Welcome!

Web Security: HTTP and Injections

Who am I?

- Juri Sacchetta
- PhD candidate, NECSTLab — Politecnico di Milano
- juri.sacchetta@polimi.it

The range

A fictional retail bank, deliberately broken, running for this session only

Platform bocconi.necst.it — register, pick a challenge, read the brief

Systems each challenge links to its own host under meridian.necst.it

Scoring every challenge hides a string shaped like flag{...}; paste it into the platform

Tools your browser is enough. Right-click, then Inspect

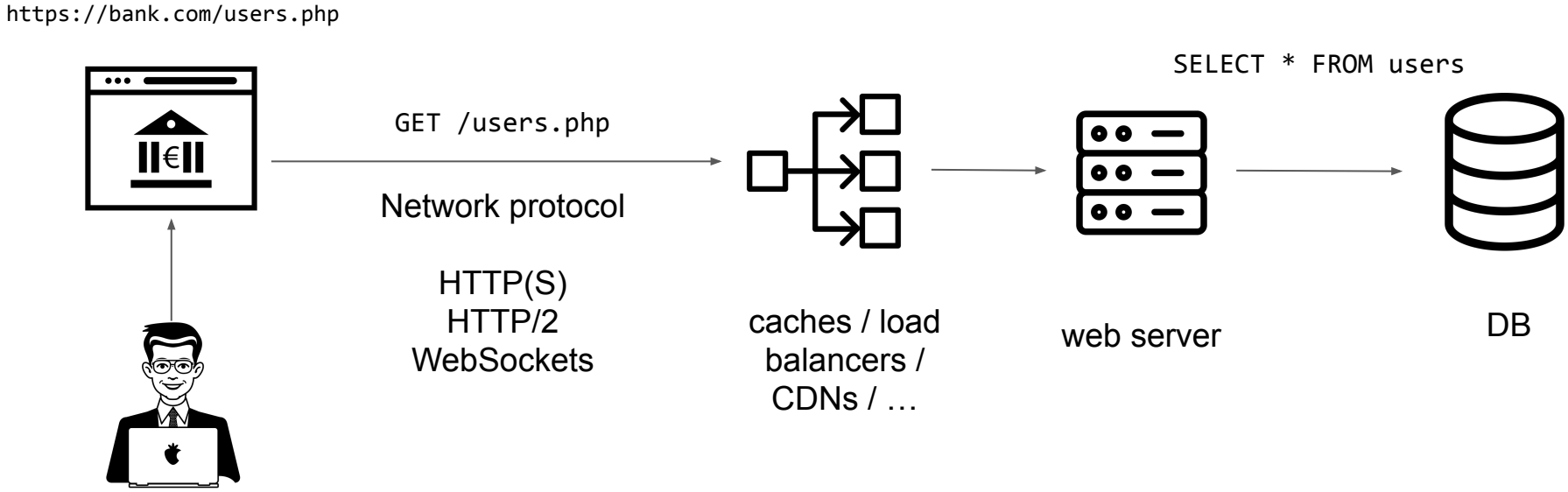
Rules attack only the hosts the platform links to. NeoBank is invented: no real institution, customer or account appears anywhere in here

POLITECNICO
MILANO 1863

UNIVERSITÀ COMMERCIALE
LUIGI BOCCONI MILANO

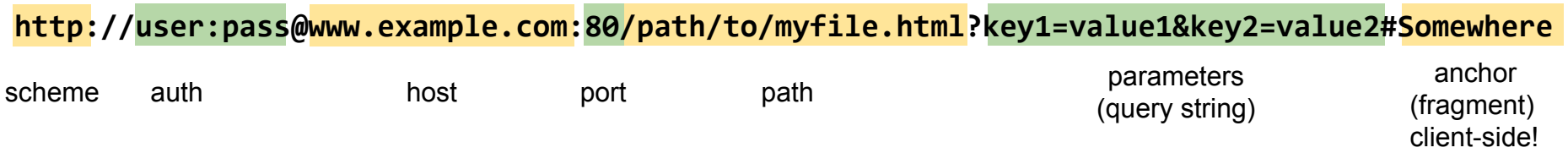
Web 101

How to Load a Webpage!



https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work
https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works

Resources are identified by a URL
What's in a URL (Uniform Resource Locator)?



1: https://tools.ietf.org/html/rfc1738

2: https://tools.ietf.org/html/rfc3986

https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL

How is a Website Made?

The main resources returned by the server following a request

When we make a request, the server responds by returning resources. On average, the structure of a website consists of **three fundamental macro-categories**:

HTML

Structure and Content

Represents the skeleton of the page. It defines the organization of text, links, images, and all structural elements that make up the document.

CSS

Style and Presentation

Manages the visual appearance. It determines colors, fonts, layouts, spacing, and ensures the page adapts correctly to any screen size (responsive).

JavaScript

Logic and Interactivity

Makes the page dynamic. It handles complex behaviors, user interaction, animations, asynchronous data updates, and responses to events.

