Web and Mobile Application Security

Introduction

Mr Laïdi FOUGHALI

l.foughali@univ-skikda.dz

(Website ⇒ www.al-moualime.com)



University of Skikda — Department of Computer Science

2nd Year Master in Computer Security (CS)

Academic Year 2026-2027

Updated on 2026-10-25 at 18:56:48



© 2026 Mr Laïdi FOUGHALI — www.al-moualime.com/wmas — Creative Commons License — BY-NC-SA 4.0 International

Outline

1. Objectives
2. The stakes
3. Organization

Module description sheet

Official information

- **Master's degree** : Computer Security
- **Semester** : S3 (M2)
- **Course unit** : UEF322 (*Core Course Unit*)
- **Credits** : 4 **Coefficient** : 3
- **Recommended prerequisites** : Network security and digital vulnerability analysis
- **Assessment** : Continuous assessment and exam

Main reference

The Tangled Web: A Guide to Securing Modern Web Applications
Michał Zalewski (2011) — No Starch Press

Course motto

This course was designed around an approach centered on pedagogical effectiveness, active understanding, and the practical application of knowledge.

Teaching objectives

What this module aims for

Understand the fundamental principles and concepts of secure Web browsing, Web development architecture, the main vulnerabilities and attacks specific to the Web, as well as the mechanisms and best practices for developing and configuring Web applications.

Mobile component

Understand the role of encryption in the security of mobile applications and devices, and describe common encryption use cases.

Course progression

Chapters 1 through 7 cover Web application security — from attack methods to Web services — while chapter 8 is dedicated to mobile security. Each chapter includes objectives, a detailed treatment, key takeaways, and application exercises.

Course vision: think like an attacker, build like a defender

Three core skills to master

1. **Understand — the attacker's model**
   - Know how the Web and mobile actually work
   - Identify an application's attack surface
   - Reason in terms of threats and attack vectors
2. **Secure — design and code defensively**
   - Validate inputs, escape outputs
   - Robust authentication, sessions, and access control
   - Encrypted communications and secure storage
3. **Audit — test and fix**
   - Search for vulnerabilities (Web and mobile)
   - Exploit in a controlled environment to understand impact
   - Write a report and propose fixes

Why study this subject?

Everything runs through the Web and mobile

Today, almost every service relies on an application:

- Your bank → a Web + mobile application
- Your medical record → a health application
- Your purchases → an e-commerce platform
- Your conversations → messaging apps and social networks

Result: the application is now attackers' #1 entry point.

The target has shifted

- The network perimeter is increasingly well protected.
- Applications, on the other hand, directly expose data.
- The browser and the smartphone have become the new battlefield.

The CIA model applied to applications

The CIA model — Foundation of application security

C — Confidentiality

- Prevent unauthorized access to user data
- Examples: HTTPS/TLS, authentication, access control
- Risk: leakage of personal data (XSS, injection)

I — Integrity

- Guarantee that data and actions are not tampered with
- Examples: anti-CSRF tokens, signatures, input validation
- Risk: actions performed without the user's knowledge

A — Availability

- Ensure the service remains accessible to legitimate users
- Examples: rate limiting, anti-DDoS protection
- Risk: disruption of a critical service

The foundation of the Web: the Same-Origin Policy

The browser security model

The browser runs code from sites that do not trust one another. Its central rule is the **Same-Origin Policy** :

- An origin = protocol + domain + port
- A script from one origin cannot read data from another origin
- Almost every Web flaw bypasses or abuses this boundary

Key idea of the course

Understanding **where this boundary lies** — and how attacks (XSS, CSRF, origin leaks) cross it — is the key to Web security.

The real cycle of an attack against an application

Typical stages observed in real incidents

1. Reconnaissance — mapping the application and its entry points
2. Vulnerability hunting — injection, XSS, weak authentication, outdated components
3. Exploitation — taking over a session or an account
4. Privilege escalation — gaining administrator access or internal APIs
5. Exfiltration — progressive theft of data
6. Persistence / cleanup — maintaining access, erasing traces

Link to the course

Understand → anticipate the vector Secure → close the door Audit → detect before the attacker does

The course's running scenario

Case study: an e-commerce platform and its mobile app

A company exposes:

- A Website (customer accounts, payment, cart)
- An API consumed by a mobile application
- A database of accounts, orders, and payment methods

An attacker chains an injection, a session theft, and API abuse.

Your mission throughout the course

- Web part: prevent injection, XSS, and CSRF
- Authentication part: secure sessions and tokens
- Mobile part: protect storage, communications, and APIs
- Audit part: test the application and produce a professional report

What data is at risk?

Three types of critical data

1. **Personal data**
   - Identity, contacts, messages, location
   - Browsing and purchase history
2. **Financial data**
   - Bank cards, accounts, transactions
3. **Technical secrets and tokens**
   - Session cookies, API tokens, keys

Why is this serious?

