You've Been Hacked

An (Interactive) Course on Web Security

Paul Duplys

@duplys

December 25, 2020

0x0: Preliminaries

0x1: Web Security 101

0x2: Recon

0x3: Stateful Attacks

0x4: Attacks on Authentication

0x5: Cross-Site-Scripting (XSS)

0x6: SQL Injection

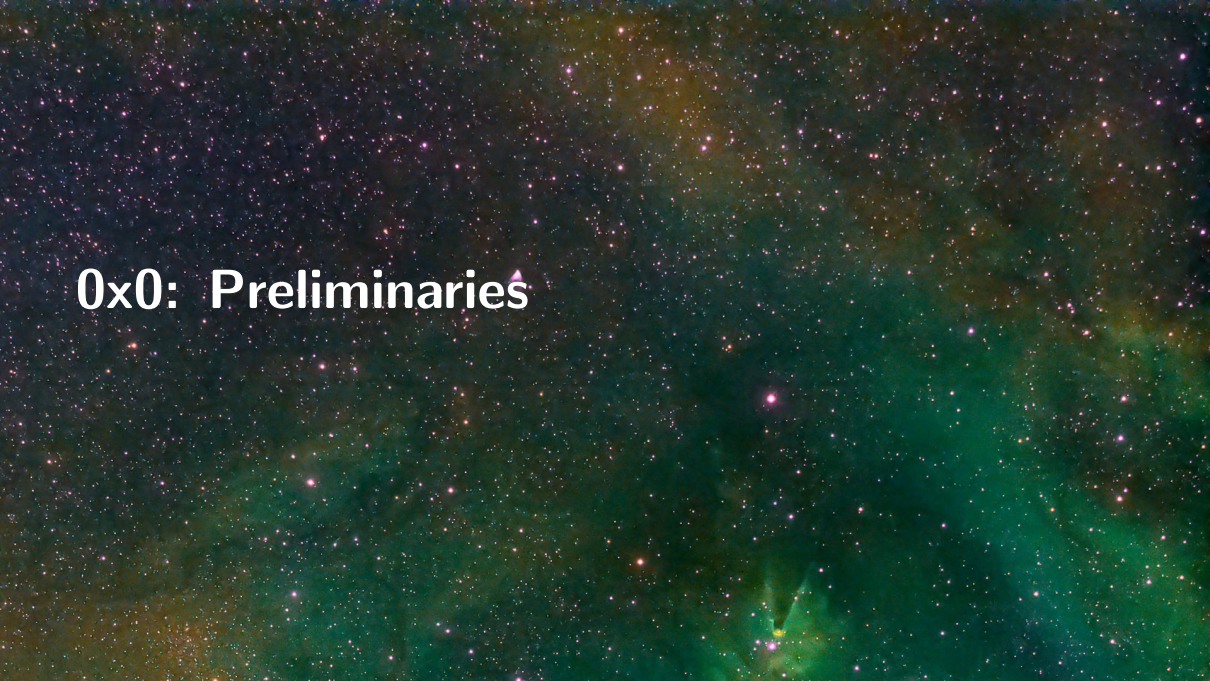
0x7: Other Injection-Based Vulnerabilities

0x8: Attacks on File Operations

0x9: Buffer Overflows, Format Strings and Integer Bugs

0xA: Architectural Attacks

0xB: Attacks on the Web Server


0x0: Preliminaries

whoami

short intro/bio.

man slides

(Interactive) course on web security based on Carsten Eiler's book "You've Been Hacked".
Who is the audience? How can I use the book? How can I explore the app?