Drawing a WebPage! (HTML)

```
1.  <html>
2.      <head>
3.          <title>Page Title</title>
4.      </head>
5.      <body>
6.          <h1>This is an HTML Page!</h1>
7.          <p>This is a paragraph</p>
8.          <a href="https://www.w3schools.com/html/">
9.              more about html
10.         </a>
11.     </body>
12. </html>
```

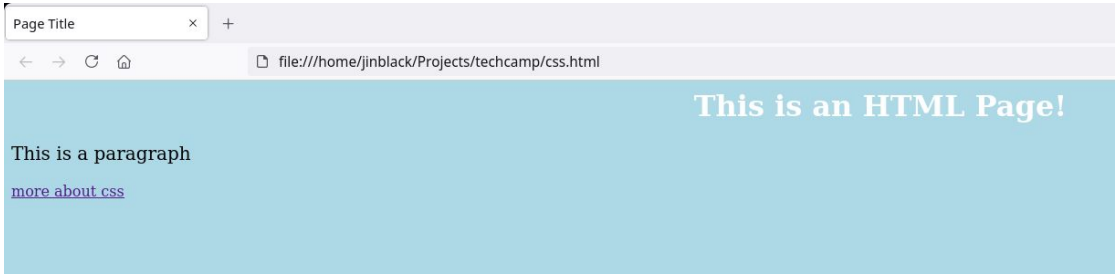
Drawing a Pretty WebPage! (CSS)

```html
<html>
    <head>
        <title>Page Title</title>
        <link rel="stylesheet" href="css.css">
    </head>
    <body>
        <h1>This is an HTML Page!</h1>
        <p>This is a paragraph</p>
        <a href="https://www.w3schools.com/css/">
            more about css
        </a>
    </body>
</html>
```

```css
body {
    background-color: lightblue;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: verdana;
    font-size: 20px;
}
```

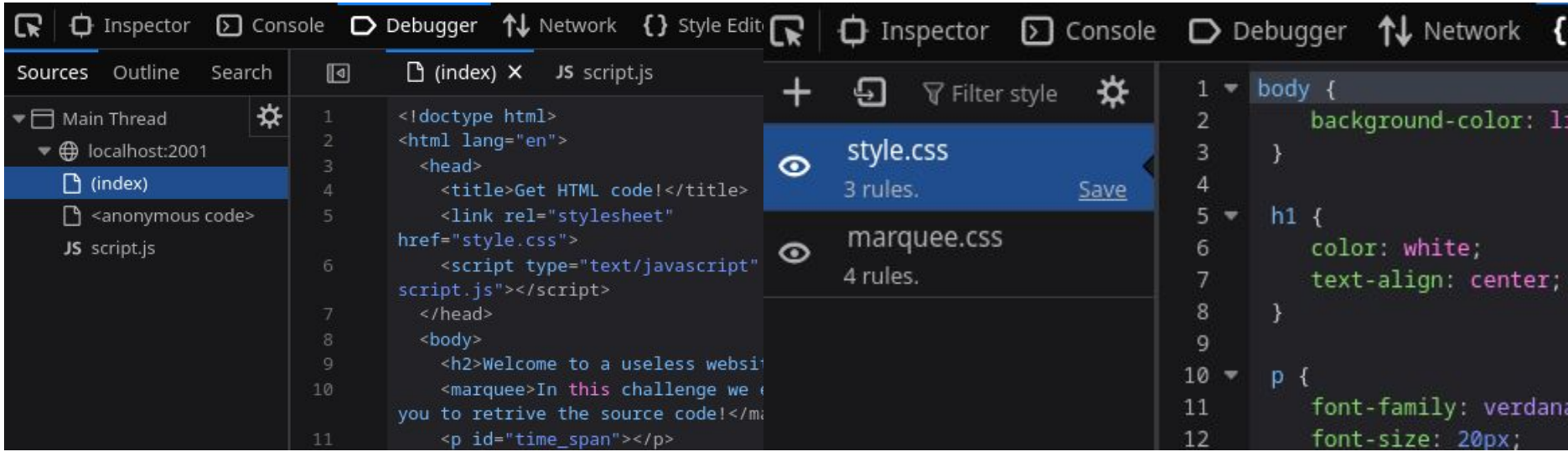


Dynamic WebPage! (JS)

