Web Security – Presentation of the Course

Web Hacking

Fall Term 2026/27

Emmanuel Benoist | BFH-TI

Your Lecturer

Emmanuel Benoist

- email: `emmanuel.benoist(at)bfh.ch`
- PhD at the University of Caen (France)

Professor at the Berner Fachhochschule

- Teaching Computer Science in Biel since 1999
- Specialities: CS basics, operating systems, web programming and **web security**

Member of the ICE (Institute for Cybersecurity and Engineering)

- Web security
- Privacy protection on the web
- e-Health

The Course: Web Security

Goals of the course

- You know the most important dangers for web sites and their users
- You have an overview of the measures to take to build secure web applications
- You can name and explain the basic principles of web security
- You have **applied** these principles to real applications, both as an attacker and as a defender

Part of the module BTI 4203 – IT-Security 2.1, a.k.a. Web Security

- The other half of the module is taught by Kai Brünnler

Also offered as an optional module for ESC students of Unil

- Hacking of Web Sites BTI 7267

Programme of the Course

Ch.	Topic	Exercise application
1	A01 Broken Access Control	InvoiceDesk – an invoicing API
2	A02 Security Misconfiguration	ConfigBox – a deployed service
3	A03 Software Supply Chain Failures	RateWatch – a build pipeline
4	A04 Cryptographic Failures	VaultNotes – a notes service
5	A05 Injection	HelpDesk – four interpreters
6	A06–A10 of the Top 10	QuickWallet – five categories
7	Auditing and testing	– (methodology, no application)

- The six first chapters follow the **OWASP Top 10:2025**, in its own order of importance
- The last chapter ties them together: **how to audit** an application methodically and **how to report** what you found

Our Backbone: the OWASP Top 10:2025

One chapter each

1. Broken Access Control
2. Security Misconfiguration
3. Software Supply Chain Failures
4. Cryptographic Failures
5. Injection

Together in chapter 6

6. Insecure Design
7. Authentication Failures
8. Software or Data Integrity Failures
9. Security Logging and Alerting Failures
10. Mishandling of Exceptional Conditions

- The Top 10 is **not a standard** – it is an awareness document, built from data on hundreds of thousands of applications plus a community survey
- We use it as a **map**: every category is a family of bugs with its own attacks and its own defences

What Changed in 2025

- **Two new categories**
  - **A03 Software Supply Chain Failures**: ranked #1 by the community survey; it grew out of "Using Components with Known Vulnerabilities" and now covers the *whole* build-and-distribute chain
  - **A10 Mishandling of Exceptional Conditions**: what happens when things go wrong – bad error handling and, above all, **failing open**
- **Merged**: SSRF (A10:2021) is now part of **A01**
- **Renamed**: A07 Authentication Failures, A09 Security Logging **and Alerting** Failures
- **Moved**
  - Up: Security Misconfiguration (#5 → #2)
  - Down: Cryptographic Failures (#2 → #4), Injection (#3 → #5), Insecure Design (#4 → #6)
- **Unchanged**: Broken Access Control is #1, as in 2021

How a Lesson Works

Each chapter: a lecture and a hands-on exercise

- **Lecture**: the weakness, how it is exploited, real incidents, and the defences that actually work
- **Exercise** (90 minutes, **teams of two**): a small web application with *deliberately* planted weaknesses
  - **Attack** it, **find** the flaws in the code, **fix** them, and **write a test** so they stay fixed

What you get for each exercise

- A handout (PDF) and a ZIP archive with the sources, a `Dockerfile` and a test skeleton
- Runs with Python 3.12 in a virtual environment, or with Docker

Tools: an editor, a browser, and an intercepting proxy (Burp Suite Community or OWASP ZAP)

Rules of Engagement

- The exercise applications are **training applications** containing intentional weaknesses
  - Run them **on your own machine only**, bound to `127.0.0.1`
  - Never expose them to a network you do not control
- **Never test a system you are not explicitly authorised to test**
  - The techniques of this course, applied to third-party systems, are **illegal**: Swiss Criminal Code, art. 143^{bis} (unauthorised access to a data processing system)
  - An authorisation is a *written* scope, agreed before you start
- **Why we attack at all**: you cannot defend what you cannot break. Every attack in this course exists to make one defence concrete

Examination

Module BTI 4203 (Pa module)

- No intermediary test
- **The exercises are not graded**, and there is nothing to hand in – they are there for you to learn by doing

One written exam

- Two hours (120 minutes), during the examination weeks
- **Half of the time** for the web security part of the module
- Everything from the lectures *and* from the exercises may be asked

Material and Information

Moodle page of the course

- `https://moodle.bfh.ch/` – announcements and organisation

Web site of the course

- `http://www.benoist.ch/SoftSec/`
  - Slides, one deck per chapter
  - Exercises: the handout (PDF) and the application (ZIP)
  - Examples and resources

Questions

- During the lesson, or by email: `emmanuel.benoist(at)bfh.ch`

Bibliography

- **OWASP Top 10:2025** – the backbone of this course
  `https://top10.owasp.org/2025/`
- **OWASP Web Security Testing Guide (WSTG)** – the methodology of chapter 7
  `https://wstg.owasp.org/latest/`
- **OWASP Cheat Sheet Series** – the defences, one page per topic
  `https://cheatsheetseries.owasp.org/`
- **OWASP ASVS** – verification requirements, level by level
  `https://owasp.org/www-project-application-security-verification-standard/`
- **PortSwigger Web Security Academy** – free labs for every category
  `https://portswigger.net/web-security`
- *The Web Application Hacker's Handbook*, D. Stuttard and M. Pinto, Wiley
- *Web Security Testing Cookbook*, P. Hope and B. Walther, O'Reilly

And Who Are You?

Proficiency in web programming?

Proficiency in security?

German speaking / French speaking?

Type of study?

Have you ever found a vulnerability – on purpose or by accident?

Questions?

?