

I. Web (Server-Side Vulns)

Diego Soria y Gabriel de Miguel

Universidad
Rey Juan Carlos

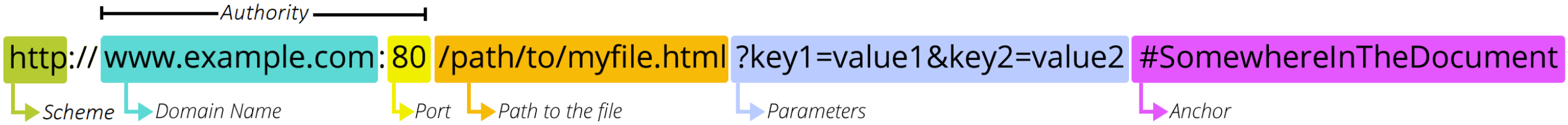
Universidad Rey Juan Carlos

ÍNDICE

1. URL
2. Funcionamiento de una WEB
3. Frameworks
4. HTTP y HTTPS
5. Fuzzing
6. Cookies

URL

URL: Uniform Resource Locator



PARTES COMUNES:

- Scheme (protocolo)
- ***Authority*** (userinfo@host:port)
- *Path*: ruta al recurso
- *Parameters*: pares clave-valorfile://localhost/etc/passwd
- Anchor: sección espefífica de la página

EJEMPLOS

- https://google.es/search?q=como+ganar+dinero
- ftp://ftp.funet.fi/pub/doc/rfc/rfc1738.txt
- mailto:raul.martin@urjc.es?subject=Que+aula+es

Funcionamiento de una página web

Universidad
Rey Juan Carlos

Archivos esenciales en una web

Los archivos esenciales son 3, un lenguaje de marcado HTML, uno de estilo CSS y otro funcional JavaScript





Universidad Rey Juan Carlos

HTML: Hyper Text Markup Lenguage

HTML es un lenguaje de marcado, que se forma por etiquetas y texto plano.

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Solo HTML</title>
</head>
<body>
    <h1>Bienvenido a mi página web</h1>
    <p>Este es un ejemplo básico de una página web utilizando solo HTML.</p>
</body>
</html>
```

Universidad Rey Juan Carlos

HTML: Hyper Text Markup Lenguage



Universidad Rey Juan Carlos

CSS: Cscading Styles Sheet

Se utiliza para dar estilo al contenido estructurado. También se puede usar con otros lenguajes como XML o SBG

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HTML con CSS</title>
    <link rel="stylesheet" href="styles.css"> <!-- Vincula el CSS -->
</head>
<body>
    <h1>Bienvenido a mi página web</h1>
    <p>Este es un ejemplo básico de una página web utilizando HTML y CSS.</p>
</body>
</html>
```

Universidad Rey Juan Carlos

CSS: Cscading Styles Sheet

Se utiliza para dar estilo al contenido estructurado. También se puede usar con otros lenguajes como XML o SBG

```css
/* Estilos generales */
body {
    font-family: 'Helvetica Neue', Arial, sans-serif;
    background-color: #282c34;
    color: #fff;
    margin: 0;
    padding: 0;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
```

```css
/* Estilo del título */
h1 {
    color: #61dafb;
    font-size: 3em;
    text-transform: uppercase;
    letter-spacing: 5px;
    border-bottom: 2px solid #61dafb;
    padding-bottom: 10px;
    margin-bottom: 20px;
}

/* Estilo del párrafo */
p {
    color: #b0bec5;
    font-size: 1.5em;
    max-width: 600px;
    text-align: center;
    line-height: 1.6;
    margin: 20px;
    border-left: 4px solid #61dafb;
    padding-left: 15px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}
```

Universidad Rey Juan Carlos

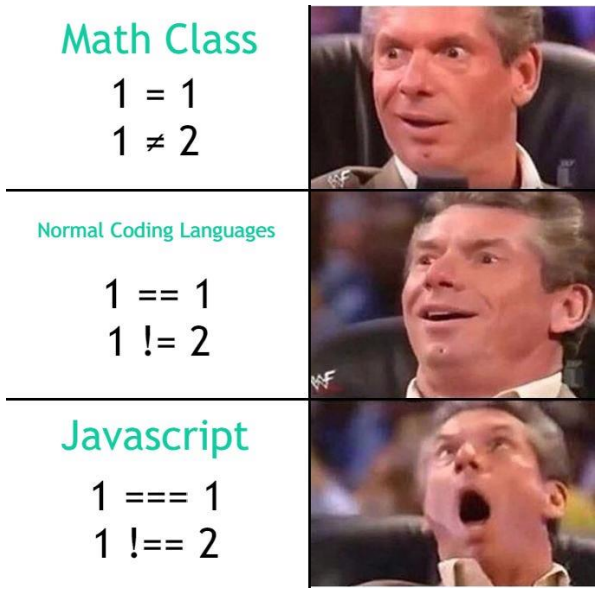
CSS: Cscading Styles Sheet



Universidad Rey Juan Carlos

JavaScript

JavaScript es un lenguaje de programación dinámico que permite agregar interactividad y funcionalidades complejas a las páginas web




```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HTML, CSS y JavaScript</title>
    <link rel="stylesheet" href="styles.css"> <!-- Vincula el CSS -->
</head>
<body>
    <h1>Haz clic en este título</h1>
    <p>Este es un ejemplo básico de una página web utilizando HTML, CSS y JavaScript.</p>

    <script src="script.js"></script> <!-- Vincula el JavaScript -->
</body>
</html
```

Universidad Rey Juan Carlos

JavaScript

JavaScript es un lenguaje de programación dinámico que permite agregar interactividad y funcionalidades complejas a las páginas web

```
function cambiarColor() {
    const titulo = document.querySelector('h1');
    titulo.style.color = titulo.style.color === 'blue' ? '#61dafb' : 'blue';
}

document.querySelector('h1').addEventListener('click', cambiarColor);
```

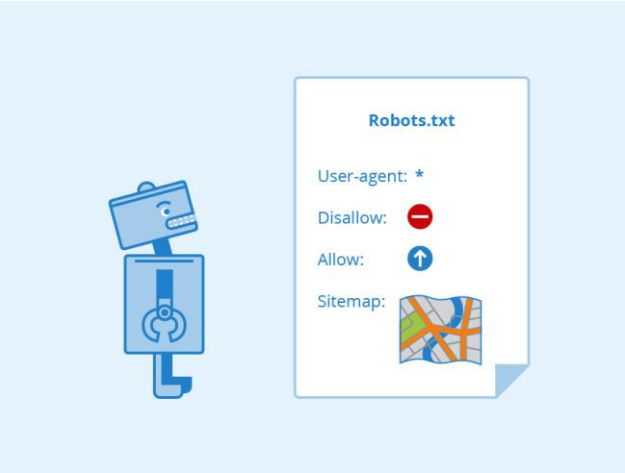
JavaScript


index.html
Open with Arc
HAZ CLIC EN ESTE TÍTULO
Este es un ejemplo básico de una página web utilizando HTML, CSS y JavaScript.

