Web and Mobile Application Security

Chapter 6 — Session Tracking and Authentication

Mr Laïdi FOUGHALI

l.foughali@univ-skikda.dz

(Website ⇒ www.al-moualime.com)

University of Skikda — Computer Science Department

2nd Year Master in Information Security (IS)

Academic Year 2026-2027

Updated on 2026-09-22 at 23:41:35

Outline

Chapter objectives

By the end of this chapter, you will be able to

- Know how a site is authenticated
- Understand state management between client and server
- Master cookie security and session integrity

6.1 — How a website is authenticated

The TLS certificate, foundation of trust

The browser checks that the presented certificate is:

- Valid and not expired
- Issued for the visited domain
- Signed by a trusted certificate authority

The padlock reflects the success of these checks (chapter 4).

On the user's side

The server authenticates the user via their credentials, possibly strengthened with multi-factor authentication (MFA).

6.2 — Tracking state between client and server

HTTP is a stateless protocol

Each request is independent: a mechanism is needed to recognize a user across requests.

The session

An identifier sent with every exchange, carried by a **session cookie** or a **token** (e.g. JWT). A secure mechanism requires:

- Unpredictable identifier, high entropy
- Encrypted transmission, limited lifetime
- Regeneration after authentication, invalidation on logout

6.3 — Cookies and session integrity

Cookie security attributes

- **Secure** : sent only over HTTPS
- **HttpOnly** : inaccessible to JavaScript (limits theft via XSS)
- **SameSite** : restricts cross-site sending (mitigates CSRF)
- **Domain / Path / Expiration** : scope and validity period

Threats to session integrity

Hijacking (stealing the identifier) and **fixation** (attacker-imposed identifier).

Key takeaways

The essentials of this chapter

- A site is authenticated by its TLS certificate
- Since HTTP is stateless, the session relies on an unpredictable, protected identifier
- Secure, HttpOnly, and SameSite are essential for cookie security
- Hijacking and fixation are the main threats to session integrity

Questions and exercises

To practice

1. Explain the respective roles of the Secure, HttpOnly, and SameSite attributes.
2. Since HTTP is stateless, how does a server recognize a logged-in user?
3. Differentiate session hijacking from session fixation.
4. Why must the session identifier be regenerated after authentication?