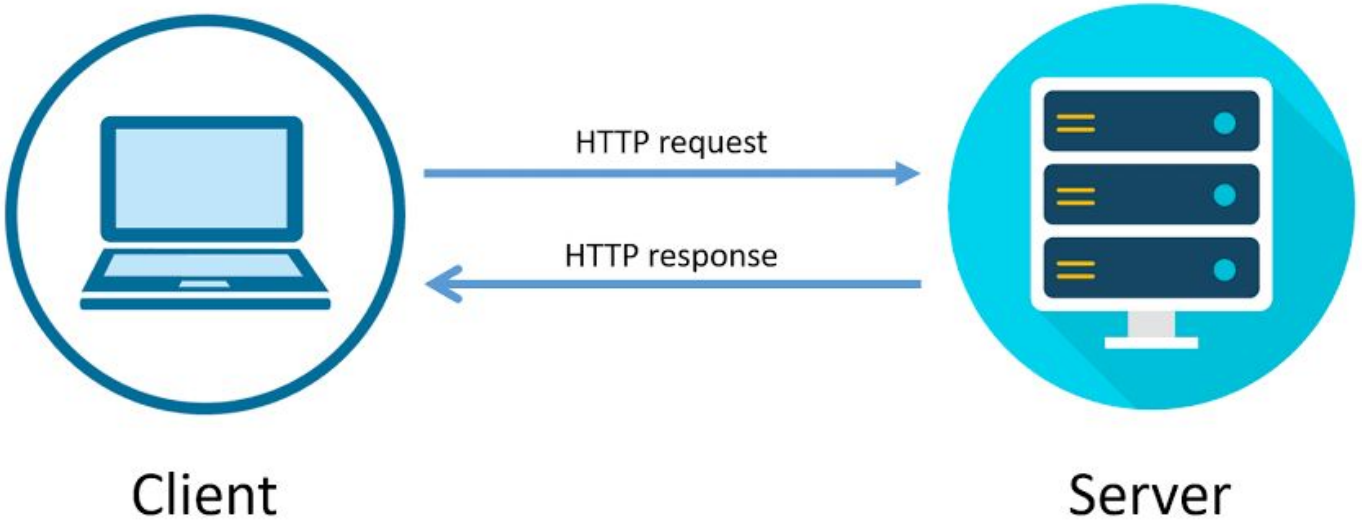
```html
<html>
    <head>
        <title>Page Title</title>
        <script src="js.js"></script>
    </head>
    <body>
        <h1>This is an HTML Page!</h1>
        <p id="demo">This is a paragraph</p>
        <button type="button" onclick="changeColor()">This is a Button</button>
        </br>
        <a href="https://www.w3schools.com/js/">
            more about js
        </a>
    </body>
</html>
```

```javascript
function changeColor(){
    document.getElementById('demo').style.color='red';
}
```

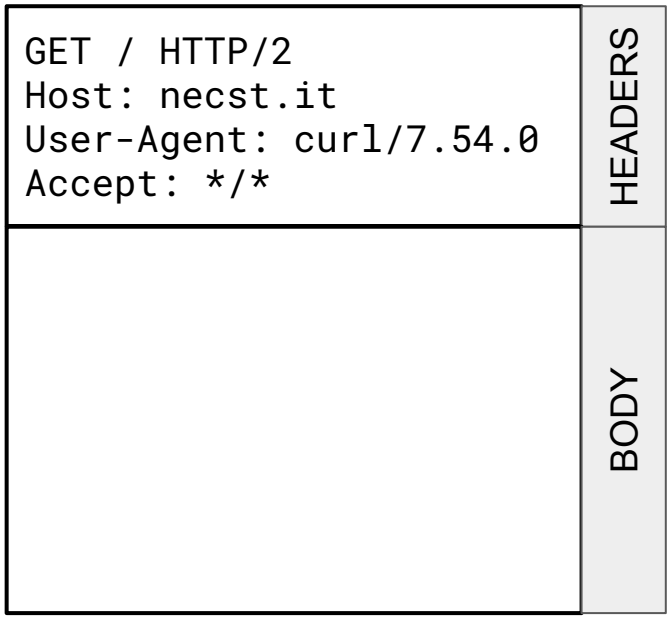
Browser Developer Tools - Source Code



How the language works? HTTP!



Getting the Page! (HTTP - Request)