Universidad Rey Juan Carlos

ROBOTS.TXT

- Es un archivo de texto sencillo cuya única función es la de indicar a los rastreadores, bots e inteligencias artificiales los directorios y rutas a los que no pueden acceder.
- En CTFs permite conocer rutas ocultas o expuestas de manera involuntaria



urjc.es/robots.txt

```
User-agent: *
Crawl-delay: 10
Disallow: /administrator/
Disallow: /bin/
Disallow: /cache/
Disallow: /cli/
Disallow: /components/
Disallow: /includes/
Disallow: /installation/
Disallow: /language/
Disallow: /layouts/
Disallow: /libraries/
Disallow: /logs/
Disallow: /media/
Disallow: /modules/
Disallow: /plugins/
Disallow: /templates/
Disallow: /tmp/
Disallow: /images/Intranet/
```

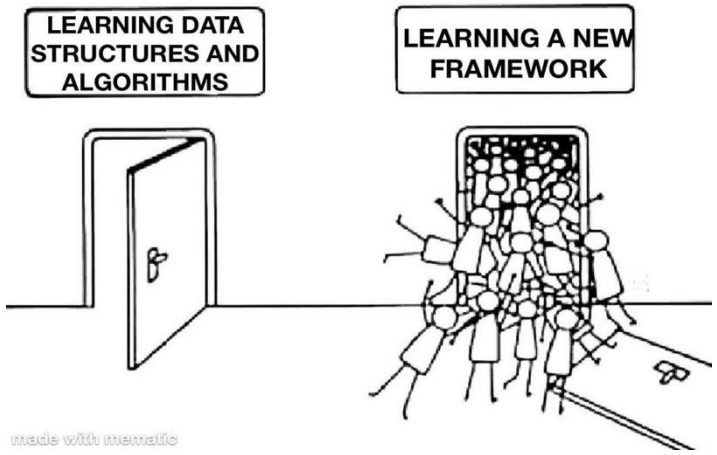
Frameworks

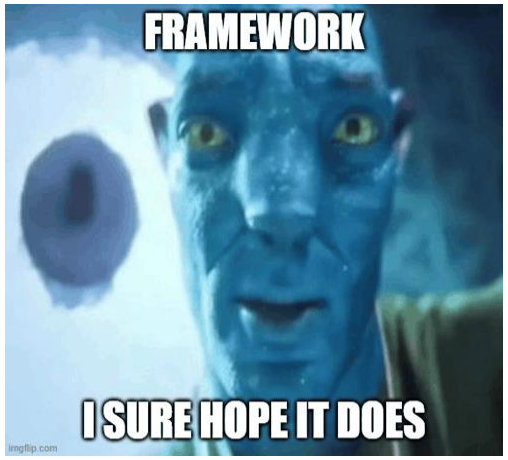


Universidad
Rey Juan Carlos

¿Para qué sirven?

- Un **framework** es una estructura o plantilla conceptual y tecnológica que ofrece una base predefinida de herramientas, bibliotecas y reglas para desarrollar software.
- Permiten aligerar y facilitar mucho la creación de webs.
- De cara a ciberseguridad. Saber qué frameworks o elementos extra utiliza una página nos da una gran ventaja sobre qué comportamiento o potenciales vulnerabilidades va a tener.

