

## DevSecOps & Cloud Security - Examen práctico - Marilyn Shirley Olivera

### Índice:

➤ Configuración de Host en GCP .....	1 – 4
○ Instalación de Coolify instancia de servidor de GCP	
○ Configuración de cuenta de Coolify	
○ Asignación de dominio a Coolify	
➤ Integro Coolify a GitHub Actions .....	4 – 8
○ Configuración en Coolify	
○ Reconfigurando pipeline	
○ Ejecución del pipeline	
○ Configuración de dominio	
➤ A experimentar con las vulnerabilidades: “¡Finalmente hora de jugar!”.....	8 – 24
○ Reflected XSS Example 1 .....	8
○ Reflected XSS Example 2 .....	9
○ Reflected XSS Example 3 .....	9
○ Stored XSS Example .....	12
○ CSRF Example .....	13
○ Fuzzing .....	15
○ Authentication Bypass .....	18
○ Directory Traversal .....	19
○ Insecure Direct Object Reference (IDOR) .....	20
○ Injections and remote code execution .....	21
○ Mixed topics .....	22
➤ Intento 1 de resolución con Oracle y Coolify .....	25 – 29
➤ Implementación de workflow ZAP proxy .....	30 – 31

## DevSecOps & Cloud Security

Nombre y Apellido: Marilyn Shirley Olivera Cam8 Host en la Nube: GCP

Cam8

## Host en la Nube: GCP

### **Links importantes:**

- URL de la aplicación: <https://marilynolivera.alexander.net.ar/>
  - Consigna: [https://drive.google.com/file/d/168edR\\_0Wbj\\_0800LmcOU\\_xxxxGXDkjYj/view](https://drive.google.com/file/d/168edR_0Wbj_0800LmcOU_xxxxGXDkjYj/view)
  - Repositorio de aplicación provisto: <http://github.com/GinoDevOps/vulnerable-web-app>
  - Repositorio mio con fork de aplicación: <https://github.com/marilynolivera/vulnerable-web-app>

Inicialmente opté por usar un Servidor de Oracle, pero luego de tener trabas y no saber como solucionarlas me decidí a usar GCP Cloud en la que aún cuento con la prueba gratuita.

Para desarrollar el pipeline voy a usar GitHub Actions, voy a subir la imagen a GitHub Container Registry (GCR) y voy a usar Coolify para hacer el despliegue dentro del VPC de GCP (Instancia de Máquina Virtual (VM) de Compute Engine).

**- Configuración de Host en GCP:** Crear una instancia de Compute Engine en GCP, asegurandome de seleccionar una configuración que use la arquitectura x86\_64 (AMD64), siguiendo las recomendaciones para el host que provee Coolify <https://coolify.io/docs/get-started/installation>

Estimación mensual	
<b>USD34.46</b>	
Equivale a alrededor de USD0.05 por hora	
Paga por lo que usas, con facturación por segundo y sin pagos por adelantado	
Elemento	Estimación mensual
2 CPU virtuales + 4 GB de memoria	USD24.46
Disco persistente balanceado de 100 GB	USD10.00
<a href="#">Logging</a>	<a href="#">El costo varía ↗</a>
<a href="#">Supervisión</a>	<a href="#">El costo varía ↗</a>
Programación de instantáneas	<a href="#">El costo varía ↗</a>
<b>Total</b>	<b>USD34.46</b>
<a href="#">Precios de Compute Engine ↗</a>	
<a href="#">Precios de Cloud Operations ↗</a>	
<a href="#">Menos</a>	

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1020-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

5 updates can be applied immediately.
4 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Dec  4 01:45:07 2025 from 35.235.244.33
oliveramarilynshirley@marilynolivera-hackademypрактиco-host:~$
```

- Instalación de Coolify instancia de servidor de GCP: <https://coolify.io/docs/get-started/installation>

<https://coolify.io/docs/knowledge-base/server/firewall>

```
curl -fsSL https://cdn.cool labs.io/coolify/install.sh | sudo bash
```

To run the Docker daemon as a fully privileged service, but granting non-root users access, refer to <https://docs.docker.com/go/dammon-access/>

**WARNING:** Access to the remote API on a privileged Docker daemon is equivalent to root access on the host. Refer to the “Docker daemon attack surface” documentation for details: <https://docs.docker.com/go/attack-surface/>

"We messed up the kinesis again guys."

# Congratulations!

Your instance is ready to use!

You can access Coolify through your Public IPV4: <http://37.46.119.77:8000>

If your Public IP is not accessible, you can use the following Private IPs:

<http://10.0.0.1:8000>  
<http://10.0.1.1:8000>  
<http://10.0.2.46:8000>

Warning: It is highly recommended to backup your Environment variables file (`/data/coolify/source.env`) to a safe location, outside of this server (e.g. into a `password manager`).

```
oliveramarilynshirley@marilynolivera-hackademypractico-host:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
1573f4379f17 traefik:v3.6 "entrypoint.sh --pi..." 3 minutes ago Up 3 minutes (healthy) 0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp, 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
fe8aef435d3 ghc.io/coollabio/sentinel0.1.0.18 "/app/sentinel" 3 minutes ago Up 3 minutes (healthy)
910ad1809a9 ghc.io/coollabio/coolify:1.0-beta.452 "coolify" 4 minutes ago Up 3 minutes (healthy) 8000/tcp, 8443/tcp, 9000/tcp, 0.0.0.0:8000->8080/tcp, :::8000->8080/tcp
3e92d29f4e30 ghc.io/coollabio/coolify-realtime:1.0.10 "/bin/sh /socket-ent..." 4 minutes ago Up 4 minutes (healthy) 0.0.0.0:6001-6002->6001-6002/tcp, :::6001-6002->6001-6002/tcp
7f964ea59de8 redis:7-alpine "coolify-redis" 4 minutes ago Up 4 minutes (healthy) 6379/tcp
a713ca52fa0 postgres:15-alpine "coolify-db" 4 minutes ago Up 4 minutes (healthy) 5432/tcp
```