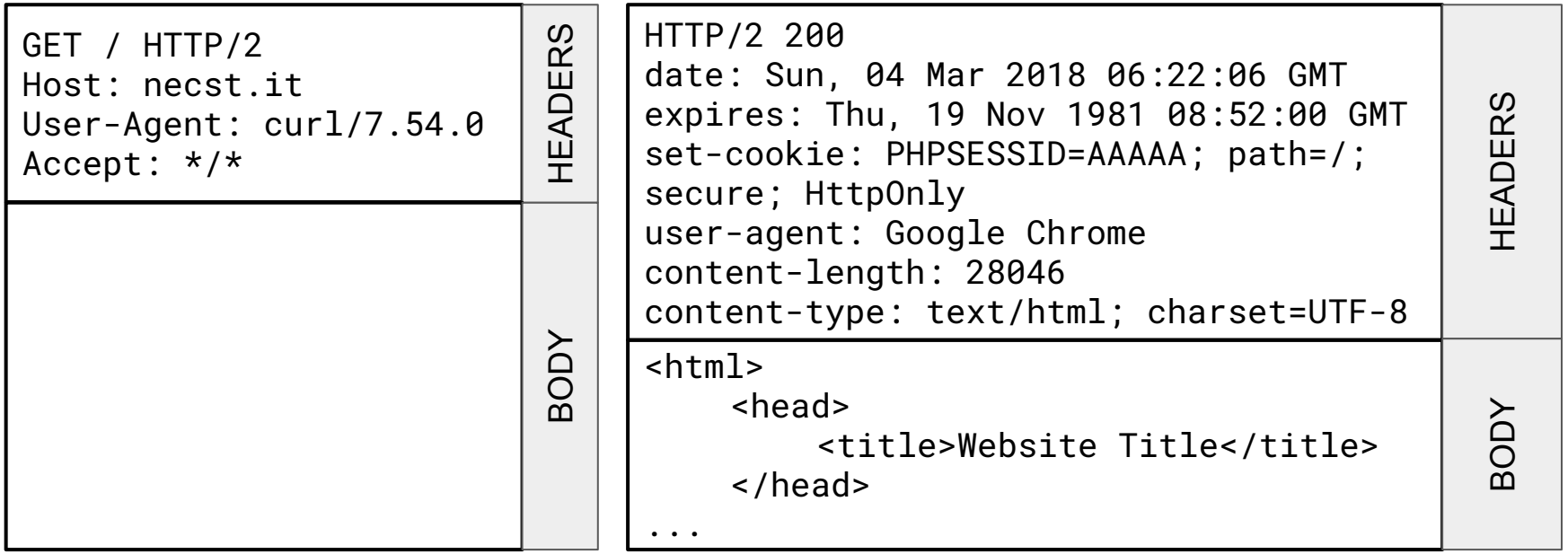


Methods
GET used for loading a page and for idempotent actions
POST used to submit forms and for state-changing actions
HEAD, PUT, DELETE, OPTIONS, ...

Request Headers
Host : The hostname. Useful for multiples website on the same IP address. (virtual Hosting)
User-Agent: The Browser type and version
Accept: what MIME type the client expect

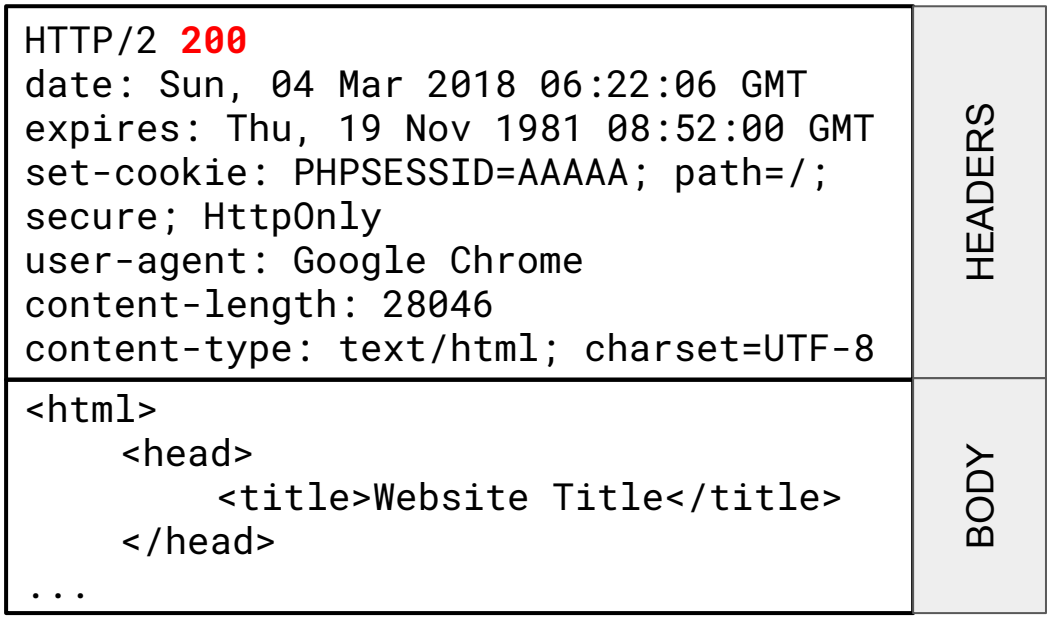
Request Body (Only for POST, PUT, and other Methods)
Contains the data (username, password, and more…)
GET Requests do not usually have a body

Getting the Page! (HTTP - Response)