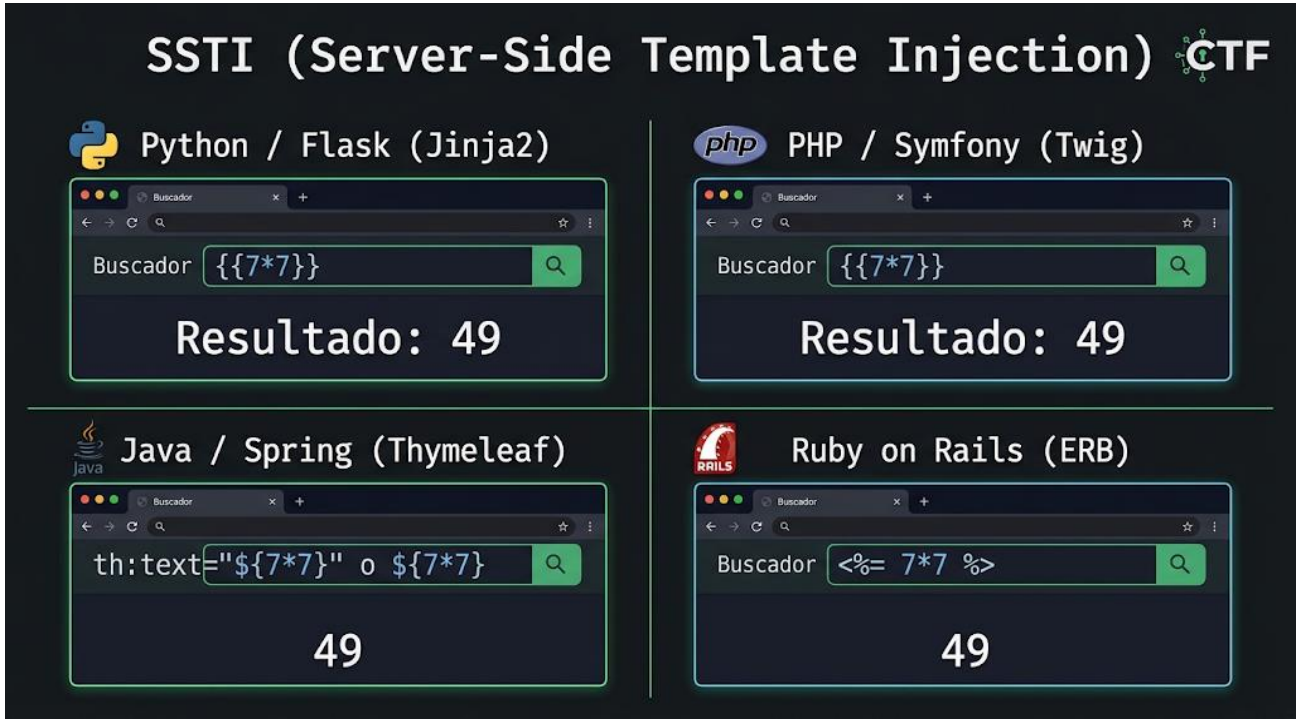




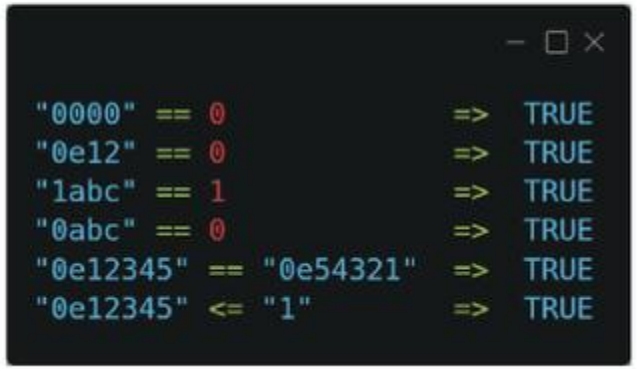
Universidad Rey Juan Carlos

SSTI: Server-side Template Injection

Es una vulnerabilidad que ocurre cuando el servidor confunde el input del usuario con código legítimo del motor de plantillas (Jinja, Twig, Spring, etc.). Y, por tanto, lo ejecuta.