```
oliveramarilynshirley@marilynolivera-hackademypractico-host:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
1573f4379f17 traefik:v3.6 "entrypoint.sh --pi..." 3 minutes ago Up 3 minutes (healthy) 0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp, 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
fe8aef435d3 ghc.io/coollabio/sentinel0.1.0.18 "/app/sentinel" 3 minutes ago Up 3 minutes (healthy)
910ad1809a9 ghc.io/coollabio/coolify:1.0-beta.452 "coolify" 4 minutes ago Up 3 minutes (healthy) 8000/tcp, 8443/tcp, 9000/tcp, 0.0.0.0:8000->8080/tcp, :::8000->8080/tcp
3e92d29f4e30 ghc.io/coollabio/coolify-realtime:1.0.10 "/bin/sh /socket-ent..." 4 minutes ago Up 4 minutes (healthy) 0.0.0.0:6001-6002->6001-6002/tcp, :::6001-6002->6001-6002/tcp
7f964ea59de8 redis:7-alpine "coolify-redis" 4 minutes ago Up 4 minutes (healthy) 6379/tcp
a713ca52fa0 postgres:15-alpine "coolify-db" 4 minutes ago Up 4 minutes (healthy) 5432/tcp
```

- Considerando que es necesario una ip fija para coolify y hacer la conexión con el Webhook reserve la ip:

<input type="checkbox"/>	Nombre	Dirección IP	Tipo de acceso	Región	Tipo ↓	Versión
<input type="checkbox"/>	ip-host-hackademyfinal	34.46.119.77	Externo	us-central1	Estática	IPv4

- Abro el puerto 8000 para poder ingresar y configurar el coolify

Políticas de firewall [+ Crear política de firewall](#) [+ Crear regla de firewall](#) [Más](#)

[Actualizar](#) [Configurar registros](#) [Borrar](#)

Filtro Escribir el nombre o valor de la propiedad [?](#)

<input type="checkbox"/>	Nombre	Tipo	Destinos	Filtros	Protocolos/puertos	Acción	Prioridad
<input type="checkbox"/>	<a href="#">allow-coolify-8000</a>	Entrada	http-server	Rangos de IP:	tcp:8000	Permitir	1000

<http://34.46.119.77:8000>

- Reinicio de la instancia de nuevo y accedi a coolify para iniciar su configuración de la cuenta:

- Cree la cuenta:

**Choose Server Type**  
Select where to deploy your applications and databases. You can add more servers later.

<b>This Machine</b> Deploy on the server running Coolify. Best for testing and single-server usage.	<b>RECOMMENDED</b> <b>Remote Server</b> Connect via SSH to any server—cloud VPS, bare metal, or home infrastructure.	<b>RECOMMENDED</b> <b>Hetzner Cloud</b> Deploy servers directly from your Hetzner Cloud account.
--	--	--

**TECHNICAL DETAILS**  
Servers run your applications, databases, and services (collectively called resources). All CPU-intensive operations run on the target server.

**Localhost:** The machine running Coolify. Not recommended for production workloads due to resource contention.

**Remote Server:** Any SSH-accessible server—cloud providers (AWS, Hetzner, DigitalOcean), bare metal, or self-hosted infrastructure.

**Project Setup**  
Create your first project to organize applications, databases, and services.

[Create "My First Project"](#)

**TECHNICAL DETAILS**  
**Project Organization:** Group related resources (apps, databases, services) into logical projects.  
**Environments:** Each project includes a production environment by default. Add staging, development, or custom environments as needed.  
**Team Access:** Projects inherit team permissions and can be managed collaboratively.

**Setup Complete!**  
Your server is connected and ready. Start deploying in minutes.

**WHAT'S CONFIGURED**

- ✓ Server: localhost host.docker.internal
- ✓ Project: My first project Production environment ready
- ✓ Docker Engine Installed and running

[Deploy Your First Resource](#)

- **Asignación de dominio:** mi pareja posee un dominio, por lo que usará un subdominio con su nombre, lo maneja con cloudflare, así ya no accedemos desde la IP directamente y se le puede aplicar https: <https://www.youtube.com/watch?v=m68uOVaiFqU&t=488s>

1. Desde Coolify, en Settings podemos configurar el dominio: <https://coolify.alexander.net.ar>

The screenshot shows the Coolify Settings interface. The 'General' tab is selected. It displays the domain as 'https://coolify.alexander.net.ar' and the instance's public IPv4 as '2001:db8:1'. The instance's public IPv6 is also shown as '2001:db8:1'. The instance's timezone is set to 'America/Argentina/Buenos\_Aires'.

2. Agregar en cloudflare el dominio y la ip:

The screenshot shows the Cloudflare DNS settings page. An A record named 'coolify' is listed, pointing to the IP address 34.46.119.77. The status is 'Proxied' and TTL is set to 'Auto'.

3. Ahora se puede acceder desde <https://coolify.alexander.net.ar> y puedo cerrar el puerto 8000:

The screenshot shows two windows. On the left, a browser window displays an error message: 'This site can't be reached' with code 'ERR\_CONNECTION\_TIMED\_OUT'. On the right, the Coolify login page is shown, featuring fields for 'Correo electrónico' and 'Contraseña', and a link for '¿Has olvidado tu contraseña?'. Below the login form, a note says 'El registro está deshabilitado. Por favor, contacte al administrador.'

## - **Integro Coolify a GitHub Actions:** <https://coolify.io/docs/applications/ci-cd/github/github-actions>

Son necesarios dos valores para poder hacer la conexión con Coolify: Webhook y un API token

- Configuración en Coolify (Obtener el Webhook):

1. Creo en Coolify un nuevo recurso que espera recibir una imagen de docker lista, desde el aviso de github, para desplegarla en el host, para esto se debe: Crear un nuevo recurso -> Tipo Imagen de Docker(indica lo que va a recibir) -> Configurar los datos de imagen que puede recibir(el nombre que elegí antes)

The screenshot shows the Coolify dashboard. In the top right, there's a 'New Resource' section for creating a Docker-based application. Below it, a 'Dashboard' section shows a 'Projects' list with 'MarilynOlivera - ExamenPracticoHackademy'. A 'Docker Image' resource is being created, with the 'Image Name' set to 'ghcr.io/marilynolivera/vulnerable-web-app' and the 'Tag (optional)' set to 'latest'. To the right, a detailed view of the 'Docker Image' resource shows its configuration, including the image name and tag.

Esto lo mencioné en mi primer intento, el nombre de la imagen debe seguir la estructura estándar de los registros de contenedores: registry.hostname / owner / repository-name\

El nombre debe ir todo en minúscula, le puse mi nombre ya que hice un fork:

<https://ghcr.io/marilynolivera/vulnerable-web-app>

2. Webhook: Esta URL es la dirección a la que GitHub Actions enviará la señal POST para decirle a Coolify "Nueva imagen lista, desplegala". Se debe ingresar al área de webhook de el recurso:

The screenshot shows the Coolify Resources interface. At the top, there are buttons for 'Resources' (+ New), 'Clone', and 'Delete Environment'. Below this, a breadcrumb navigation shows 'MarilynOlivera - ExamenPracticoHackademy > production'. A search bar is present. Under the heading 'Applications', a card displays a deployed Docker image named 'docker-image-ds8gs0gsw0g44oooswggc8o8'. The URL 'http://ds8gs0gsw0g44oooswggc8o8.34.46.119.77.sslip.io' is listed below the card. On the left, a sidebar lists 'General', 'Advanced', 'Environment Variables', 'Persistent Storage', 'Servers', 'Scheduled Tasks', and 'Webhooks' (which is selected). On the right, under 'Webhooks', there is a section for 'Deploy Webhook (auth required)' with the URL 'https://coolify.alexander.net.ar/api/v1/deploy?uuid=ds8gs0gsw0g44oooswggc8o8&force=false'. Below this, there is a section for 'Manual Git Webhooks' with links for GitHub and GitLab.

<https://coolify.alexander.net.ar/api/v1/deploy?uuid=ds8gs0gsw0g44oooswggc8o8&force=false>

3. Creamos un token API de Coolify: <https://coolify.io/docs/api-reference/authorization>

Para poder crearla se debe activar API Access en API Settings

Luego se debe crear una en Keys&Tokens, en el apartado API Tokens, hay que darle permisos de deploy:

The screenshot shows the Coolify API Tokens creation page. It starts with a note: 'Tokens are created with the current team as scope.' Below this, a 'New Token' form is shown with a 'Description' field containing 'CoolifyGitActions' and a 'Create' button. Under 'Permissions', the 'deploy' option is selected. In the 'Token Permissions' section, the 'deploy' permission is checked. A note says 'Please copy this token now. For your security, it won't be shown again.' Below this is a redacted token value. The 'Issued Tokens' section shows a single token with the same details and a 'Revoke token' button.

Como mencioné en mi primer intento con estos datos ya se puede hacer la implementación en github action, es necesario enviar una solicitud GET a ese punto final del webhook (autenticado con el token) para activar la implementación con este formato:

```
curl --request GET "${% raw %}{{ secrets.COOLIFY_WEBHOOK }}{{ endraw %}}" --header "Authorization: Bearer ${% raw %}{{ secrets.COOLIFY_TOKEN }}{{ endraw %}}"
```

- Reconfigurando pipeline, de despliegue en Oracle a despliegue en GCP:

4. Actualizo los datos en variables secretas del repositorio: COOLIFY\_TOKEN COOLIFY\_WEBHOOK

The screenshot shows the GitHub Repository secrets page. It lists two secrets: 'COOLIFY\_TOKEN' and 'COOLIFY\_WEBHOOK', both of which were updated 'now'. Below this is a 'Confirm access' dialog box from GitHub, asking for verification. The dialog shows the user's email and a redacted password field, with a 'Verify' button at the bottom.

Tengo el MFA activado.

5. En mi primer intento mencionó que se debe definir un token mas, un Personal Access Token (PAT) de GitHub que se usa para autenticarse en GitHub Container Registry(GHCR), necesario para subir

imágenes Docker a mi registro privado/público: un GH\_TOKEN, lo creé y lo guardé en los secrets de el repositorio, a este valor no debo actualizarlo:

Name	Last updated
COOLIFY_TOKEN	2 minutes ago
COOLIFY_WEBHOOK	2 minutes ago
GH_TOKEN	yesterday

write:packages  
 read:packages

Upload packages to GitHub Package Registry  
Download packages from GitHub Package Registry

- Prueba de funcionamiento, en mi primer intento me base en un template de ejemplo que da coolify y lo adapte (detalles en esa implementación):

<https://github.com/andrasbacsai/github-actions-with-coolify/blob/main/.github/workflows/build.yaml>

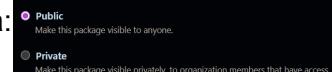
Como ya actualice las claves de coolify no debo realizar modificaciones en el código del pipeline:

```
.github > workflows > 📄 deploymen
1   # For more details, read this: https://coolify.io/docs/github-actions
2   name: Despliegue app - Examen Practico Hackademy
3   on:
4     push:
5       branches: ["master"]
6     workflow_dispatch: #permite ejecucion manual
7
8   env:
9     REGISTRY: ghcr.io
10    IMAGE_NAME: "marilynolivera/vulnerable-web-app"
11
12  jobs:
13    build:
14      runs-on: ubuntu-latest
15      permissions:
16        contents: read
17        packages: write
18      steps:
19        - uses: actions/checkout@v3
20        - name: Login to ghcr.io
21        - uses: docker/login-action@v2
22        - with:
23          registry: ${{ env.REGISTRY }}
24          username: ${{ github.actor }}
25          password: ${{ secrets.GH_TOKEN }}
26        - name: Build image and push to registry
27        - uses: docker/build-push-action@v4
28        - with:
29          context: .
30          file: Dockerfile
31          platforms: linux/amd64
32          push: true
33          tags: ${{ env.REGISTRY }}/${{ env.IMAGE_NAME }}:latest
34        - name: Deploy to Coolify
35        - run:
36          curl --request GET '${{ secrets.COOLIFY_WEBHOOK }}' --header 'Authorization: Bearer ${{ secrets.COOLIFY_TOKEN }}'
```

Este pipeline automatiza el despliegue:

- Se sube código a GitHub
- Se construye una imagen de Docker
- Se sube al registro de GitHub (GHCR)
- Automáticamente se notifica a Coolify
- Automáticamente Coolify despliega la nueva versión

- En mi primer intento tuve problema con el acceso de coolify a la imagen guardada en GHCR, ya que ahora por default son privadas, la hice pública:



### - Ejecución del pipeline:

Lo corri y ahora sí se levantó el contenedor:

marilynolivera / vulnerable-web-app

Despliegue app - Examen Practico Hackademy

A GCP, todo vuelve a ser como debe ser #18

Summary

build succeeded 12 minutes ago in 4s

build

Set up job  
Run actions/checkout@v3  
Login to ghcr.io  
Build image and push to registry  
Deploy to Coolify  
Post Build image and push to registry  
Post Login to ghcr.io  
Post Run actions/checkout@v3  
Complete job

```
oliveramarilynshirley@marilynolivera-hackademypractico-host:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d61235fbc349  ghcr.io/marilynolivera/vulnerable-web-app:latest  "docker-entrypoint.s..." 2 minutes ago Up 2 minutes  80/tcp, 8080/tcp, ds8gs0gsw0g4oooswggc8o8-180350373477
bcc10786da81  ghc...io/coollabsio/sentinel:0.0.18  "/app/sentinel" 2 hours ago Up 2 hours (healthy)  coolify-sentinel
1573f4379f17  traefik:v3.6  "/entrypoint.sh --pi..." 16 hours ago Up 2 hours (healthy)  0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, ::443->443/udp, coolify-proxy
910a2d1809a9  ghcr.io/coollabsio/coolify:4.0.0-beta.452  "docker-php-serversi..." 16 hours ago Up 2 hours (healthy)  8000/tcp, 8443/tcp, 9000/tcp, 0.0.0.0:8000->8000/tcp, :::8000->8000/tcp
3e92d29f4e30  ghc...io/coollabsio/coolify-realtime:1.0.10  "/bin/sh /soketi-ent..." 16 hours ago Up 2 hours (healthy)  0.0.0.0:6001-6002->6001-6002/tcp, coolify-realtime
7f964ea59de8  redis:7-alpine  "docker-entrypoint.s..." 16 hours ago Up 2 hours (healthy)  6379/tcp, coolify-redis
a713ca92faa0  postgres:15-alpine  "docker-entrypoint.s..." 16 hours ago Up 2 hours (healthy)  5432/tcp, coolify-db
```

Para acceder a la aplicación vulnerable se debe usar el dominio temporal que generó Coolify para el recurso:

- En este momento entrando a <http://ds8gs0gsw0g4400oswggc8o8.34.46.119.77.sslip.io> accedo a la aplicación:



- **Configuración de dominio:** <https://coolify.io/docs/knowledge-base/cloudflare/tunnels/overview>  
De forma similar a como configuro el túnel para acceder a Coolify debo configurar este, solo que se hace sobre la configuración del recurso:

1. Configure en Cloudflare mi dominio deseado:

Type	Name	Content	Proxy status	TTL	Actions
A	marilynolivera	34.46.119.77	Proxied	Auto	Edit
A	coolify	34.46.119.77	Proxied	Auto	Edit

2. Configuró en el recurso el dominio con mi nombre en lugar del temporal:

### 3. Guardé y reinicié:

The screenshot shows the Docker Deployment Log for a container named 'vulnerable-web-app'. The log output is as follows:

```
2025-Dec-04 19:06:28.066283 Starting deployment of ghr.io/marilynolivera/vulnerable-web-app:latest to localhost.
2025-Dec-04 19:06:28.514714 Preparing container with helper image: ghr.io/coollabsio/coolify-helper:1.0.12
2025-Dec-04 19:06:23.738866 -----
2025-Dec-04 19:06:23.758512 Rolling update started.
2025-Dec-04 19:06:23.928913 Pulling latest images from the registry.
2025-Dec-04 19:06:26.418894 New container started.
2025-Dec-04 19:06:26.458864 Removing old containers.
```

Ahora desde <https://marilynolivera.alexander.net.ar> accedo a la aplicación vulnerable que despliegue:

The browser window shows the URL [marilynolivera.alexander.net.ar](https://marilynolivera.alexander.net.ar). The page title is 'EkoParty Hackademy-Vulnerable Web App'. The main content area contains the following text:

Cada Link contiene vulnerabilidades intencionalmente. Juega con cada uno para tener una idea de cómo funcionan.

- [Reflected XSS Example 1](#)
- [Reflected XSS Example 2](#)
- [Reflected XSS Example 3](#)
- [Stored XSS Example](#)
- [CSRF Example](#)
- [Fuzzing](#)
- [Authentication Bypass](#)
- [Directory Traversal](#)
- [Insecure Direct Object Reference \(IDOR\)](#)
- [Injections and remote code execution](#)
- [Mixed topics](#)

The browser window shows the URL [ds8gs0gsw0g4400oswggc8o8.34.46.119.77.sslip.io](https://ds8gs0gsw0g4400oswggc8o8.34.46.119.77.sslip.io). The page displays a '404 page not found' error message.

### - A experimentar con las vulnerabilidades: “Finalmente hora de jugar”

- [Reflected XSS Example 1](#)
- [Reflected XSS Example 2](#)
- [Reflected XSS Example 3](#)
- [Stored XSS Example](#)
- [CSRF Example](#)
- [Fuzzing](#)
- [Authentication Bypass](#)
- [Directory Traversal](#)
- [Insecure Direct Object Reference \(IDOR\)](#)
- [Injections and remote code execution](#)
- [Mixed topics](#)

#### 1. Reflected XSS Example 1: <https://portswigger.net/web-security/cross-site-scripting/reflected>

The browser window shows the URL [marilynolivera.alexander.net.ar/reflected\\_xss?foobar>Hello%20world!](https://marilynolivera.alexander.net.ar/reflected_xss?foobar>Hello%20world!). The page displays the text 'Hello!' followed by 'The following text is reflected from the url: Hello world!'

The screenshot shows a browser window with the URL `marilynolivera.alexander.net.ar/reflected_xss?foobar=Bye%20world!`. The page content is "Hello!" followed by "The following text is reflected from the url: Bye world!". This demonstrates a reflected XSS attack where the user's input is directly reflected back to them.

Cambiando la URL cambia al comportamiento de la página, ya que tiene los parámetros expuestos de los que depende para su comportamiento, en la URL, como parte de la solicitud HTTP insegura.

## 2. Reflected XSS Example 2:

The screenshot shows a browser window with the URL `marilynolivera.alexander.net.ar/reflected_xss_2?foo=bar`. The page content is "Not all XSS injections are completely obvious. You may need to view the page source, or use special characters to break out of existing html.". This is a common message used in XSS examples to encourage users to inspect the page source.

→ No todas las inyecciones XSS son completamente obvias. Quizás necesites ver el código fuente de la página o usar caracteres especiales para acceder al HTML existente.

Esto contiene el código fuente del html: Se utiliza la variable payload para determinar la class del div, esta variable es recibida a través del parámetro foo de la solicitud HTTP.

The screenshot shows a browser window with the URL `marilynolivera.alexander.net.ar/reflected1.html`. The page content is the source code: "`<div class="{{payload}}>Not all XSS injections are completely obvious. You may need to view the page source, or use special characters to break out of existing html.`". This shows how the user's input is reflected back into the page's HTML structure.

Es decir, podría modificar el estilo del div e incluso cambiar el comportamiento o agregar más parámetros, ya que se pueden agregar etiquetas debido a que se espera un “ para cerrar la declaración de los parámetros y así agregar código ejecutable, todo desde la URL, considerando que los espacios, comillas y mas caracteres especiales tienen representaciones especiales:

- Espacios: %20 - Comillas: %22 y así es con varios valores mas que no son interpretables.  
<https://help.smartsheet.com/es/articles/2478871-url-query-string-form-default-values>

bar%22%3Cdiv%3EMarilyn%20paso%20por%20aca%20%3AD%3C%2Fdiv%3E

The screenshot shows a browser window with the URL `marilynolivera.alexander.net.ar/reflected_xss_2?foo=bar`. The page content is "Marilyn paso por aca :D". Below it, the message "Not all XSS injections are completely obvious. You may need to view the page source, or use special characters to break out of existing html." is visible. This demonstrates how a reflected XSS attack can be used to inject arbitrary text into a page.

Así quedó con un efecto porque muestra como texto imprimible “> después de usar el parametro en la clase del div jaja.

Hay varias cosas que se pueden hacer <https://portswigger.net/web-security/cross-site-scripting/cheat-sheet>. La IA me sugirió la siguiente, que usa onmouseover, un parámetro usado para ejecutar su contenido cuando el puntero del mouse se mueve sobre el elemento al que está asociado, en este caso hace ejecuta un alert cada vez que se pase el mouse por la cadena contenida en el div:

%22%20onmouseover%3D%22alert%28%27XSS%20Explotado%20por%20Marilyn%27%29%22%20X%3D%22

The screenshot shows a browser window with the URL `marilynolivera.alexander.net.ar/reflected_xss_2?foo=bar%20onmouseover%3D"alert%28%27XSS%20Explotado%20por%20Marilyn%27%29%20X%3D"`. The page content is "Not all XSS injections are completely obvious. You may need to view the page source, or use special characters to break out of existing html.". This shows how an onmouseover event can be triggered by the user's input to execute JavaScript.

The screenshot shows a browser window with the URL `marilynolivera.alexander.net.ar/reflected_xss_2?foo=bar%20onmouseover%3D"alert%28%27XSS%20Explotado%20por%20Marilyn%27%29%20X%3D"`. A tooltip appears over the text, showing "marilynolivera.alexander.net.ar says XSS Explotado por Marilyn". This is a common way for XSS filters to warn users about detected attacks.

## 3. Reflected XSS Example 3:

Nos indica que en este caso hay un bloqueador de XSS y que lo verifique/compruebe:

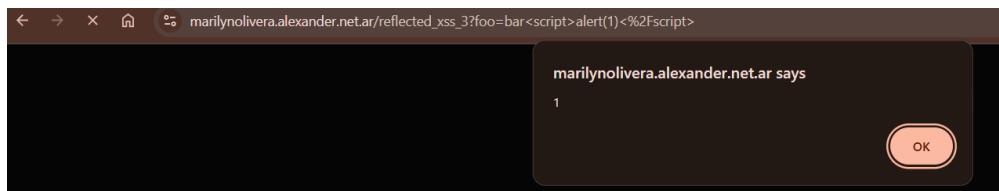
The screenshot shows a browser window with the URL `marilynolivera.alexander.net.ar/reflected_xss_3?foo=bar`. The page content is "Many browsers have built-in XSS auditors that attempt to block reflected XSS attacks. Try reflecting <script>alert(1)</script> into the page, and you may find that it doesn't work. If you check the browser's javascript console, a message will be displayed, noting that it block an attempted script execution."