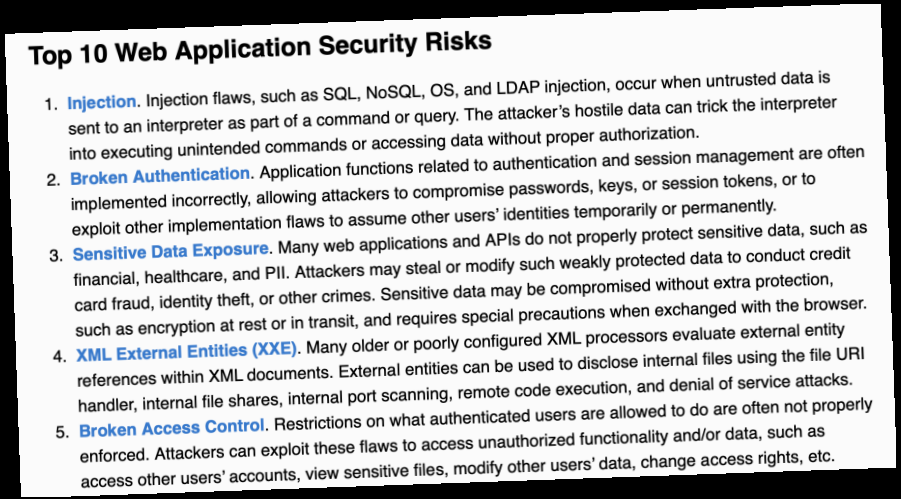
Where are the instruction located for how to build the Docker files and use the repository?

0x1: Web Security 101

In a nutshell, **to find vulnerabilities in your web application**, ...

1. ... test various values for parameters used by the web application and see what happens (conceptually similar to fuzzing)
2. ... check web application code for bugs that may lead to security vulnerabilities (typically missing checks of input values or missing countermeasures against certain types of attacks)

The Open Web Application Security Project (OWASP) maintains a list of Top 10 vulnerabilities in web applications.

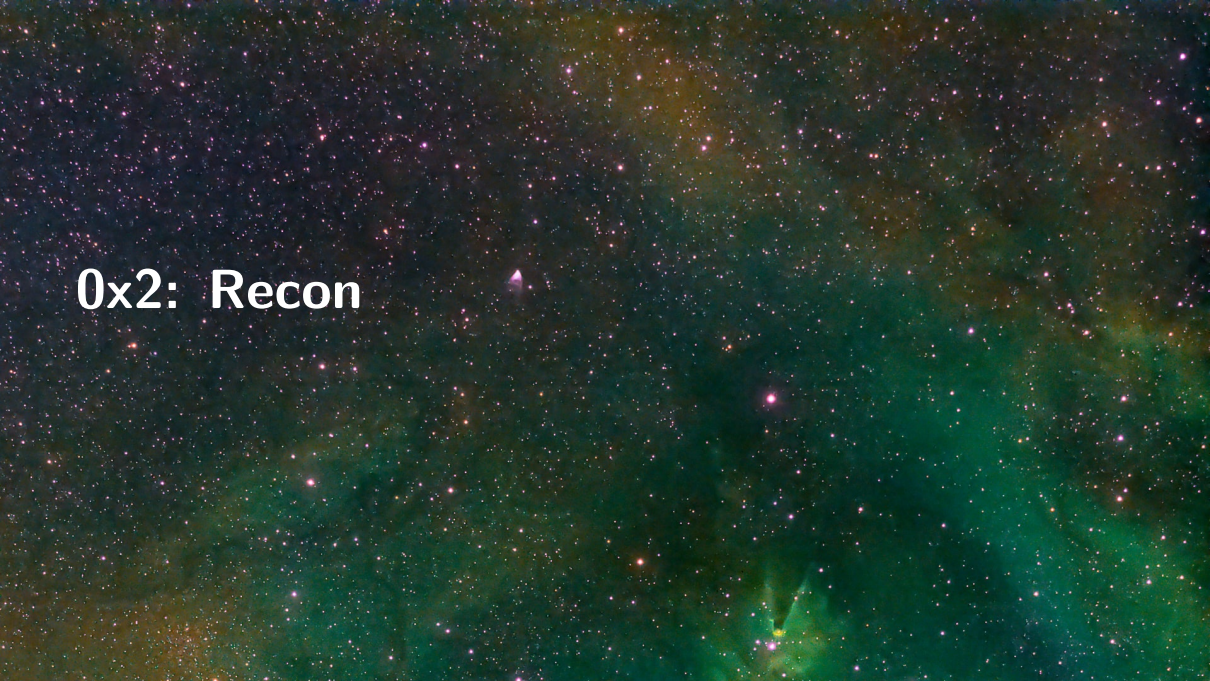


Top 10 Web Application Security Risks

1. **Injection**. Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
2. **Broken Authentication**. Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
3. **Sensitive Data Exposure**. Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
4. **XML External Entities (XXE)**. Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
5. **Broken Access Control**. Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

Fahrplan

- Get to know your target
- Test for stateful attacks
- Test for attacks on authentication
- Test for cross-site-scripting (XSS)
- Test for SQL injection
- Test for other injection-based vulnerabilities
- Test for attacks on file operations
- Test for buffer overflows, format strings and integer bugs
- Test for architectural attacks
- Test for attacks on the web server


0x2: Recon

reconnaissance: *n*. 1. Military observation of a region to locate an enemy or ascertain strategic features. 2. Preliminary surveying or research.