Formas de analizar el framework: wappalyzer

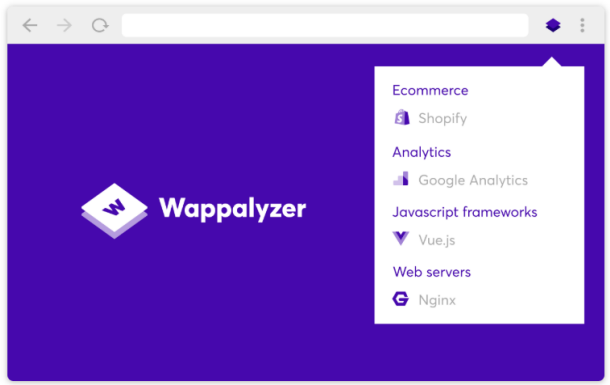
Wappalyzer

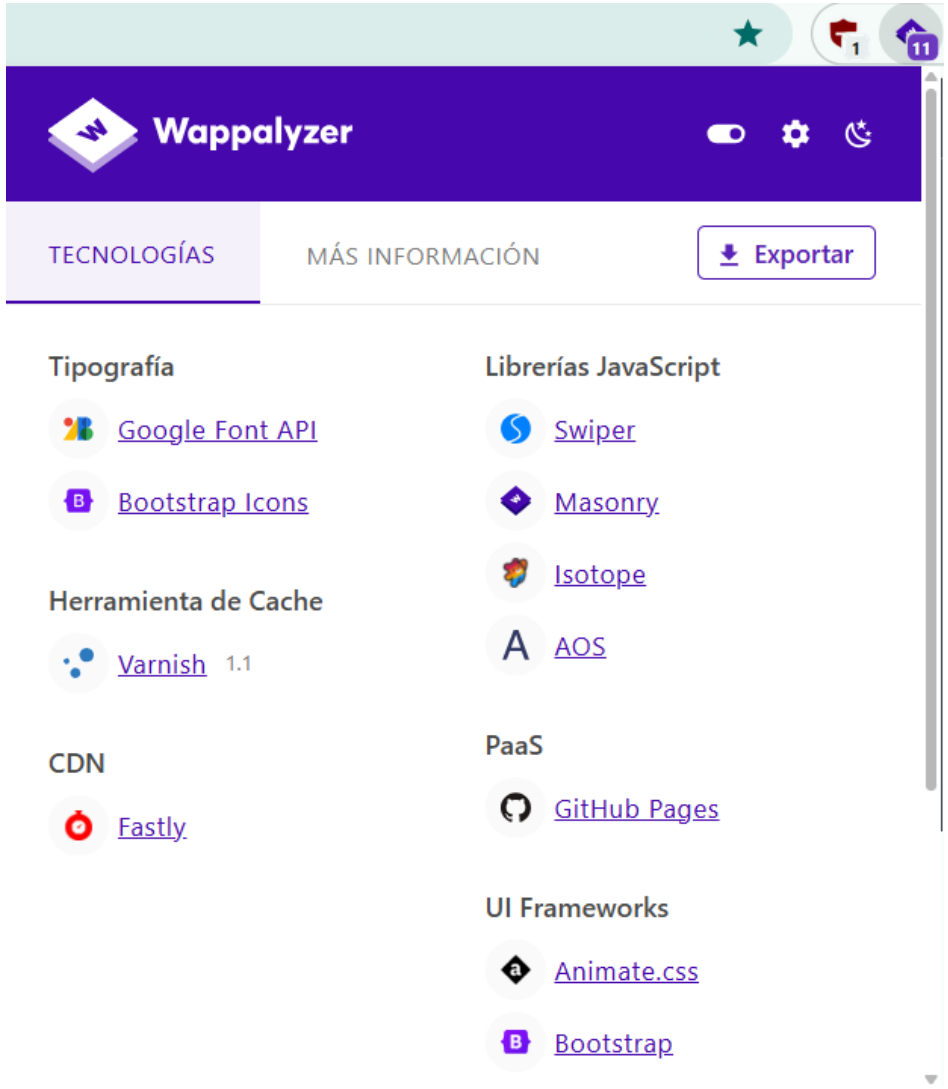
Wappalyzer - Technology profiler

Desinstalar

wappalyzer.com Destacado 4,0 ★ (2 mil valoraciones) Compartir

Extensión DevTools 3.000.000 usuarios

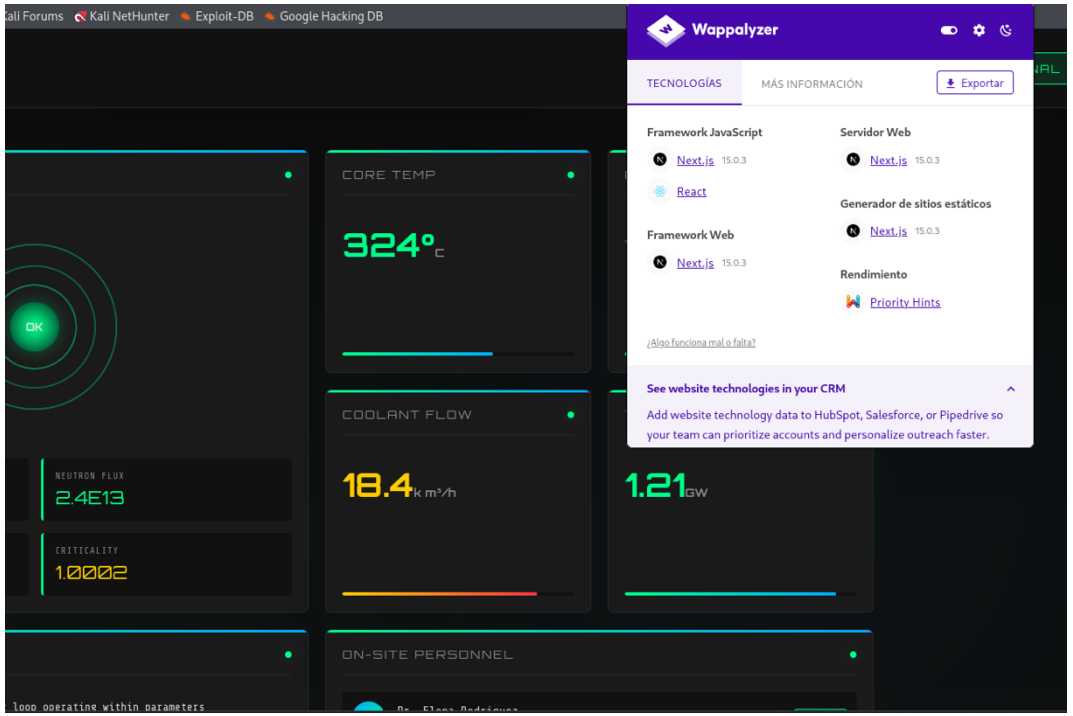


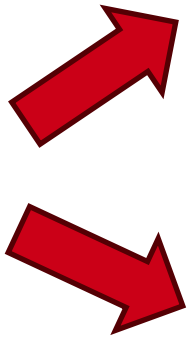


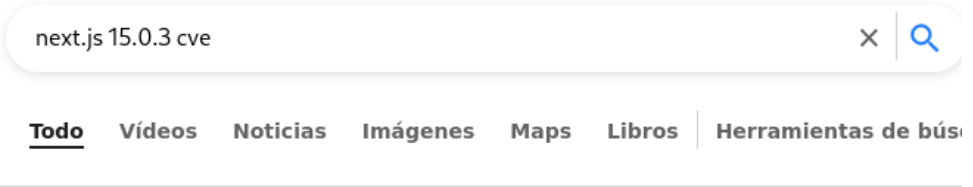
Publicidad no remunerada :(

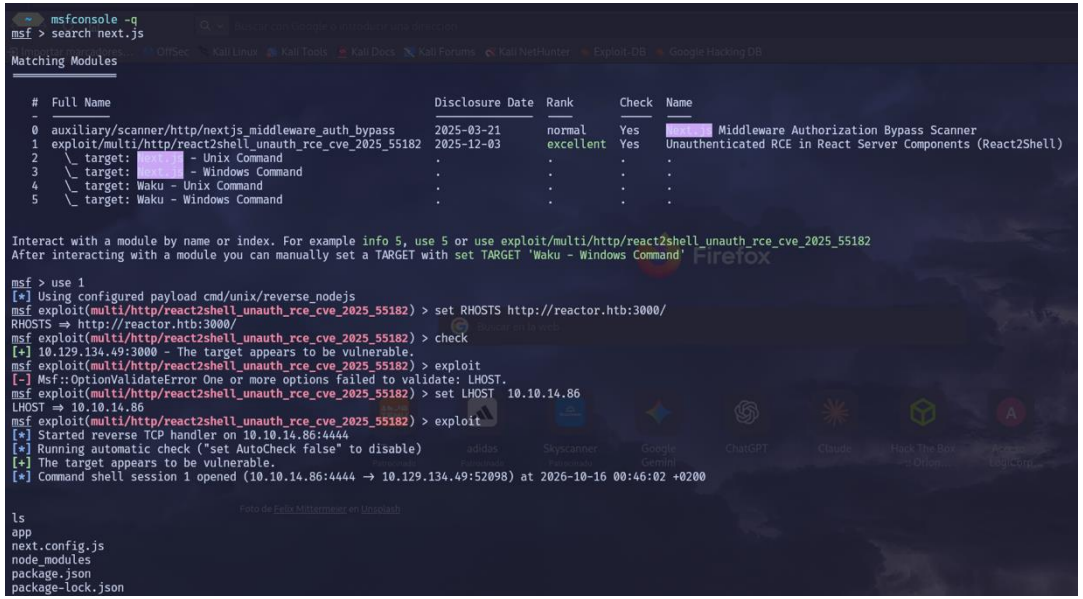
Universidad Rey Juan Carlos

Formas de analizar el framework: wappalyzer









Universidad Rey Juan Carlos

Formas de analizar el framework: Whatweb

```
~ tldr whatweb                                                    ✔ 12:56:47 a. m.
Escáner web de nueva generación.
Más información: <https://github.com/urbanadventurer/WhatWeb#usage>.

Escanea sitios web/objetivos en busca de tecnologías web:

    whatweb website1 website2 ...

Lee objetivos/sitios web desde un archivo:

    whatweb --input-file archivo_objetivos

Analiza un sitio web/objetivo en modo detallado:

    whatweb --verbose example.com

Ejecuta un escaneo agresivo en un sitio web:

    whatweb --aggression 3 example.com

Escanea una red y suprime los errores:

    whatweb --no-errors 192.168.0.0/24

Lista de complementos:

    whatweb --list-plugins

Lista de complementos:

    whatweb --info-plugins nombre_del_complemento

~ whatweb google.com                                              ✔ 12:56:50 a. m.
http://google.com [301 Moved Permanently] Country[UNITED STATES][US], HTTPServer[gws], IP[216.58.204.174], RedirectLocation[http://www.google.com/], Title[301 Moved], UncommonHeaders[content-security-policy-report-only], X-Frame-Options[SAMEORIGIN], X-XSS-Protection[0]
https://google.com [301 Moved Permanently] Country[UNITED STATES][US], HTTPServer[gws], IP[216.58.204.174], RedirectLocation[https://www.google.com/], Title[301 Moved], UncommonHeaders[content-security-policy-report-only,alt-svc], X-Frame-Options[SAMEORIGIN], X-XSS-Protection[0]
http://www.google.com/ [200 OK] Cookies[AEC], Country[UNITED STATES][US], HTML5, HTTPServer[gws], HttpOnly[AEC], IP[142.251.152.119], Script, Title[Google], UncommonHeaders[content-security-policy-report-only], X-Frame-Options[SAMEORIGIN], X-XSS-Protection[0]
https://www.google.com/ [200 OK] Cookies[AEC,SOCS,__Secure-ENID], Country[UNITED STATES][US], HTML5, HTTPServer[gws], HttpOnly[AEC,__Secure-ENID], IP[142.251.154.119], Script, Title[Google], UncommonHeaders[content-security-policy-report-only,accept-ch,alt-svc], X-Frame-Options[SAMEORIGIN], X-XSS-Protection[0]
```

HTTP y HTTPS

HTTPS

Estos son los protocolos que hacen que la web funcione, la diferencia entre ellos es que HTTPS es HTTP con TLS, es decir cifrado.

