DNS Protection Against Spoofing and Poisoning Attacks

Mohammed Abdulridha Hussain[1,2], Hai Jin[1], Zaid Alaa Hussien[1,3],
Zaid Ameen Abduljabbar[1,2], Salah H. Abbdal[1], Ayad Ibrahim[2]
[1]Cluster and Grid Computing Lab
Services Computing Technology and System Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, 430074, China
[2]University of Basrah, Basrah, Iraq
[3]Southern Technical University, Basrah, Iraq
mohsubber@gmail.com, hjin@hust.edu.cn

Abstract— **Domain name system is among the core part of TCP/IP protocol suite and the standard protocol used by the Internet. The domain name system consists of mapped website names with Internet protocol, which facilitates browsing by not requiring users to remember numeric notation addresses. The nature of the system, which involves transferring information in plain text, makes it vulnerable to security attacks. The domain name system suffers from spoofing and cache poisoning attacks that are intended to steal the private information of users. In this paper, a scheme is proposed to prevent the aforementioned attacks by using an asymmetric cipher to encrypt the important information in messages and to protect these messages from manipulation. The proposed scheme is examined and implemented using Linux platform and C programming language. The proposed scheme protects DNS against spoofing and poisoning attacks while the results show small fraction of delay in time comparing with the applied DNS. There are also additional commercial benefits since it does not result in additional costs.**

Keywords- DNS; DNS-spoofing; DNS cache poisoning; DNS protection

I. INTRODUCTION

Internet-based hosts and entities are identified through their *Internet Protocol* (IP) addresses. However, users prefer to use *Uniform Resource Locators* (URL) or host names instead of numeric addresses. Therefore, raising the needs for a system to translate host names to IP address, which is became *Domain Name System* (DNS) [1].

The DNS is designed as a client-server application. The DNS client-side application, which is called a resolver, receives the URL from the browser and sends a mapped request query to the DNS server. The complete domain name cannot be stored on a single server; thus, the DNS server is divided into a hierarchy of servers. This hierarchy can consist of many levels, but it generally includes three levels: root, zone, and local DNS servers. Moreover, when the local DNS server cannot respond to the resolver query, the local server creates a query for the zone DNS server. If the zone does not possess the answer, then a query for the root DNS server is created, and when the response is received, the answer is stored in the DNS server cache for future use. The types of DNS messages are described briefly as follows [2]:

- DNS Query: This is created by the resolver and it contains an *identifier* (ID). The ID differs for each message and port number, and it contains the *Name Server* (NS).
- DNS Response: This is created by the server and it contains the same ID to identify the query, NS, port number, and IP.

The DNS messages can be transferred without any encryption or authentication mechanisms, thereby raising the risk of attacks. The DNS in general suffers from spoofing and cache poisoning attacks that are intended to redirect client traffic to an attacker machine or to a fake website.

Several enhancements have been made to enable the DNS to resist and obstruct the aforementioned attacks. Simply stated, one example is randomly creating the DNS message ID and port number [3], which delays, reduces, and resists the attacker. However, with current Internet speed and bandwidth, even random IDs and port numbers can be predicted easily. The 0x20-bit encoding method [4] is intended to resist attackers by randomly mixing the domain name in the DNS messages and changing character cases. The objective of the attacker is to guess the valid domain name by using the correct cases of characters. The main drawback of the 0x20-bit encoding method is that the attacker can easily guess the domain name if it is short.

With current processor speeds, deriving the correct domain name is not so difficult. Thus, certain algorithms are used to address security concerns. One of them is DNSSEC [5], a set of cryptography algorithms that can prevent DNS attacks. *Public key infrastructure* (PKI), such as RSA, DSA, and Diffie–Hellman, are used in DNSSEC to create a digital signature to authenticate the sender of DNS messages. The digital signature is an encrypted hash value using the private key. However, DNSSEC has the following disadvantages: first, it is not deployed although it is an Internet standard; second, it causes an increase in traffic when the servers share certificates and keys; third, the scheme is complicated and is therefore time consuming and changing in the DNS protocol in the Internet infrastructure.

The objective of this paper is to present a scheme to protect the DNS from spoofing and cache poisoning attacks. To achieve this objective, PKI is used in encrypting the valuable fields in the DNS messages while DNSSEC is used by PKI to sign the messages. The proposed scheme is tested

978-1-5090-2535-0/16 $31.00 © 2016 IEEE
DOI 10.1109/ICISCE.2016.279