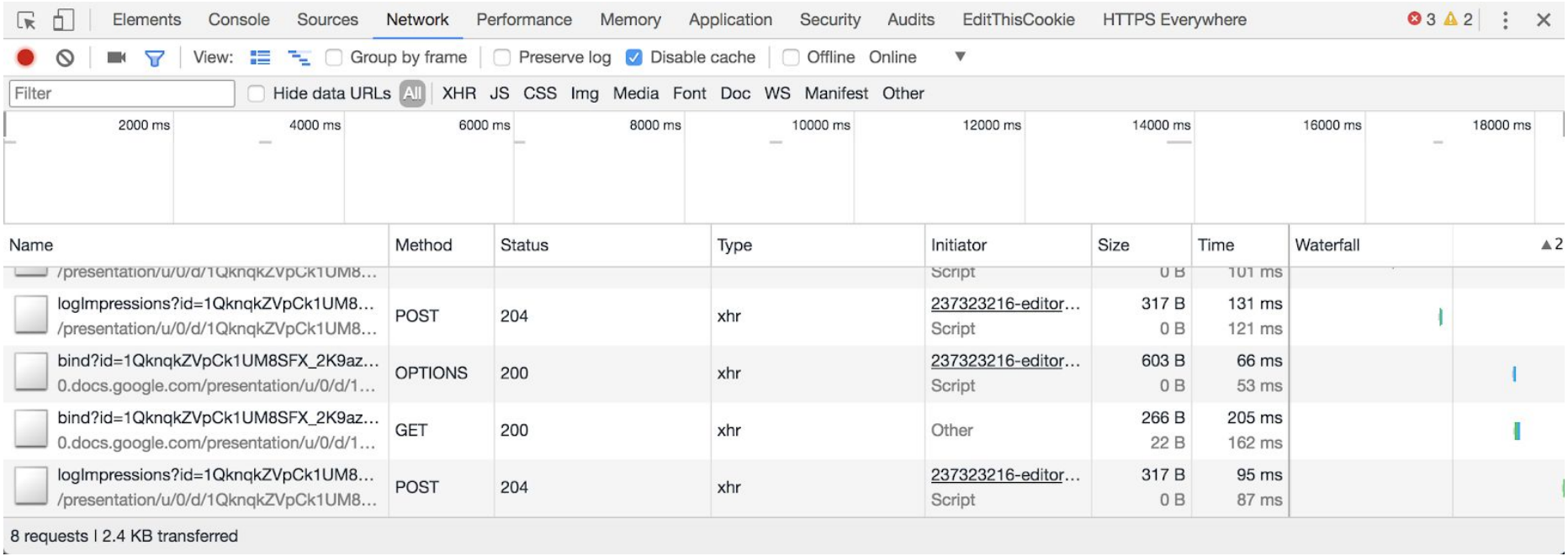


Getting the Page! (HTTP - Response)

Response Codes
1xx Information response
(e.g., 101 Switching Protocol)
2xx Successful response
(e.g., 200 OK)
3xx Redirection messages
(e.g., 302 Found)
4xx Client error responses
(e.g., 404 Not Found)
5xx Server error responses
(e.g., 500 Internal Server Error)



In Browser Developer Tools



Getting the Page! (HTTP - Request)

```
GET / HTTP/2
Host: necst.it
User-Agent: curl/7.54.0
Accept: */*
Cookie: PHPSESSID=AAAAAA
```

HEADERS

BODY

Cookies

- Browsers save cookies
- Send them on further interactions with a server in the Request Headers

Getting the Page! (HTTP - Response)

Cookies

- HTTP is a stateless protocol
- Unique strings assigned by the server
- Used to "identify" a user over multiple connections
- User profiling
- Authentication handling

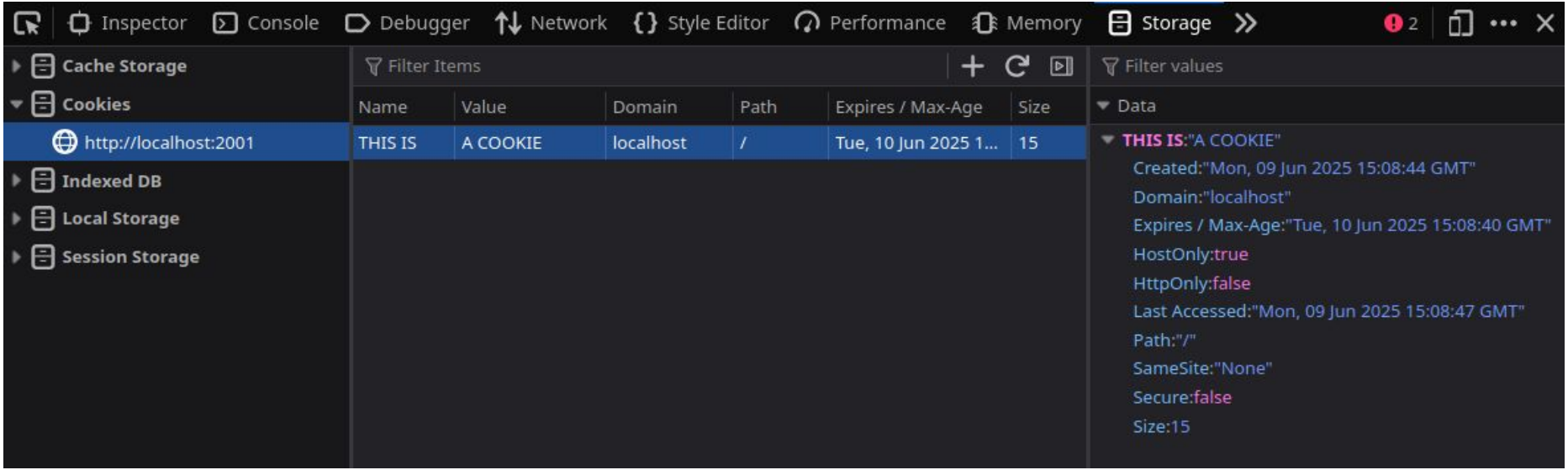
```
HTTP/2 200
date: Sun, 04 Mar 2018 06:22:06 GMT
expires: Thu, 19 Nov 1981 08:52:00 GMT
set-cookie: PHPSESSID=AAAAA; path=/;
secure; HttpOnly
x-frame-options: SAMEORIGIN
content-length: 28046
content-type: text/html; charset=UTF-8
```