```

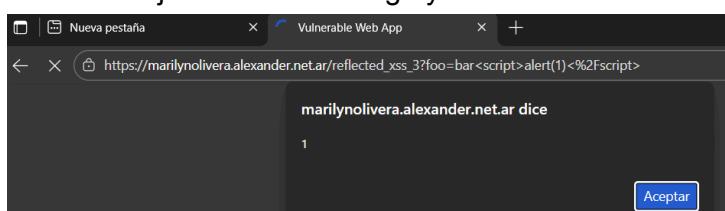
views >反映了2.html > ...
1 Many browsers have built-in XSS auditors that attempt to block reflected XSS attacks.
2 <br><br>
3 Try reflecting &lt;script&ampgtalert(1)&lt;/script&ampgt into the page, and you may find that it doesn't work.
4 <br><br>
5 If you check the browser's javascript console, a message will be displayed, noting that it blocks an attempted script execution.
6
7 <div style="display:none;">{{payload}}</div>
8

```

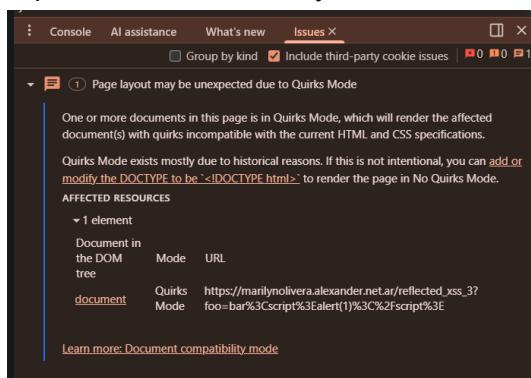
Intento reflejar desde Chrome <script>alert(1)</script>: bar%3Cscript%3Ealert(1)%3C%2Fscript%3E



Intente reflejar también en edge y funciona:



- En la consola de chrome en el área de issues se indica que la aplicación no cuenta con la declaración <!DOCTYPE html>, sino que está en Modo Quirks, que existe principalmente por razones históricas, lo que hará que los documentos afectados con peculiaridades sean incompatibles con las especificaciones HTML y CSS actuales. Esto podría explicar porque la defensa no lo detecta:



→ Puedo probarlo añadiendo la línea <!DOCTYPE html> al inicio de el archivo HTML de la aplicación, lo que activará el pipeline y desplegará la actualización automáticamente:

[https://marilynolivera.alexander.net.ar/reflected\\_xss\\_3?foo=bar%3Cscript%3Ealert\(1\)%3C%2Fscript%3E](https://marilynolivera.alexander.net.ar/reflected_xss_3?foo=bar%3Cscript%3Ealert(1)%3C%2Fscript%3E)

Agregando <!DOCTYPE html>

```

views >反映了2.html > ...
1 <!DOCTYPE html>
2 Many browsers have built-in XSS auditors that attempt to block reflected XSS attacks.
3 <br><br>
4 Try reflecting &lt;script&ampgtalert(1)&lt;/script&ampgt into the page, and you may find that it doesn't work.
5 <br><br>
6 If you check the browser's javascript console, a message will be displayed, noting that it blocks an attempted script execution.
7
8 <div style="display:none;">{{payload}}</div>
9

Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 403 bytes | 201.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/marilynolivera/vulnerable-web-app.git
   c57b353..320fb3b master -> master
PS D:\Repos\Hackademy\ExamenPractico\vulnerable-web-app>

```

● Agregando <!DOCTYPE html>  
 Despliegue app - Examen Practico Hackademy #19: Commit 320fb3b pushed by marilynolivera

master now  
In progress

← Despliegue app - Examen Practico Hackademy  
 ✓ Agregando <!DOCTYPE html> #19

[Summary](#)

Jobs  
 build

Run details  
 Usage  
 Workflow file

Triggered via push 1 minute ago	Status	Total duration	Artifacts
marilynolivera pushed -> 320fb3b master	Success	39s	-

deploy.yml  
 on: push

build 36s

Success

Started: 2025-12-05 21:02:18 UTC  
 Ended: 2025-12-05 21:03:08 UTC  
 Duration: 00m 50s  
 Finished 7 minutes ago  
 Commit: HEAD Manual

```

Code Blame 8 lines (7 loc) · 403 Bytes
1  <!DOCTYPE html>
2  Many browsers have built-in XSS auditors that attempt to block reflected XSS attacks.
3  <br><br>
4  Try reflecting &lt;script&gt;alert(1)&lt;/script&gt; into the page, and you may find that it doesn't work.
5  <br><br>
6  If you check the browser's javascript console, a message will be displayed, noting that it block an attempted script execution.
7
8  <div style="display:none;">{{payload}}</div>

```

Sigue sin detectarlo.

→ Estuve buscando informacion y no logré hacer que funcione:

<https://www.invicti.com/blog/web-security/xss-auditors>

<https://socradar.io/blog/securing-browser-cross-site-scripting-xss-attacks/>

Según la respuesta que da la IA de google al buscar o están activos:

No, los navegadores modernos ya no tienen funciones integradas. Auditores o filtros XSS Los principales proveedores de navegadores, como Google, Microsoft y Apple, descontinuaron esta función debido a que los ataques solían evadir estos auditores y podían introducir nuevas vulnerabilidades de seguridad y filtraciones de información entre sitios.

Historial y eliminación

- Internet Explorer fue el primero en introducir un filtro XSS en 2008.
- Google Chrome agregó su XSS Auditor en 2010, pero se eliminó permanentemente en la versión 78 de Chrome en 2019.
- Microsoft Edge también eliminó su filtro XSS en favor de estándares de seguridad modernos.
- Firefox nunca implementó una característica similar.

Defensa moderna

En lugar de auditores del lado del navegador, la responsabilidad de prevenir ataques de secuencias de comandos entre sitios (XSS) ahora recae principalmente en los desarrolladores web a través de una implementación sólida del lado del servidor y el uso de estándares de seguridad modernos.

Aparentemente es cuestión de que el servidor le envía una indicación al browser de que bloquee la solicitud porque detecta XSS

Revertir el cambio en la aplicación:

```

PS D:\Repos\Hackademy\ExamenPractico\vulnerable-web-app> git log
commit 320fb3b7b5995d3ff5995518aa60670b5d28eb28 (HEAD -> master, origin/master, origin/HEAD)
Author: Marilyn Olivera <105252740+marilynolivera@users.noreply.github.com>
Date:   Fri Dec 5 18:01:33 2025 -0300

    Agregando <!DOCTYPE html>

```

```

Revert "Agregando <!DOCTYPE html>"

This reverts commit 320fb3b7b5995d3ff5995518aa60670b5d28eb28.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#       modified:   views/reflected2.html
#
# Changes not staged for commit:
#       modified:   .gitignore
#

```

← Aca agregue un comentario

