can be mitigated by deploying different security extensions at different levels in the DNS tree.

We also describe the first implementation of the SK-DNSSEC [1] protocol. The code is freely downloadable and released under an open-source license.

1 Introduction

The Domain Name System (DNS) is one of the world's largest distributed databases, whose main function is to translate human readable *domain names* to their corresponding IP *addresses*. Its tree-like structure allows a hierarchical distribution of domain names that facilitates fast name resolution and sub-division of the management load for domain administrators. The role of DNS is vital as it is involved in virtually every Internet transaction. Considering the importance of DNS, it is surprising that a secure version of it is not currently deployed. Vulnerabilities in the DNS system were noticed as early as 1990, in the seminal paper by Bellovin [2]. Several known threats to the DNS system are summarized in [3], some of which include packet interception, packet ID guessing, query prediction and cache poisoning. Because the DNS packets are not cryptographically

* The full version of the paper is available on the authors' website.

Y.G. Desmedt et al. (Eds.): CANS 2005, LNCS 3810, pp. 288–303, 2005.

signed, it is possible for a malicious party to inject, intercept or modify these packets with the intent of disrupting the DNS service [2, 3, 4, 5].

To have a secure DNS, two security requirements have to be met at a minimum: *Data origin authentication* and *data integrity*. The main proposal to secure the existing DNS is based mostly on public-key cryptography (PK-DNSSEC [6]), has received a lot of attention and exists as an IETF standard. A different solution (SK-DNSSEC [1]) makes use almost exclusively of symmetric-key cryptography.

This work presents the first implementation of the SK-DNSSEC protocol, which allows us to compare its performance with plain-DNS and PK-DNSSEC. We evaluated the performance tradeoff induced by the security overhead and identified the advantages and disadvantages of both security extensions. With regard to the computational cost, we show that PK-DNSSEC outperforms SK-DNSSEC for authoritative and referral name servers, while SK-DNSSEC performs better for recursive name servers. We argue that a hybrid approach with PK-DNSSEC deployed for top-level domains, where the information is static, and SK-DNSSEC for low-level domains, where the information is more dynamic, would leverage the benefits of both worlds.

Our experiments also show that PK-DNSSEC generates considerably more network traffic and has higher query latency than plain-DNS or SK-DNSSEC. Furthermore, SK-DNSSEC exhibits several other advantages over PK-DNSSEC, some of which are: it has simpler key management, it is less intrusive for zone files and it uses less memory for caching. All these aspects make SK-DNSSEC a valid alternative to PK-DNSSEC, especially if DNS security is needed in dynamic environments.

The rest of this paper is structured as follows. We review background and related work in Sect. 2. We present some details of the SK-DNSSEC implementation in Sect. 3. In Sect. 4 we empirically evaluate the performance of plain-DNS, SK-DNSSEC and PK-DNSSEC and conduct a comparative analysis of these three models. In Sect. 5 we discuss several aspects that can have a significant impact on the functionality of a secure DNS. Section 6 concludes the paper.

2 Background and Related Work

A *zone* is a part of the domain name space and the name server that manages a zone is called *authoritative* for that zone. The basic data unit in a zone is called a *Resource Record (RR)*. Clients that query name servers are called *resolvers*. The process by which resolvers retrieve data on a domain name is called *resolution*, and it usually involves a series of queries to servers along the path from the root node to the target name. A *recursive (caching)* name server, upon receiving a query, will resolve the query, cache it and return the answer. A *referral* name server does not return a final answer, but rather does a referral, meaning it redirects the query to the next name server in the DNS tree on the path to the server authoritative for the queried name.