HTTPS va a añadir los siguientes pasos al HTTP

- Cifrar la petición con una clave simétrica
- Enviar el mensaje
- El que reciba la petición lo descifra con la misma clave simétrica



Universidad Rey Juan Carlos

MÉTODOS HTTPS

MÉTODO	PROPÓSITO	¿ES SEGURO?	¿ES IDEMPOTENTE?	TIPO
GET	Obtener/leer un recurso	SI	SI	Lectura
POST	Crear un nuevo recurso	NO	NO	Escritura
PUT	Reemplazar un recurso	NO	SI	Escritura
PATCH	Modificar un recurso	NO	NO	Escritura
DELETE	Eliminar un recurso	NO	SI	Escritura
HEAD	GET pero solo cabeceras	SI	SI	Lectura
OPTIONS	Consultar opciones del servidor	SI	SI	Lectura

Universidad Rey Juan Carlos

ENVIAR PARÁMETROS

POR URL

https:/página.com?campo=valor

Método GET

EN EL CUERPO

{ "usuario": "user", "email":"user@example.com", "role":user...}

nombre=user&email=user%40example.com&role=user

Métodos de Escritura

EN LAS CABECERAS

Referer: google.com

Authorization: Bearer <token_jwt>

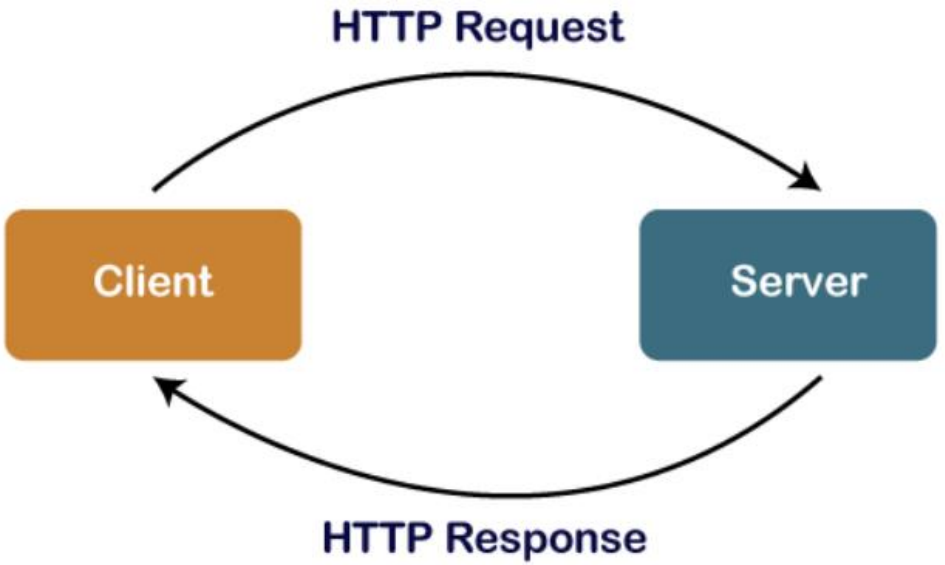
Content-Type: application/json

Cualquier Método

Códigos de estado

¿ERROR 404? Ya iba tocando saber que significa



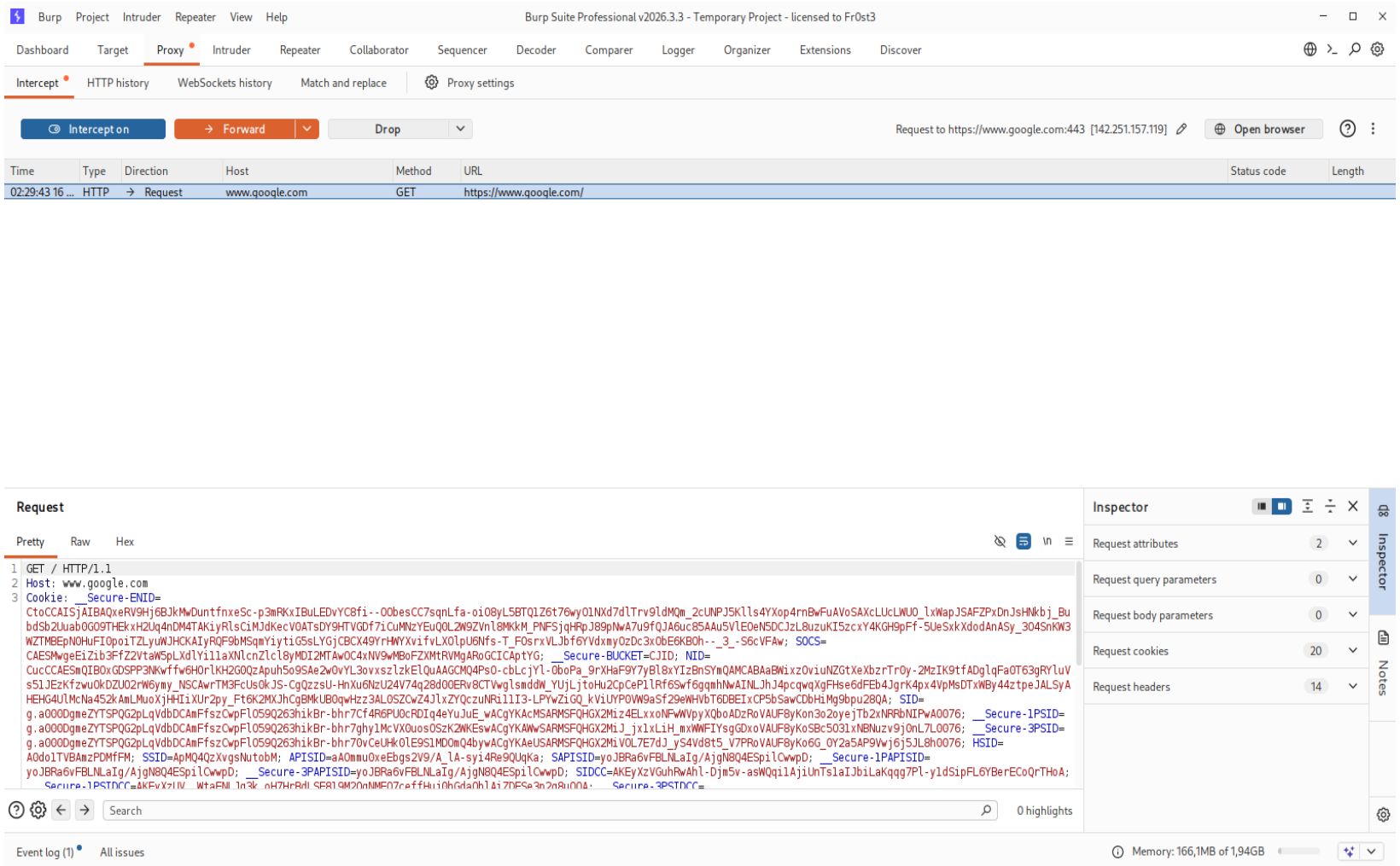


Más información sobre los códigos de estado: https://developer.mozilla.org/es/docs/Web/HTTP/Status

