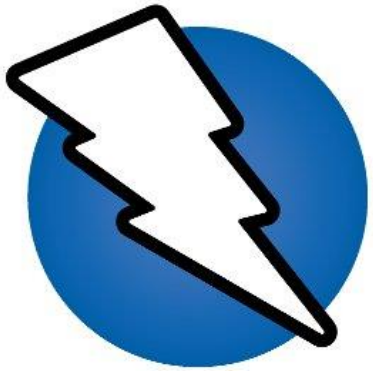
Dynamic Security Analyses

Alessandro Bocci
name.surname@phd.unipi.it

Advanced Software Engineering (Lab)
16/11/2023

What will you do?

- Learn some attacks to services with WebGoat
- Exploit browsers' developer tools
- Exploit OWASP ZAP to modify HTTP requests



Software Prerequisites

- OWASP ZAP (needs Java 11+)
- Firefox or Chrome Browser
- WebGoat docker image (`webgoat/webgoat`)

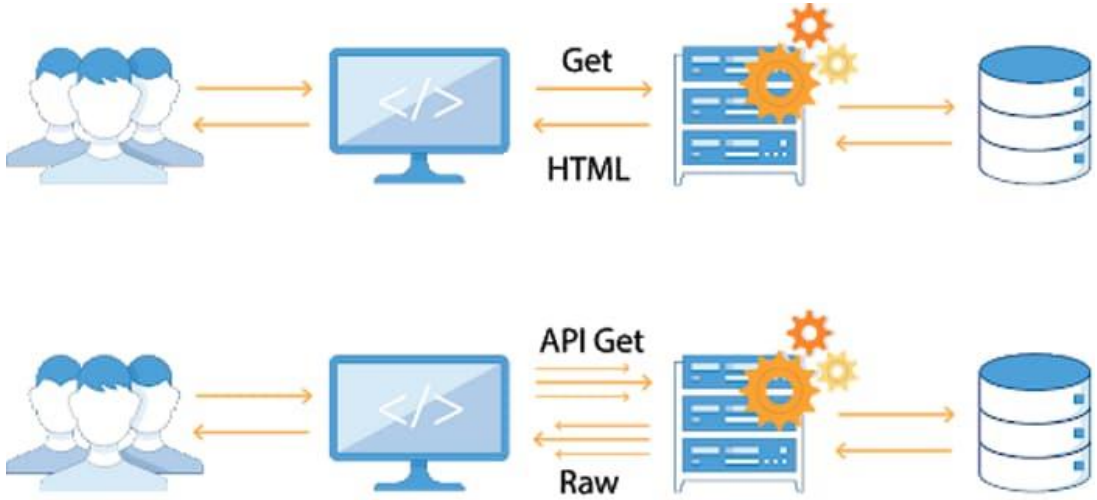
Secure coding practices

- Complexity of source code leads to more security vulnerabilities.
- Exponential increase of defects as number of lines of code increases.
- Functional and security testing is utterly important.

- Two types of testing:
  - Static (bandit-like)
  - Dynamic (what we see today ☺)

API-based applications

- Client devices are getting more powerful
- Logic moves from backend to frontend, i.e. servers are used more as data proxies – clients render data
- Clients consume raw data and maintain/monitor user's status
- Lots of parameters in http requests (e.g. objects ids, filters)
- API disclose s.t. about implementation



Top 10 API Vulnerabilities

- API expose microservices to consumers.
- It is therefore important to make them secure and avoid known pitfalls.