This data is resold, and a single stolen token can open access to thousands of accounts.

What you will learn (1/2)

Technical skills — Web

By the end of the course, you will be able to:
Secure a Web application

1. Prevent injections (SQL, commands, etc.)
2. Neutralize XSS (*Cross-Site Scripting*) and CSRF (*Cross-Site Request Forgery*)
3. Secure sessions, cookies, and authentication
4. Deploy HTTPS/TLS and a CSP (*Content Security Policy*)

Transferable skills

The principles learned apply to every development framework: PHP, Node.js, Django, Spring, Laravel, etc.

What you will learn (2/2)

Technical skills — Mobile and auditing

Secure and audit a mobile application

1. Protect local storage and communications (TLS, pinning)
2. Understand the Android / iOS permission model
3. Resist reverse engineering and secure APIs
4. Test an application and write an audit report

This course prepares you to

- Build applications that are secure by design
- Work as an application penetration tester
- Prepare for recognized certifications (OSCP Offensive Security Certified Professional, CEH Certified Ethical Hacker, GWAPT GIAC Web Application Penetration Tester)

The big security problem

Where do flaws concentrate?

Security effort is often poorly distributed:

- Heavy investment in network and infrastructure
- Much less in the application layer
- Yet it is the application that directly handles the data

The image to remember

It's like securing a building with:

- An armored gate (network firewall)
- Cameras everywhere (monitoring)
- But wide-open apartment doors (the vulnerable application)

Most breaches exploit an application-layer flaw.

The most common Web vulnerabilities

OWASP Top 10 — the major risk families

- Broken access control, which now includes SSRF
- Security misconfiguration
- Software supply chain failures
- Cryptographic failures
- Injection (SQL, commands, LDAP) and XSS
- Insecure design
- Authentication failures
- Software or data integrity failures
- Security logging and alerting failures
- Mishandling of exceptional conditions (errors, edge cases)

Source: OWASP Top 10:2025 — the worldwide reference for Web application security risks

Who attacks, and how?

Two attacker profiles

1. **External attackers**
   - Exploit injection, XSS, outdated components
   - Automate large-scale vulnerability hunting
2. **Insider threats and human error**
   - Misconfiguration, secrets exposed in code
   - Privilege abuse, negligence

The human factor

Phishing and social engineering remain involved in a large share of incidents.

Source: Verizon Data Breach Investigations Report (DBIR)

What changes on mobile

OWASP Mobile — risks specific to mobile

- Insecure data storage on the device
- Insecure communication (no TLS / no pinning)
- Weak authentication and authorization
- Insufficient cryptography and embedded secrets
- Insufficient binary protection (reverse engineering)
- Poor management of permissions and privacy

An expanded attack surface

The mobile application adds to the threats of the mobile network itself: eavesdropping, MITM (*Man-In-The-Middle*), denial of service.

Source: Survey on Threats and Attacks on Mobile Networks

How much does a cyberattack cost?

Average cost of a data breach

According to the **IBM Cost of a Data Breach** report:

- **Average global cost**: several million dollars per incident
- **Healthcare sector**: often the costliest
- Detection and containment: several months on average

Source: IBM Cost of a Data Breach Report (Ponemon Institute)

Why is it so expensive?

- Regulatory fines (GDPR, PCI-DSS)
- Loss of trust and customers
- Business disruption and remediation

Famous real-world cases

Three incidents that shaped application security

1. **Equifax (2017)**
   - About 147 million people affected
   - Cause: an unpatched Web component (Apache Struts)
2. **British Airways (2018)**
   - Theft of payment data from hundreds of thousands of customers
   - Cause: malicious client-side script injection (*Magecart*)
3. **Mobile health applications**
   - Sensitive data leaks from insecure storage and communications
   - Source: Security and Privacy Analysis of Mobile Health Applications

Testing to defend better

What is an application penetration test?

Same logic as an attacker, but with authorization and method:

- Map the application and its entry points
- Search for and exploit vulnerabilities
- Assess the real impact
- Document and propose fixes

Goal: find the flaws before the attacker does.

Ethical and legal framework

You only test systems you are **authorized** to test.
Any action outside that scope is illegal.