Universidad Rey Juan Carlos

Herramientas

BURPSUITE



FUZZING

Universidad Rey Juan Carlos

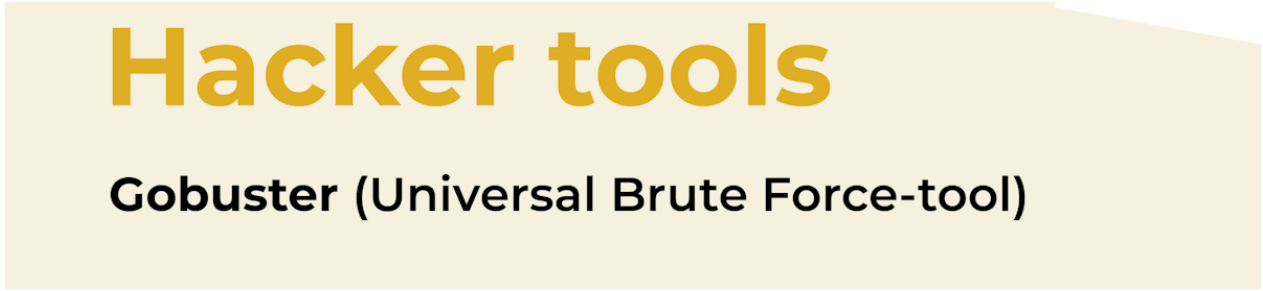
FUZZING

Fuzzing nos sirve para descubir directorios, parámetros o demás campos de una página web, por fuerza bruta

Si recordamos lo que es el código de estado, realizando peticiones a una página web, podemos comprobar si un recurso existe o no.

Para ello existen diferentes herramientas: wfuzz, ffuz, dirb, gobuster.
Hoy os voy a hablar de la más potente de ellas.

Wordlist recomendada: (…)/SecLists/Discovery/Web-Content/common.txt



Universidad Rey Juan Carlos

FUZZING: COMANDOS

Escaneo de directorios

```
ffuf -u https://www.example.com/FUZZ -w ~/ctf/wordlists/SecLists/Discovery/Web-Content/common.txt

gobuster dir -u https://www.example.com/ -w ~/ctf/wordlists/SecLists/Discovery/Web-Content/common.txt


ffuf -u https://example.com -H "Host: FUZZ.example.com" -w ~/ctf/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt

gobuster vhost -u https://example.com/ -w ~/ctf/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt --append-domain
```

Escaneo de Hosts

Cookies



Universidad
Rey Juan Carlos

Cookies

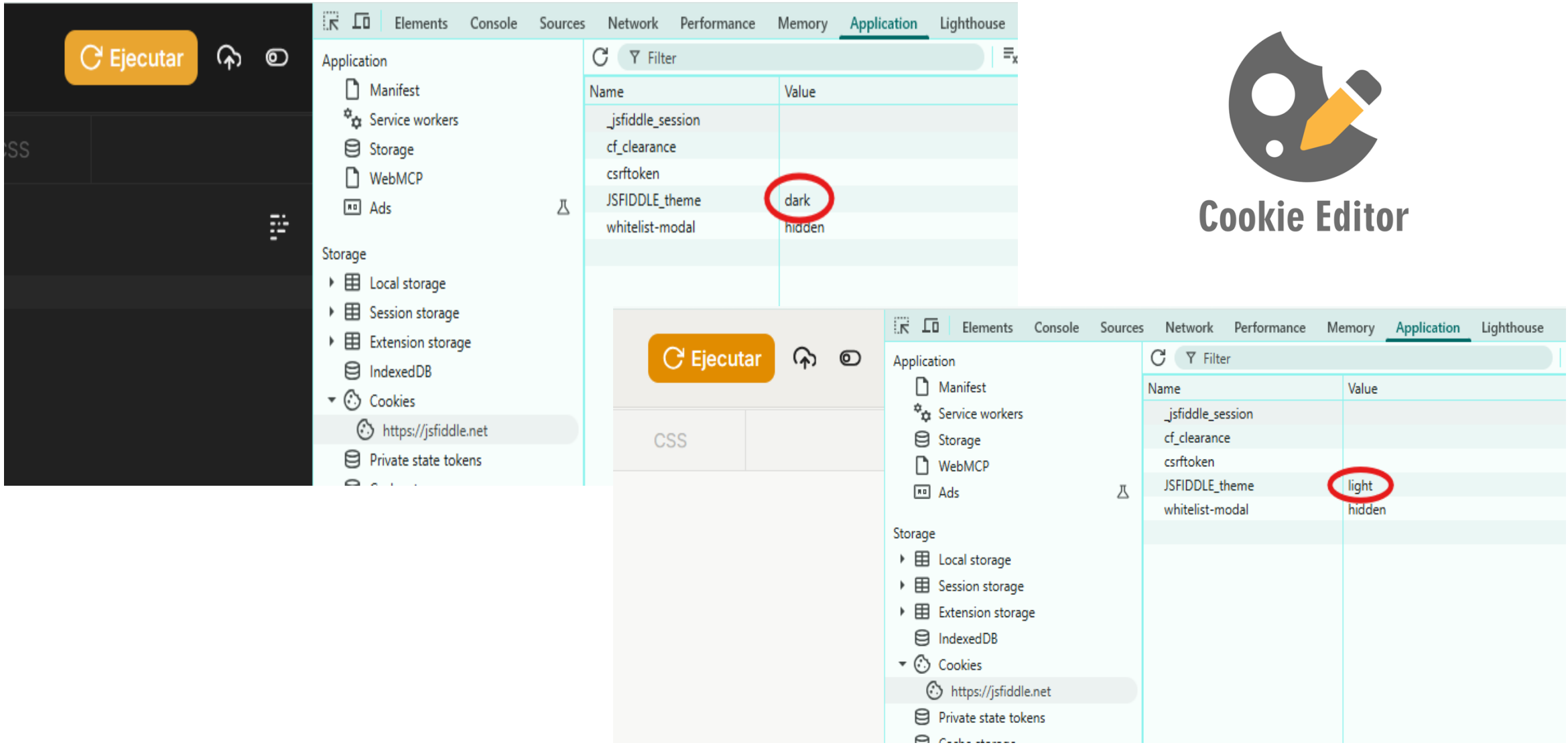


¿Qué son las Cookies?