OWASP TOP10

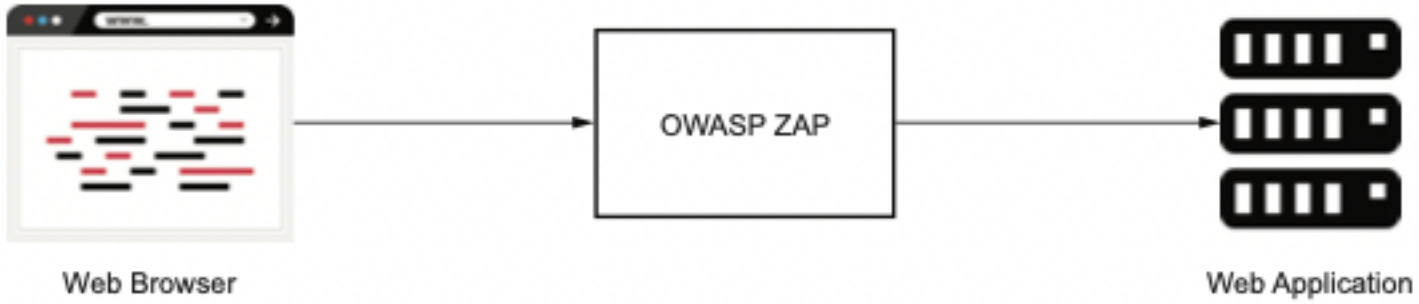
- Here is the top 10 list of API security vulnerabilities:
  1. Broken object-level authorization
  2. Broken authentication
  3. Excessive data exposure
  4. Lack of resources and rate limit
  5. Broken function-level authorisation
  6. Mass assignment
  7. Security misconfiguration
  8. Injection
  9. Improper asset management
  10. Insufficient logging and monitoring

https://apisecurity.io/encyclopedia/content/owasp-api-security-top-10-cheat-sheet-a4.pdf

Dynamic Analysis

- It checks code while it executes.
- It generates various types of input parameters to trigger as many execution flows as possible.

- OWASP ZAP is a tool for dynamic analysis that helps finding vulnerabilities in running web apps with penetration testing.
- ZAP acts as a proxy between the client app and the server.



WebGoat

- WebGoat is a deliberately insecure web application maintained by OWASP designed to teach web application security lessons.

- **WARNING 1:** *While running this program your machine will be extremely vulnerable to attack. You should disconnect from the Internet while using this program.*
- **WARNING 2:** *This program is for educational purposes only. If you attempt these techniques without authorization, you are very likely to get caught.*

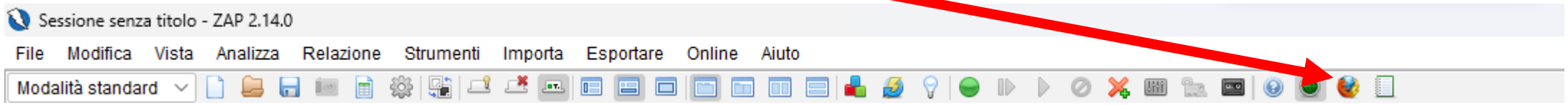
- After disconnecting from the Internet:

```
docker run -p 127.0.0.1:8081:8080 -p 127.0.0.1:9090:9090 webgoat/webgoat
```

Launch ZAP

- Select the configuration on the right and click ’Start’.

- Open your browser from ZAP



- Connect to http://localhost:8081/WebGoat
- Register as a new user and login.

Today's Lab

Resolve the following WebGoat excercises:

GENERAL

1. HTTP Basics.
2. HTTP Proxies
3. Developer Tools
4. CIA Triad

A1 - Broken Access Control

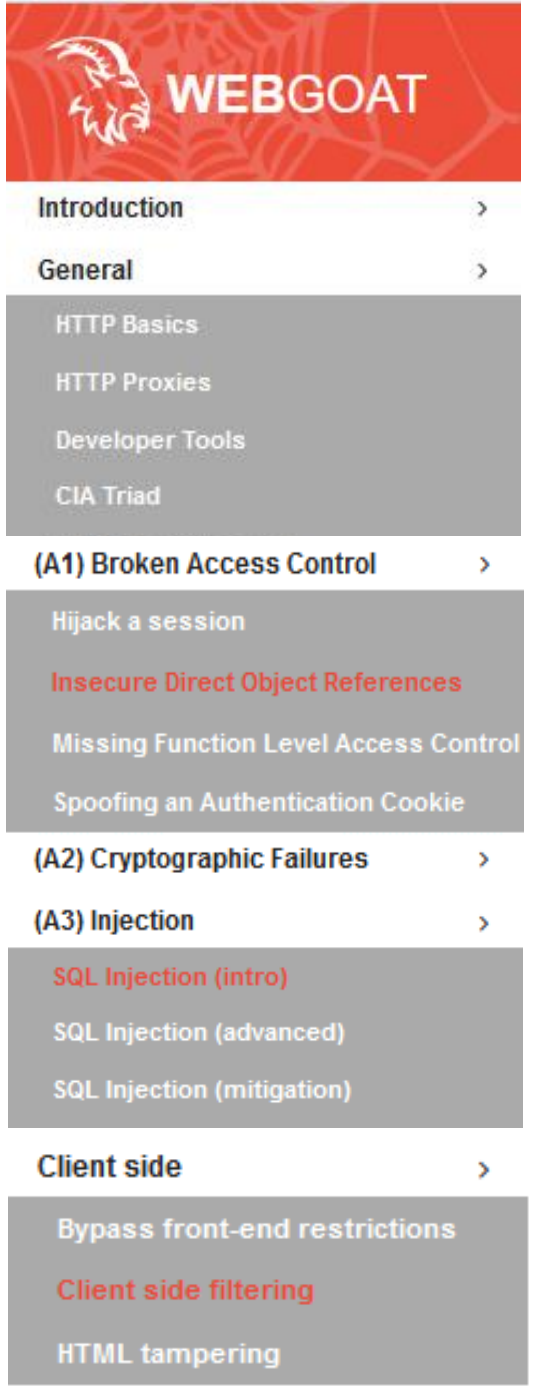
5. Insecure Direct Object References

A3 – Injection

6. SQL Injection (Intro)

Client side

7. Client side filtering



WebGoat Lessions

- Explanations on vulnerabilities (grey).
- Excercise(s) on how exploit them (red).
- Suggest how to mitigate them (grey).

Explanation\Suggestion

Passed Excercise

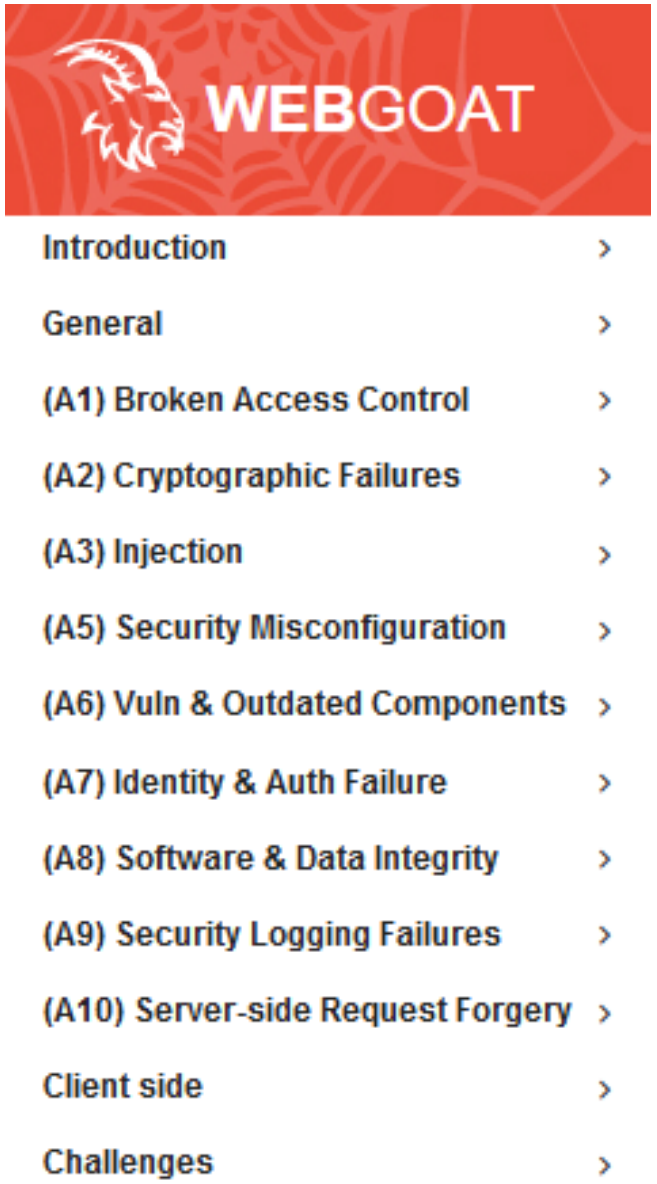
NotPassed Excercise

BONUS STAGE!

Bonus stage

You have plenty of excercises on WebGoat ☺

- Some of them needs you to code something, others needs a third party software (like we did with ZAP).
- Choose the ones more interesting for you and try them!
- Our Suggestion: A3 – Cross site scripting



Lab take away

- ❑ Familiarise with the most common vulnerabilities of services.
- ❑ Excercise on how to exploit such vulnerabilities.
- ❑ Learn how to avoid them.