General Approach

To test a web application for security vulnerabilities, you must first get to know it. You must understand what functions are used and what parameters are used by these functions. You have to test every parameter whether it can be exploited (e.g., using illegitimate values). You also need to check whether the web application code contains known vulnerabilities.

Sample frame title

In this slide, some important text will be highlighted because it's important. Please, don't abuse it.

Remark
Sample text

Important theorem

Sample text in red box

Examples

Sample text in green box. The title of the block is "Examples".

Tools

- ZAProxy
- ...

Example

- one

Example

- one
- two

Example

- one
- two
- theorem

Example 1

One

Example 1

One Two

Example 1

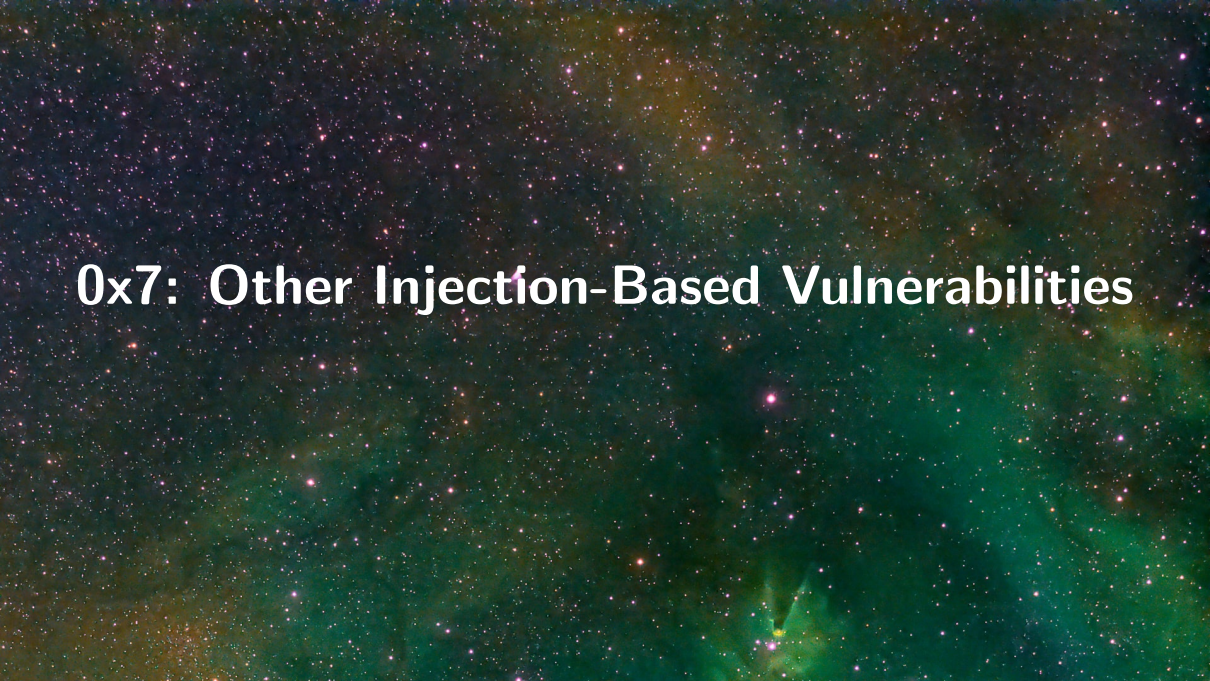
One Two Three


0x3: Stateful Attacks

0x4: Attacks on Authentication

0x5: Cross-Site-Scripting (XSS)

0x6: SQL Injection


0x7: Other Injection-Based Vulnerabilities

0x8: Attacks on File Operations

0x9: Buffer Overflows, Format Strings and Integer Bugs

0xA: Architectural Attacks

0xB: Attacks on the Web Server