Hindawi
Security and Communication Networks
Volume 2017, Article ID 9479476, 15 pages
https://doi.org/10.1155/2017/9479476

Research Article

Enc-DNS-HTTP: Utilising DNS Infrastructure to Secure Web Browsing

Mohammed Abdulridha Hussain,[1,2] Hai Jin,[1] Zaid Alaa Hussien,[1,3] Zaid Ameen Abduljabbar,[1,2] Salah H. Abbdal,[1] and 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*

Correspondence should be addressed to Hai Jin; hjin@hust.edu.cn

Received 6 August 2016; Revised 31 October 2016; Accepted 15 November 2016; Published 3 April 2017

Academic Editor: Pascal Lorenz

Online information security is a major concern for both users and companies, since data transferred via the Internet is becoming increasingly sensitive. The World Wide Web uses *Hypertext Transfer Protocol* (HTTP) to transfer information and *Secure Sockets Layer* (SSL) to secure the connection between clients and servers. However, *Hypertext Transfer Protocol Secure* (HTTPS) is vulnerable to attacks that threaten the privacy of information sent between clients and servers. In this paper, we propose Enc-DNS-HTTP for securing client requests, protecting server responses, and withstanding HTTPS attacks. Enc-DNS-HTTP is based on the distribution of a web server public key, which is transferred via a secure communication between client and a *Domain Name System* (DNS) server. This key is used to encrypt client-server communication. The scheme is implemented in the C programming language and tested on a Linux platform. In comparison with Apache HTTPS, this scheme is shown to have more effective resistance to attacks and improved performance since it does not involve a high number of time-consuming operations.

1. Introduction

Digital information and electronic services are delivered to users through the Internet. Information and services are often sensitive, particularly in applications such as online banking, which requires secure transactions [1]. Services are provided through a web server, and the user contacts the server by using an Internet browser such as Internet Explorer (IE) or Google Chrome. The client and server communicate using *Hypertext Transfer Protocol* (HTTP).

Modern web security relies on a combination of *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) with HTTP, and this is known as *Hypertext Transfer Protocol Secure* (HTTPS). However, almost all browsers and servers apply SSL/TLS to secure information transactions [2].

SSL uses server certificates to publish the public keys of server; each of these certificates is signed by a trusted third party, known as the *Certificate Authority* (CA). SSL protocol consists of two phases, the handshake phase and the data transfer phase, and the details of these are discussed further in Section 2. The handshake phase includes the sharing of both the server certificate and a symmetric algorithm key. Handshake messages are sent in plain text until the server successfully transmits the certificate to the client. In this context, data transfer is protected by a symmetric cipher.

These plain text messages in the handshake phase are particularly targeted by attackers, threatening HTTPS security. Attackers exploit a loophole by impersonating the web server and stealing user information; this approach capitalises on the user's habit of typing a *Uniform Resource Locator* (URL) without specifying HTTPS. Attackers deceive the client by making it appear that the web server is using HTTP without security. Such attacks are known as *Man-in-the-Middle* (MITM) sniffing and stripping attacks.