HEADERS

```
<html>
    <head>
        <title>Website Title</title>
    </head>
...
```

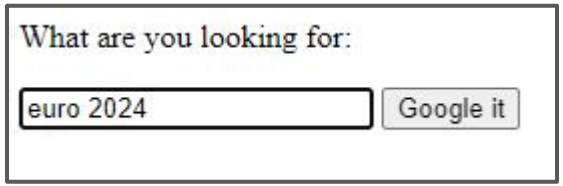
BODY

Browser Developer Tools - Console



How can we easily allow user to send data?

```
1.  <html>
2.      <head>
3.          <title>Google Page</title>
4.      </head>
5.      <body>
6.          <form method="GET" action="https://google.com/search" >
7.              <p>What are you looking for:</p>
8.              <input type="text" name="q">
9.              <input type="submit" value="Google it">
10.         </form>
11.     </body>
12. </html>
```



https://google.com/search?q=euro+2024

The browser is the tool

Three panels do everything this session needs

Inspector reads and edits the page you were served, hidden fields included

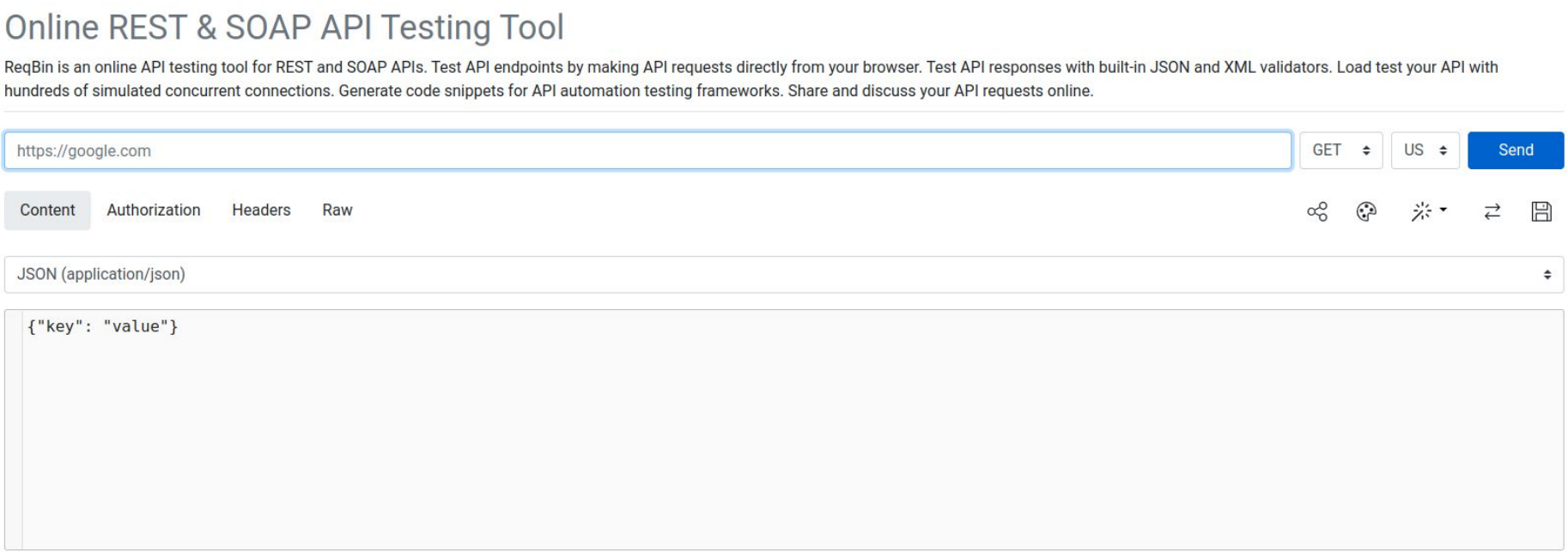
Network shows every request and response, and resends one you have edited