Course content overview (1/2)

Chapters 1 to 4

Chapter 1 — Vulnerabilities and Attack Methods

- Types of hackers, red/blue/purple teams
- Intrusion phases, zero-day flaws, malware, social engineering

Chapter 2 — Web Security Model

- The browser as an operating system
- Permissions, isolation, and the Same-Origin Policy

Chapter 3 — Web Application Security

- OWASP Top 10, injection, XSS, CSRF
- Web application protection techniques

Chapter 4 — Objectives and Problems of HTTPS

- SSL/TLS recap, hardening, certificates

Course content overview (2/2)

Chapters 5 to 8

Chapter 5 — Content Security Policy (CSP)

- Web workers, CSP, iframe isolation (sandbox)

Chapter 6 — Session Tracking and Authentication

- Authenticating a site, cookies, session integrity

Chapter 7 — XML and Web Services Security

- XXE, XML bombs, AJAX and its attacks

Chapter 8 — Mobile Security

- Mobile technologies, security model, encryption
- Network threats, privacy, best practices

Course organization

Lectures to understand the theory — 1.5h/week

- Each chapter = objectives, development, key points, exercises
- Analysis of real cases and standards (OWASP, GDPR)

Labs for hands-on practice — 1.5h/week

- Lab 1 — Reconnaissance and enumeration (Ch.1)
- Lab 2 — Same-Origin Policy and CORS (Ch.2)
- Lab 3 — SQL injection and XSS: exploit, then fix (Ch.3)
- Lab 4 — Auditing a website's TLS configuration (Ch.4)
- Lab 5 — Deploying and testing a CSP policy (Ch.5)

Assessment

Official assessment mode: **continuous assessment and exam**.

What you should already know (1/2)

Official prerequisite of the module

Network security and digital vulnerability analysis

- TCP/IP model, HTTP/HTTPS and DNS protocols
- Notions of vulnerability, scanning and flaw analysis

Useful Web knowledge

- Basics of HTML, CSS, JavaScript and the client-server model
- Use the Linux command line

What you should already know (2/2)

Basic security

- Understand Confidentiality, Integrity, Availability (CIA)
- Know what encryption and a certificate are
- Know common threats (phishing, malware)

If you have gaps

- Review the Web on **MDN Web Docs** (free)
- Practice on **OWASP Juice Shop** or **DVWA**
- Read the first chapters of the reference book

Which tools will you use? (1/2)

Web-side tools

- **Burp Suite / OWASP ZAP** : interception proxy and scanner
- **Browser developer tools** : inspecting requests/cookies
- **sqlmap** : automated SQL injection testing
- **OWASP Juice Shop / DVWA** : legal training targets

Why these tools?

- **Industry standards** : used by professional auditors
- **Free / open source** for the most part
- **Pedagogical** : you see what happens "on the wire"

Which tools will you use? (2/2)

Mobile-side tools

- **Android Studio / emulator** : test environment
- **MobSF** (*Mobile Security Framework*) : static and dynamic analysis
- **Frida / objection** : runtime instrumentation
- **Wireshark** : network traffic analysis

Do this right now

Install a browser with an interception proxy and a training target (Juice Shop) to practice at home.
Installation guide provided in the course resources.

Module bibliography (1/2)

Main reference

M. Zalewski, *The Tangled Web: A Guide to Securing Modern Web Applications*, 1st ed., No Starch Press, 2011.

Further references

- S. Mavoungou et al., *Survey on Threats and Attacks on Mobile Networks*, IEEE Access, 2016.
- A. Papageorgiou et al., *Security and Privacy Analysis of Mobile Health Applications*, IEEE Access, 2018.
- M. Oltrogge et al., *The Rise of the Citizen Developer*, IEEE S&P, 2018.

All 4 references are provided as PDFs in the course resources.

Where to get help? (2/2)

Online documentation (free!)

- OWASP (*Open Worldwide Application Security Project*) — Top 10, guides, and cheat sheets
- MDN Web Docs — Web and browser security reference
- OWASP MASVS / MASTG — mobile testing standard and guide

To stay up to date

- **PortSwigger Web Security Academy** — free labs
- **The Hacker News / Krebs on Security** — news
- **CVE** (*Common Vulnerabilities and Exposures*) — published vulnerabilities

Important laws and standards to know

- **GDPR** — personal data protection
- **PCI-DSS** (*Payment Card Industry Data Security Standard*) — payment card security
- Violation = fines of up to 4% of annual revenue!

What jobs come after this course? (1/2)

Career opportunities

- **Web / Mobile Pentester** — application penetration tester
- **Security Developer (AppSec)** — security-by-design code
- **Security Analyst** — incident detection and response
- **Security Auditor** — checks compliance and robustness
- **Bug Bounty Hunter** — rewarded vulnerability hunting

What jobs come after this course? (2/2)

Hiring sectors

- Banking, insurance, and fintech
- Healthcare and e-commerce
- Software vendors and cloud platforms
- Cybersecurity consulting firms

Job market

- Millions of unfilled cybersecurity positions worldwide
- Strong year-over-year growth in demand
- Among the **highest** salaries in IT
- AppSec specialists are in very high demand!

Closing thoughts

Welcome!

Why is this course exciting?

- You will learn to think like an attacker (to defend better)
- You will break, then fix, real applications
- You will protect critical data
- You will have an in-demand, well-paid career

The course philosophy

- **Understand** rather than memorize
- **Practice** rather than just read
- **Question** rather than accept
- **Keep learning** (security evolves every day)