```

PS D:\Repos\Hackademy\ExamenPractico\vulnerable-web-app> git revert 320fb3b7b5995d3ff5995518aa60670b5d28eb28
[master c599d80] Revert "Agregando <!DOCTYPE html>" Revertir cambio <!DOCTYPE html> This reverts commit 320fb3b7b5995d3ff5995518aa60670b5d28eb28.
1 file changed, 1 deletion(-)
PS D:\Repos\Hackademy\ExamenPractico\vulnerable-web-app> git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 425 bytes | 425.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/marilynolivera/vulnerable-web-app.git
  320fb3b..c599d80  master -> master
PS D:\Repos\Hackademy\ExamenPractico\vulnerable-web-app>

```

Se sube y corre el pipeline de nuevo para hacer la actualización y dejarlo como estaba:

En coolify:

#### 4. Stored XSS Example:

"Este es un saludo para todos los usuarios que visitan esta página. Cámbielo en el recuadro de abajo."

- Aunque recargue la página el mensaje es el mismo, esto significa que enviado desde ese cuadro de texto se guarda en una base de datos y es consultado cada vez que se ingresa en la página.  
(Cross-Site Scripting Almacenado o Persistente)  
Probé con: <script>alert('Stored XSS,( ^.\_.^')</script>

Cada vez que recargo aparece la alerta, aunque no tiene mensaje, si agregara: Nada raro está pasando 0u0<script>alert('Stored XSS,( ^.\_.^')</script>, quedaría con un mensaje.

## 5. CSRF Example: (Cross-Site Request Forgery)

This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker could figure out how to change it, your payments would be sent to them instead.

Change your account #  Submit

Your current account number is: 1234567

If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:  
<http://codepen.io/anon/pen/XXgeqP>

This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker could figure out how to change it, your payments would be sent to them instead.

Change your account #  Submit

Your current account number is: 7777777

If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:  
<http://codepen.io/anon/pen/XXgeqP>

¡Este formulario cambia tu número de cuenta bancaria! Todos los pagos que recibas a través de este sitio se enviarán a este número de cuenta. Si un atacante descubre cómo cambiarlo, tus pagos se le enviarán a él.

Tu número de cuenta actual es: 1234567.

Si te quedas atascado, visita esta página de Codepen. Utiliza un ataque CSRF válido y cambiará tu número de cuenta al cargar la página: <http://codepen.io/anon/pen/XXgeqP>

- LA falsificación de solicitudes entre sitios (CSRF) es un ataque que obliga a un usuario final a ejecutar acciones no deseadas en una aplicación web en la que está autenticado.

<https://owasp.org/www-community/attacks/csrf>

- En la mayoría de los sitios, las solicitudes del navegador incluyen automáticamente las credenciales asociadas al sitio, por lo tanto, si el usuario está autenticado en el sitio, este no podrá distinguir entre la solicitud falsificada enviada por la víctima y una solicitud legítima.
- En conclusión, un ataque CSRF obliga a un navegador web a enviar una solicitud no deseada y dañina a una aplicación en la que el usuario ya está autenticado.

Este es el código de la app:

```
views > csrf.html > ...
1 This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker
2
3 <br><br>
4
5 <form action="/csrf" method="POST">
6   <input name="account_number" placeholder="Change your account #">
7   <input type="submit">
8 </form>
9
10 <br><br>
11 Your current account number is: {{account_number}}
12
13 <br><br><br>
14 If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:
15 <br><br>
16 <a href="http://codepen.io/anon/pen/XXgeqP" target="blank">http://codepen.io/anon/pen/XXgeqP</a>
```

Vemos que se usa un método post, y se usa la variable account\_number.

- Información que provee la fuente de la aplicación: <http://codepen.io/anon/pen/XXgeqP>

Untitled  
Captain Anonymous

HTML

```
1 <form action="http://localhost:6969/csrf"
2   method="POST">
3     <input name="account_number" value="Attacker
4       account number">
5     <input type="submit">
6   </form>
7
8 <script>
9   document.forms[0].submit()
10 </script>
```

CSS

```
1
```

JS

```
1
```

Nos provee de un formulario para hacer el ataque CSRF, que imita el real, debería modificarlo para que en vez de recibir un valor ponga el que quiero, este formulario envía automáticamente el valor que queramos (<script> document.forms[0].submit() </script>), supongamos que mi numero de cuenta es: 7788899, la url de la pagina real es <https://marilynolivera.alexander.net.ar/csrf>

```

1 <form
2   action="https://marilynlivera.alexander.net.ar/csrf"
3   method="POST">
4     <input name="account_number" value="7788899">
5     <input type="submit">
6   </form>
7
8 <script>
9   document.forms[0].submit()
10</script>

```

This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker could figure out how to change it, your payments would be sent to them instead.

Your current account number is: 1234567

If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:

<http://codepen.io/anon/pen/XXgeqP>

- Para hacer CSRF el usuario legítimo debe tener iniciada la sesión en la aplicación. En el mismo navegador el atacante debe lograr que se abra la página imitación con el script malicioso:

- Tengo abierta mi sesión legítima

This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker could figure out how to change it, your payments would be sent to them instead.

Your current account number is: 1234567

If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:

<http://codepen.io/anon/pen/XXgeqP>

- Para cargar la página con el script malicioso debo exportar el html de codepen y luego abrir el html que correrá el script y simulará ser la misma sesión redirigiendo a la aplicación real, lo adjunto en con el nombre CSRF-html.zip:

Nombre	Tipo	Tamaño comprimido	Protegido ...	Tamaño	Relación	Fecha de modificación
index.html	Chrome HTML Document	1 KB	No	1 KB	0%	9/12/2025 18:34

Entonces vemos:

- Sesión original

This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker could figure out how to change it, your payments would be sent to them instead.

Your current account number is: 1234567

If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:

<http://codepen.io/anon/pen/XXgeqP>

- Nueva sesión maliciosa, rápidamente corre el script y redirige mostrando el nuevo número de cuenta:

Vulnerable Web App

MarilynHackademy

File C:/Users/mari/AppData/Local/Temp/28041cb2-a574-409c-8151-f2793edcf38b\_marilynhackademy.zip.38b/marilynhackademy/dist/index.html

7788899

This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker could figure out how to change it, your payments would be sent to them instead.

Your current account number is: 7788899

If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:  
<http://codepen.io/anon/pen/XXgeqP>

- En sesión original recargamos y ya vemos reflejado el cambio:

This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker could figure out how to change it, your payments would be sent to them instead.

Your current account number is: 123567

If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:  
<http://codepen.io/anon/pen/XXgeqP>

This form changes your bank account number! Any payments you receive through this site will be sent to this account number- if an attacker could figure out how to change it, your payments would be sent to them instead.

Your current account number is: 7788899

If you're stuck, visit this codepen page. It uses a valid CSRF attack, and will change your account number the moment you load the page:  
<http://codepen.io/anon/pen/XXgeqP>

- Esto es posible porque los navegadores envían automáticamente credenciales (como cookies de sesión) con cada solicitud a un sitio, y las aplicaciones web que no validan adecuadamente estas solicitudes o que confían demasiado en la identidad del usuario permiten que un atacante engañe al navegador.

## 6. Fuzzing: <https://owasp.org/www-project-web-security-testing-guide/latest/6-Appendix/C-Fuzzing> <https://www.zaproxy.org/blog/2020-05-15-dynamic-application-security-testing-with-zap-and-github-actions/> <https://github.com/marketplace/actions/zap-full-scan>

Fuzzing is an important tool in penetration testing; the core idea is that by passing unexpected inputs to a server, you can get it to perform in potentially exploitable ways.

This particular page demonstrates one case in which fuzzing is useful: servers with poor error handling.

Try inserting a few special characters into the URL, and the server will spit out private server information in the form of a stack trace.

<https://www.freecodecamp.org/news/web-security-fuzz-web-applications-using-ffuf/>

- El fuzzing es el proceso o técnica que consiste en enviar varias solicitudes a un sitio objetivo en un intervalo de tiempo determinado, es similar a la fuerza bruta. Se puede lograr con herramientas como Wfuzz, ffuf, etc. Es un proceso automatizado donde una herramienta se encarga de gran parte del trabajo pesado. Un analista sólo tiene que analizar las distintas características una vez finalizado el proceso.

**ZAP**

ZAP is a web application security scanner that can be used to find vulnerabilities and weaknesses in web applications. It also includes a Fuzzer.

One of the key features of ZAP is its ability to perform both passive and active scans. Passive scans involve observing the traffic between the user and the web application, while active scans involve sending test payloads to the web application to identify vulnerabilities.

Erróneamente creí que ZAP proxy podría servir e implemente un workflow, la documentación se encuentra en las partes finales del reporte.

→ Busque con que fuzzer podía probar: **FFUF** <https://bishopfox.com/blog/top-9-fuzzers>  
<https://github.com/ffuf/ffuf/blob/master/README.md>

- Viene preinstalado en kali linux, yo tengo una vm de kali, asi que probe con esta:

```
(kali㉿kali)-[~/Desktop]
$ ffuf
Encountered error(s): 2 errors occurred.
  * -u flag or -request flag is required
  * Either -w or --input-cmd flag is required
Fuzz Faster U Fool - v2.1.0-dev
```

Además para hacer fuzzing se usa una lista de palabras (wordlist) que contenga payloads e inyecciones comunes: kali linux viene con seclists integradas <https://www.kali.org/tools/seclists/c> pero en mi kali no está disponible y no me deja instalarlo directamente (apt -y install seclists). Podía descargar seclist, <https://github.com/danielmiessler/SecLists>, estaba demorando, por lo que decidí probar otra cosa.

- Me descargue la consola de kali que ahora está disponible desde.

Tuve que instalar el ffuf:

```
(marilyn@ASUS-TUF-FX505)-[~]
$ sudo apt install ffuf
Installing:
ffuf

Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 130
Download size: 3,029 kB
Space needed: 9,566 kB / 1,025 GB available
```

```
(marilyn@ASUS-TUF-FX505)-[~]
$ ffuf
Encountered error(s): 2 errors occurred.
  * -u flag or -request flag is required
  * Either -w or --input-cmd flag is required
Fuzz Faster U Fool - v2.1.0-dev
```

Ahora si pude instalar directamente el seclist:

```
(marilyn@ASUS-TUF-FX505)-[~]
$ sudo apt install seclists
Installing:
seclists

Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 130
Download size: 545 MB
Space needed: 1,935 MB / 1,025 GB available
```

<https://www.kali.org/tools/seclists/>

- Ya instalado podemos ver las colecciones que tiene:

```
(marilyn@ASUS-TUF-FX505)-[~]
$ seclists -h
> seclists ~ Collection of multiple types of security lists

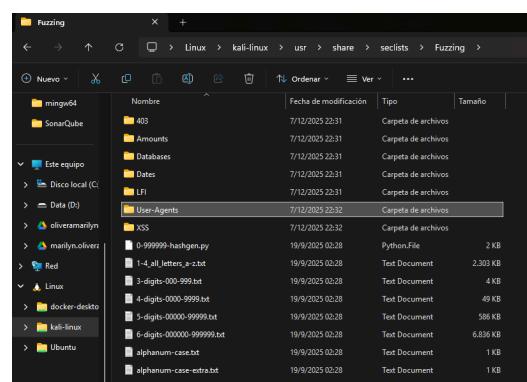
/usr/share/seclists
├── Discovery
├── Fuzzing
├── Miscellaneous
├── Passwords
├── Pattern-Matching
├── Payloads
├── Usernames
└── Web-Shells
```

Esta es la wordlist que debo usar: /usr/share/seclists/Fuzzing

Usando el comando: tree -d /usr/share/seclists/

```
Tree -d /usr/share/seclists/Fuzzing
.
├── 403
├── Amounts
├── Databases
│   └── SQLi
├── Dates
│   ├── 2019
│   ├── 2020
│   ├── 2021
│   ├── 2022
│   ├── 2023
│   ├── 2024
│   └── 2025
├── LFI
├── User-Agents
│   ├── hardware-type-specific
│   ├── layout-engine-name
│   ├── operating-platform
│   ├── operating-system-name
│   ├── software-name
│   └── software-type-specific
└── XSS
    ├── human-friendly
    ├── Polyglots
    └── robot-friendly
```

Pude ver el contenido de la colección y las opciones. Tambien puedo verlas desde el explorador de archivos donde puedo leerlos fácilmente.



- Uso de la guia del repositorio de ffuf: <https://github.com/ffuf/ffuf#ffuf-parameter-fuzzing>. Adjunto resultados en “Fuzzing.7z”.
- > Análisis de errores de parámetros GET

ffuf -w /usr/share/seclists/Fuzzing/ALGO.txt -u https://marilynlivera.alexander.net.ar/fuzzing/fuzz.html?FUZZ=test\_value -fs 4242

- Con lista de caracteres especiales y urlencode

	19/9/2025 02:28	Archivo de origen ...	1 KB
README.md			
special-chars + urlencoded.txt	19/9/2025 02:28	Text Document	1 KB
special-chars.txt	19/9/2025 02:28	Text Document	1 KB

```
ffuf -w /usr/share/seclists/Fuzzing/special-chars + urlencoded.txt -u  
https://marilynolivera.alexander.net.ar/fuzzing/fuzz.html?FUZZ=test_value -fs 4242 >  
res1GETCaraceresEspeciales+urlcoded.txt
```

El servidor respondió a todo sin problema.

> Acabo de notar que la consigna dice "insertar algunos caracteres especiales en la URL", yo lo estoy haciendo un parámetro. debería usarlo de esta forma:

```
ffuf -w "/usr/share/seclists/Fuzzing/special-chars + urlencoded.txt" -u https://marilynolivera.alexander.net.ar/fuzzing/fuzzFUZZ.html > res1URLCaracceresEspeciales+urlcoded.txt
```

```
(marilyn@ASUS-TUF-FX505)-[~]
$ ffuf -w "/usr/share/seclists/Fuzzing/special-chars + urlencoded.txt" -u https://marilynlivera.alexander.net.ar/fuzzing/fuzzFUZZ.html
> res1URLCaracteresEspeciales+urlcoded.txt