Console sends requests of your own, with any method you like

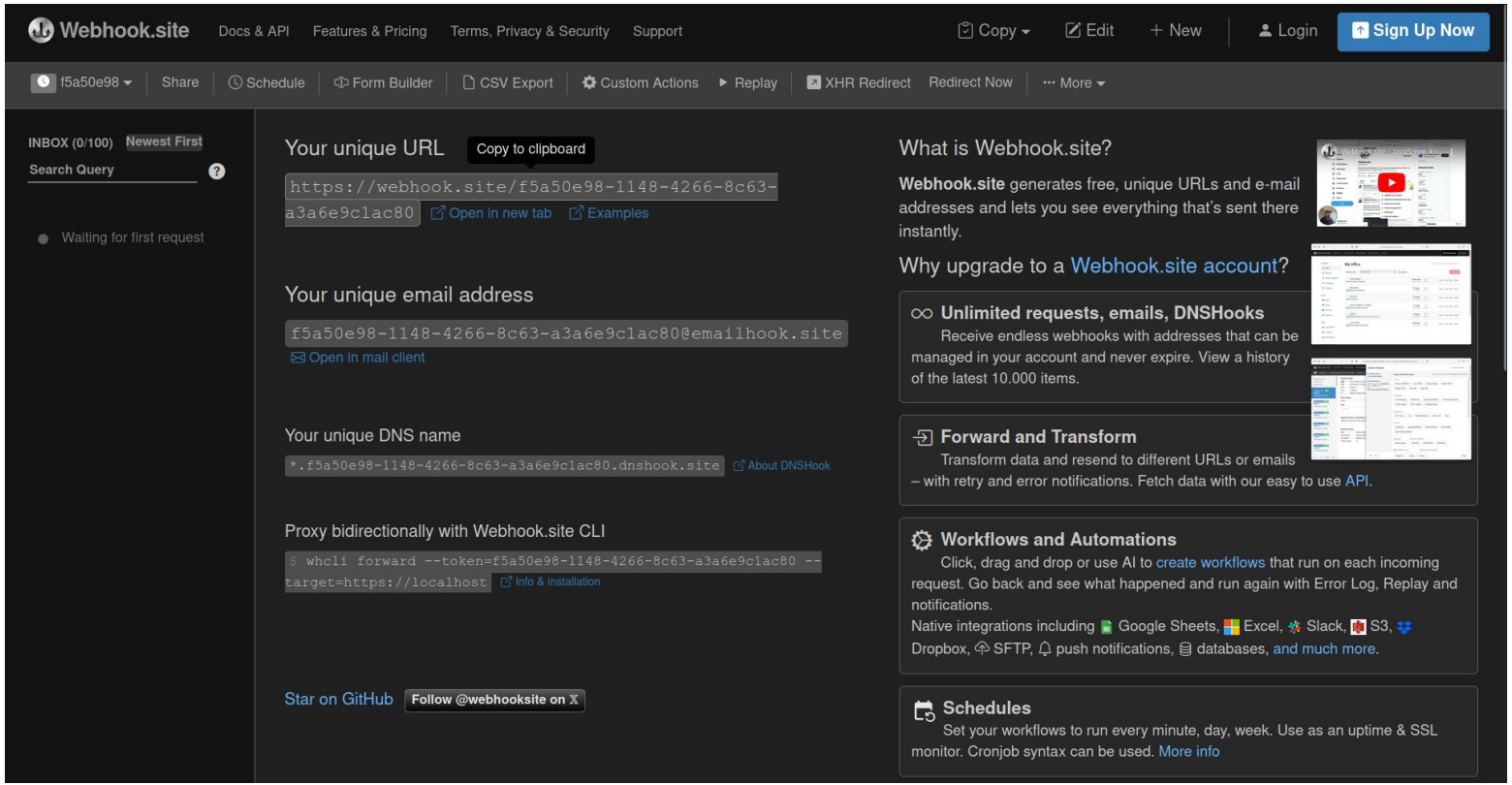
```
fetch('/', {method: 'POST'}).then(r => r.text()).then(console.log)
fetch('/', {method: 'HEAD'}).then(r => console.log(r.headers.get('Flag')))
document.cookie
```

No terminal, no proxy, nothing to install.

Reqbin.com


Online REST & SOAP API Testing Tool
ReqBin is an online API testing tool for REST and SOAP APIs. Test API endpoints by making API requests directly from your browser. Test API responses with built-in JSON and XML validators. Load test your API with hundreds of simulated concurrent connections. Generate code snippets for API automation testing frameworks. Share and discuss your API requests online.
https://google.com
GET
US
Send
Content
Authorization
Headers
Raw
JSON (application/json)
{"key": "value"}