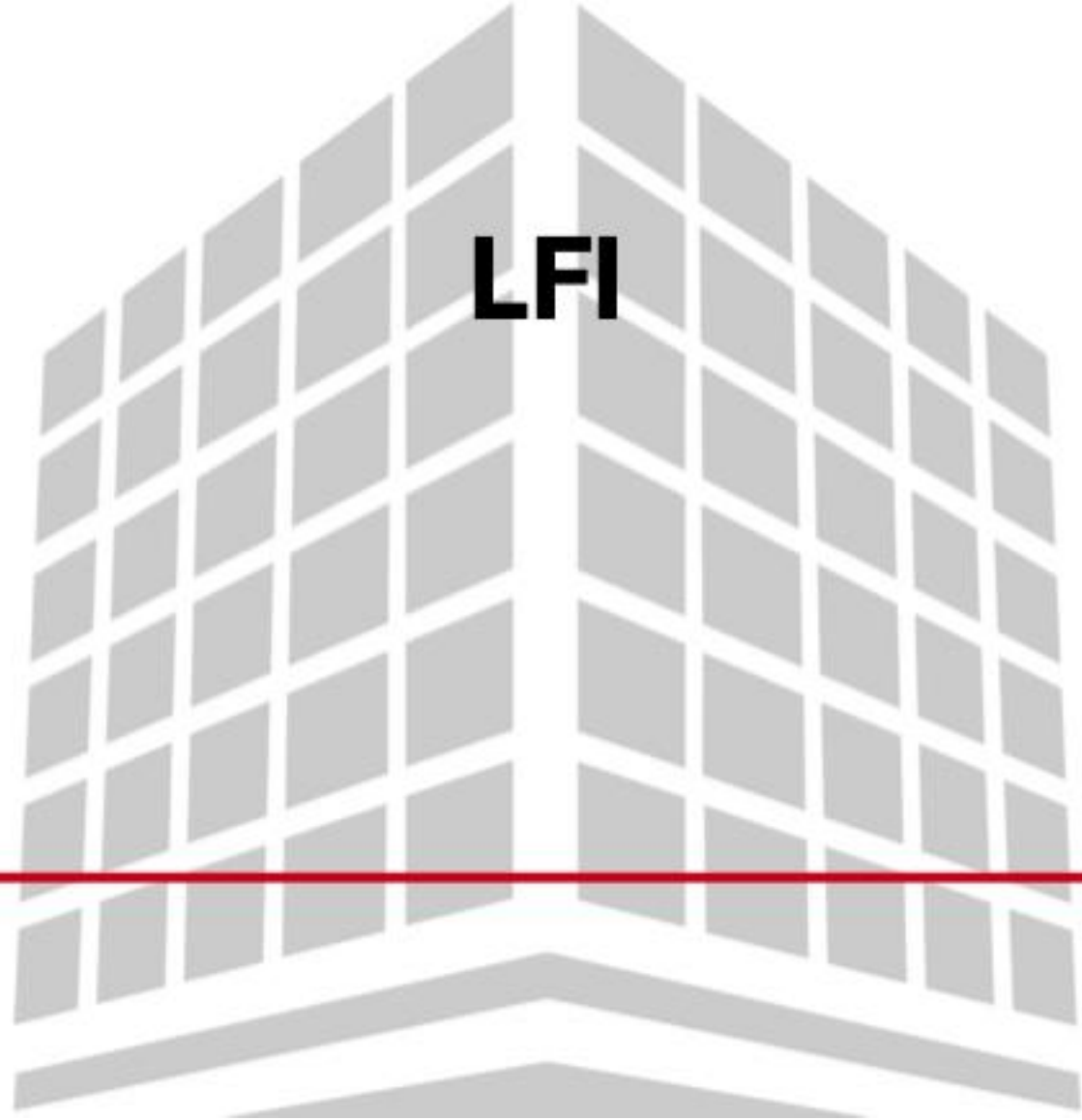
- Pequeños archivos de texto que se almacenan en el navegador del usuario.
- Crean un "registro" que ayuda a personalizar la experiencia del usuario.
- Permiten otorgar permanencia en la web a las preferencias del usuario.
- Información de inicio de sesión, estilo visual...
- **Si sabes cómo se crearon y cómo las interpreta la página: pueden manipularse, editarse o incluso crear una nueva cookie**

