Solutions to Selected Problems

Guide to Internet Cryptography

Companion Material

February 8, 2026

Preface

This document provides solutions to selected problems from the book *Guide to Internet Cryptography: Security Protocols and Real-World Attack Implications*. The material is intended for educational use in courses and self-study.

Book website: https://link.springer.com/book/10.1007/978-3-031-19439-9

1 Chapter 9: Security of HTTP

Problem 9.1 HTTP

1. Consider the URL `https://www.google.com/search?&q=http`.
   - Can you name the different components?
   - When you enter this URL in your web browser and press return – in what order are the URL components processed?
   - Which parts of the URL are transmitted in the GET request line, which in the `Host:` header?
2. Take a look at the source code of the page retrieved above: Roughly how many HTTP requests does a browser have to make in total to display the page completely, including all images, etc.?

Solution

1. Consider the URL `https://www.google.com/search?&q=http`.
   - The different components are:
     (a) `https`: Protocol identifier.
     (b) `www.google.com`: Domain Name.
     (c) `/search`: Path.
     (d) `&p=http`: Query string.
   - The URL components are processed in the following order: (b) is used to query the IP address to set up a TCP connection; (a) is used to determine that a TLS handshale must be performed; (a) is used to format an HTTP request; (c) and (d) are included in this HTTP request.
   - (c) and (d) are transmitted in the GET request line, (b) in the `Host:` header.

2. The nswer to this question may vary. When opened in Google Chrome on February 8th, 2026, the source code of the result page contained 1,941 times the string `http`. When using the built-in developer tools, 148 HTTP requests were counted.

Problem 9.2 HTTP Basic and Digest Authentication

1. Which encryption mechanism is used for HTTP Basic Authentication?
2. How are username and password transmitted with HTTP Digest Authentication?
3. Why does the use of HTTP Basic Authentication protect against CSRF attacks? Why can't an attacker generate the appropriate headers with XMLHttpRequest?

Solution

1. None. Both unsername and password are only encoded with Base64, not encrypted.
2. Only the username is transmitted, the password is used to compute the digest value.
3. In a CSRF attack, the attacker lures the victim to open the attacker's web page with his browser. Then the victim's browser is tricked to send various HTTPS requests to different websites. If no CSRF protection is in place, we must distinguish the following cases:
   - The victim is logged in to the target, e.g., via a session cookie. In this case, the `Cookie:` header is always filled with the correct value, and the CSRF attack is sucessful.
   - The victim is *not* logged in, and the login mechanism is sending username and password via an HTML form. In this case, the attacker can copy this HTML form in his malicious web page, fill in passwords from a small dictionary, and test many of these password guesses against the target server.
   - The victim is *not* logged in, and the login mechanism is sending username and password via Basic Authentication. The values for username and password are queried from the user through means not accessible to the attacker (e.g., a pop-up window not contained in the DOM). Since the attacker can not directly set the `Authenticate:` header, he cannot test his password dictionary.