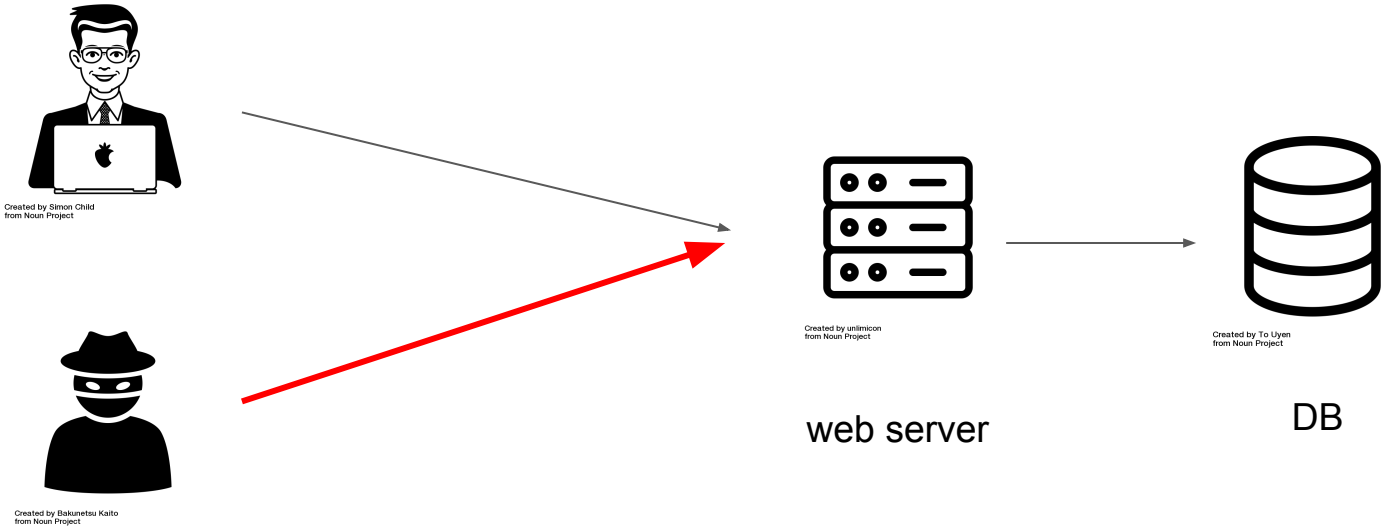
Webhook.site



Web 101: Server Attacks

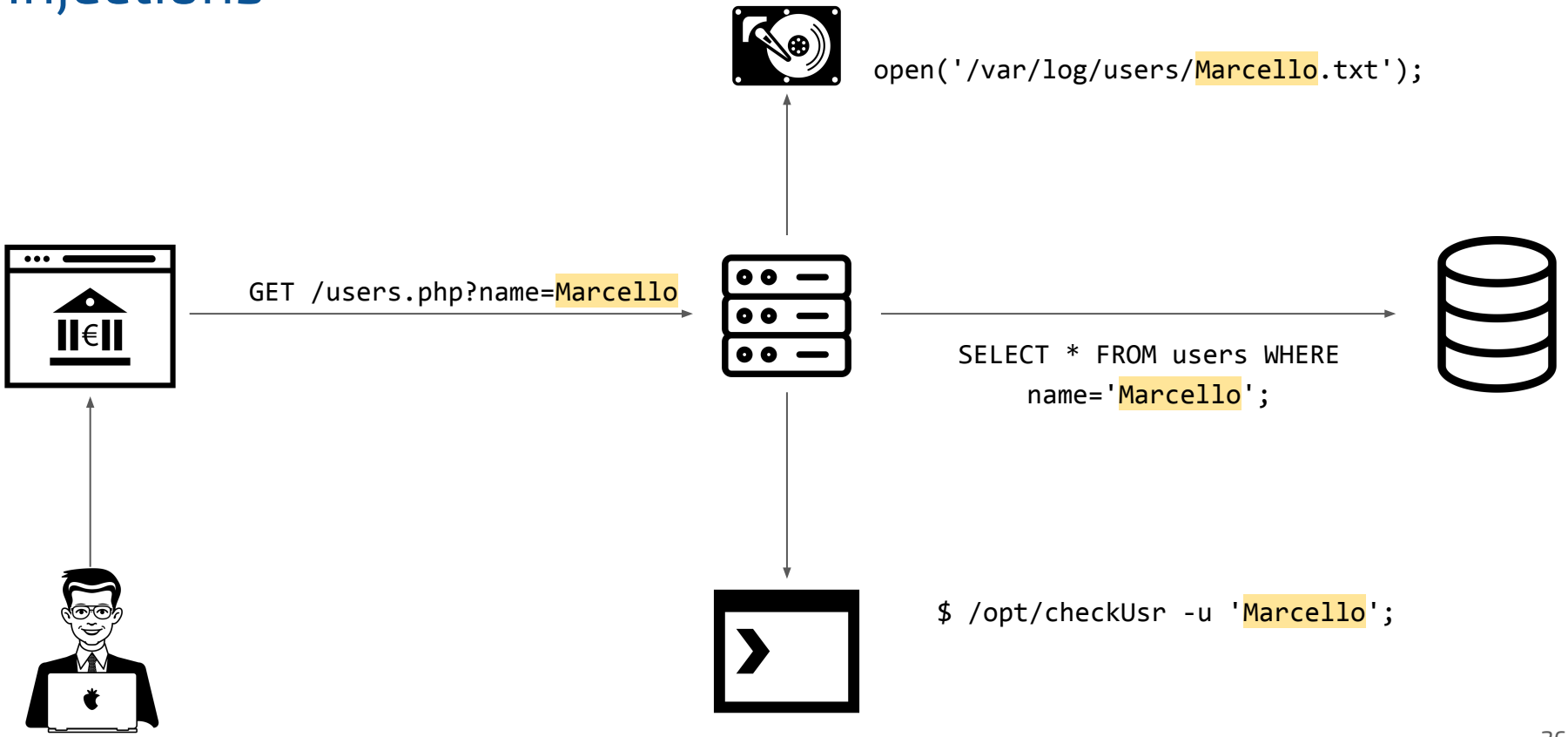
Injections

Malicious user



Goal: compromise, directly, the server(s)

Injections


open('/var/log/users/Marcello.txt');
GET /users.php?name=Marcello
SELECT * FROM users WHERE
name='Marcello';
$ /opt/checkUsr -u 'Marcello';

Never trust the user's input!

GET\POST data, headers (Referer), cookies

Data that we fetch from other servers

Data that we previously stored (where did it come?)

… needs to be checked and filtered always and properly,

otherwise bad things can (and will) happen

DEMO

Instant Transfers

Questions?