Universidad Rey Juan Carlos

Herramientas



LFI

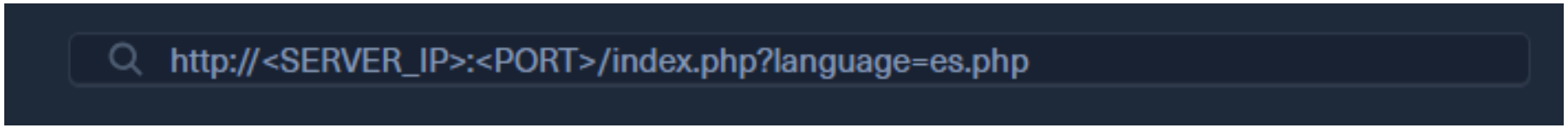


Universidad
Rey Juan Carlos

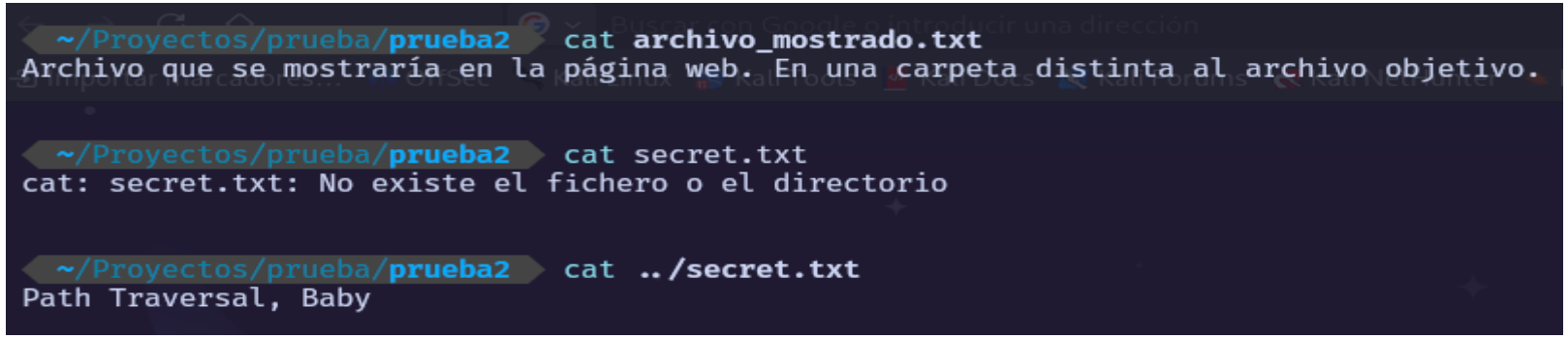
Universidad Rey Juan Carlos

LFI: Local File Inclusion

Es una vulnerabilidad web causada por una función mal implementada o sin sanitizar que permite leer archivos locales en el servidor.



PATH TRAVERSAL: Si el archivo que queremos no está en la misma carpeta que el archivo mostrado por la página, deberemos movernos entre carpetas como si estuviéramos en una terminal. Es decir, ".." representa el directorio/carpeta anterior.



Universidad Rey Juan Carlos

LFI: Identificación

```javascript
if(req.query.language) {
    fs.readFile(path.join(__dirname, req.query.language), function (err, data) {
        res.write(data);
    });
}
```

```jsp
<c:if test="${not empty param.language}">
    <jsp:include file="<%= request.getParameter('language') %>" />
</c:if>
```

```php
if (isset($_GET['language'])) {
    include($_GET['language']);
}
```

Universidad Rey Juan Carlos

LFI: BYPASS

Es común que los programadores traten de sanitizar posibles LFI mediante el uso de distintas técnicas. Según su naturaleza, podemos plantear distintas vías alternativas para conseguir enviar el mismo "payload".

Sanitización

str_replace('../', '', $page)

Bypass posible

?page=..../////..../////etc/passwd

O

?page=..././/..././/..././etc/passwd

Sanitización

Bloqueo de caracteres especiales

Bypass posible

URL Encoding doble: Codificar los caracteres % dos veces.

%2e%2e%2f (simple) ➔ %252e%252e%252f (doble)

Sanitización

Extensión de archivo forzada

include($_GET['page'] . '.php');

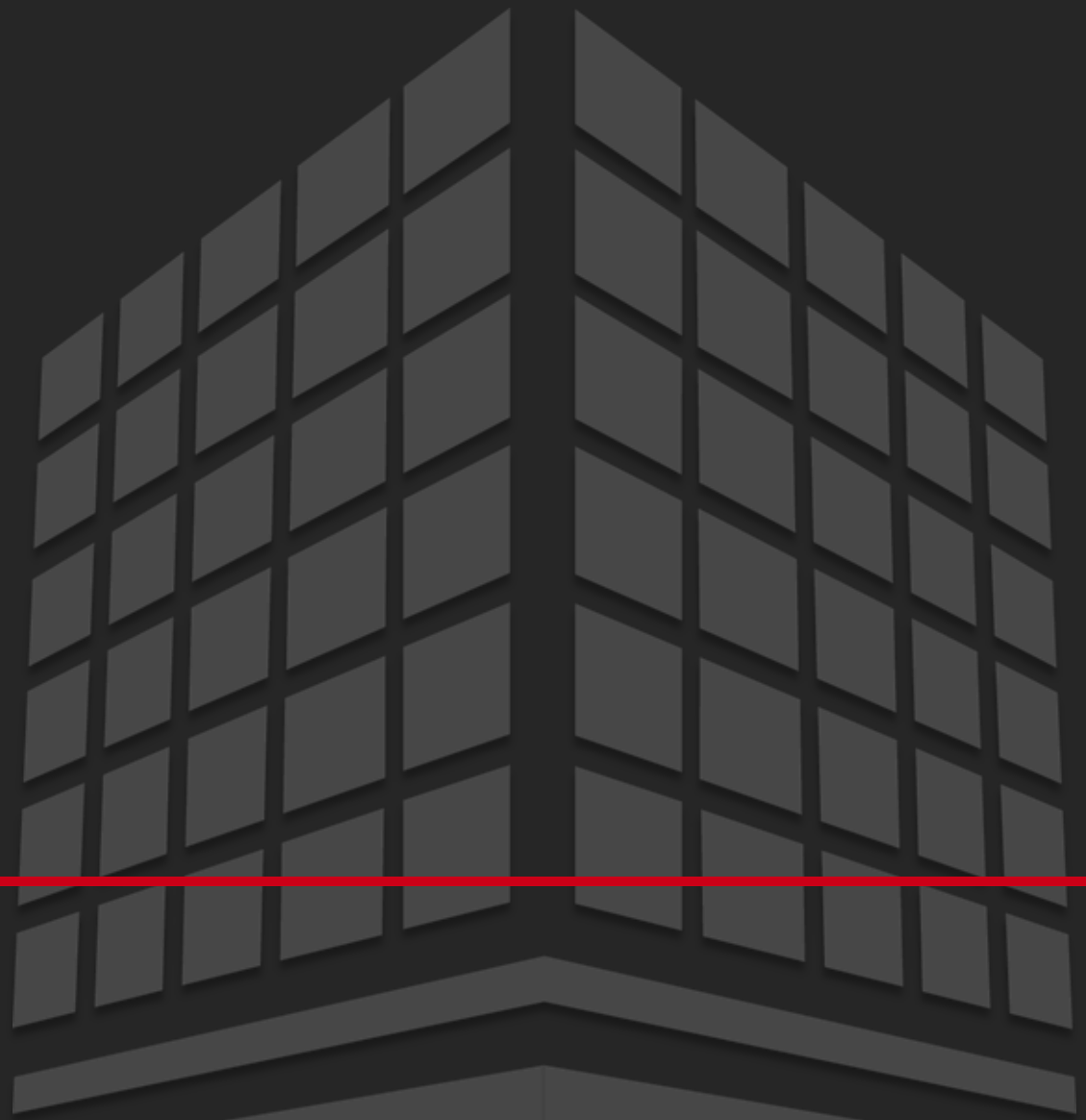
Bypass posible

(null byte injection)

?page=../../../../etc/passwd%00

RETOS FINAL





Universidad
Rey Juan Carlos