          _/\_>_/\_>
         / \_/\_>\_/\_>
        / \_/\_>\_/\_>\_/\_>
       / \_/\_>\_/\_>\_/\_>\_/\_>
      / \_/\_>\_/\_>\_/\_>\_/\_>\_/\_>
     / \_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>
    / \_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>
   / \_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>
  / \_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>\_/\_>
 v2.1.0-dev

-----
:: Method           : GET
:: URL             : https://marilynlivera.alexander.net.ar/fuzzing/fuzzFUZZ.html
:: Wordlist         : FUZZ: /usr/share/seclists/Fuzzing/special-chars + urlencoded.txt
:: Follow redirects: false
:: Calibration     : false
:: Timeout          : 10
:: Threads          : 40
:: Matcher          : Response status: 200-299,301,302,307,401,403,405,500

-----
:: Progress: [60/60] :: Job [1/1] :: 41 req/sec :: Duration: [0:00:01] :: Errors: 1 ::
```

Analice el resultado, en primera instancia encuentre este en el archivo con el resultado:

[Status: 500, Size: 1479, Words: 55, Lines: 12, Duration: 257ms] [0m]  
Lo probé y efectivamente funciona y el servidor tiene un error en el manejo de errores:

Lo probé y efectivamente funciona y el servidor tiene un error en el manejo de errores:

```
TypeError: "privatekey_a0e5613a3c7f5779b47ab34657c48cf4".db_write is not a function
at /server.js:127:55
at Layer.handle [as handle_request] (/node_modules/express/lib/router/layer.js:95:5)
at next (/node_modules/express/lib/router/route.js:144:13)
at Route.dispatch (/node_modules/express/lib/router/route.js:114:3)
at Layer.handle [as handle_request] (/node_modules/express/lib/router/layer.js:95:5)
at /node_modules/express/lib/router/index.js:284:15
at param (/node_modules/express/lib/router/index.js:365:14)
at paramCallback (/node_modules/express/lib/router/index.js:412:21)
at /server.js:122:5
at paramCallback (/node_modules/express/lib/router/index.js:415:7)
```

Puedo filtrar cuales son los caracteres especiales desde la consola:

```
(marilyn@ASUS-TUF-FX505) [~]
$ grep -i "status.*500" res1URLCaracteresEspeciales+urlcoded.txt
$ [Status: 500, Size: 1479, Words: 55, Lines: 12, Duration: 1016ms]
%24 [Status: 500, Size: 1479, Words: 55, Lines: 12, Duration: 1118ms]
^ [Status: 500, Size: 1479, Words: 55, Lines: 12, Duration: 1125ms]
@ [Status: 500, Size: 1479, Words: 55, Lines: 12, Duration: 1130ms]
%40 [Status: 500, Size: 1479, Words: 55, Lines: 12, Duration: 231ms]
%5E [Status: 500, Size: 1479, Words: 55, Lines: 12, Duration: 257ms]
```

- El fuzzing expone el hash de una clave privada: privatekey\_a0e5613a3c7f5779b47ab34657c48cf4
- Se intenta escribir en una base de datos con ese string, intentando usarlo como un objeto, lo que da error: .db\_write is not a function
- Y expone el debugging:

```
at /server.js:127:55
at Layer.handle [as handle_request] (/node_modules/express/lib/router/layer.js:95:5)
at next (/node_modules/express/lib/router/route.js:144:13)
at Route.dispatch (/node_modules/express/lib/router/route.js:114:3)
at Layer.handle [as handle_request] (/node_modules/express/lib/router/layer.js:95:5)
at /node_modules/express/lib/router/index.js:284:15
at param (/node_modules/express/lib/router/index.js:365:14)
at paramCallback (/node_modules/express/lib/router/index.js:412:21)
at /server.js:122:5
at paramCallback (/node_modules/express/lib/router/index.js:415:7)
```

(Ahora entiendo que ZAP no lo detectó porque no crea URLs nuevas sino que prueba mandando parámetros en busca de fuzzing)

## 7. Authentication Bypass:

The screenshot shows a browser window with the URL [marilynlivera.alexander.net.ar/auth\\_bypass](http://marilynlivera.alexander.net.ar/auth_bypass). The page content is as follows:

Authentication bypass can occur in any scenario on a website where content or API routes are not intended to be available to you.

For example, if you were an admin you would see a button below that allows you to ban any user! I have completely removed the button from this page, but did I forget to block the API endpoint from normal users?

Maybe I should check the network inspector...

La omisión de la autenticación puede ocurrir en cualquier escenario en un sitio web donde el contenido o las rutas API no estén disponibles.

Por ejemplo, si fueras administrador, verás un botón debajo que te permite bloquear a cualquier usuario. He eliminado el botón por completo de esta página, pero ¿olvidé bloquear el punto final de la API para los usuarios normales?

Quizás debería revisar el inspector de red...

→ El enunciado indica que está disponible una llamada a una API para usuario administrador desde usuario normal, que recibimos network al recargar la página para encontrar el llamado que tiene la funcionalidad que no fue removida del todo.

- Entre lo que nos devuelve la página se encuentra banusers.js:

The screenshot shows the Network tab in the browser developer tools. A request to `banusers.js` is listed with the following details:

Name	Headers	Preview	Response	Initiator	Timing	Cookies
auth_bypass	General					
banusers.js	Request URL: https://marilynlivera.alexander.net.ar/static/banusers.js Request Method: GET Status Code: 200 OK Remote Address: [2606:4700:3031::ac43:cba]:443 Referrer Policy: strict-origin-when-cross-origin					
gti-vt-icon2.png						
js.js						
dom.js						
js.js						

Below the table, it says: 9 requests | 84.9 kB transferred.

```
//This script is for admins only. It uses a GET request to /ban/user/<username> to permanently ban a user.
//
//It would be really bad if a normal user had access to this script.

function banuser(user){
    var req = new XMLHttpRequest();
    req.open("GET", "/ban/user/" + user, true);
    req.send();
}
```

- Con la función banuser podemos llamarla, pasándole por parámetro un usuario:  

```
> banuser("NoEstoy")
< undefined
>
```

→ El llamado funciona aunque como no tiene return la función no vemos la respuesta.
- Con la ayuda de la IA entendi que podemos modificar las funciones temporalmente desde la consola

```
> banuser("NoEstoy")
< undefined
>

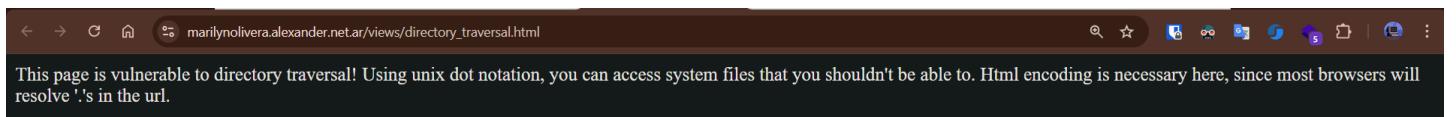
> banuser("Admin")
< undefined
Respueta <title>Vulnerable Web App</title>
<style>
    @font-face{
        font-family:'Lato-Lig';
        src: url('/static/fonts/Lato-Lig.ttf') format('truetype');
        font-weight:400;
        font-style:normal
    }

    @font-face{
        font-family:'Lato-Reg';
        src: url('/static/fonts/Lato-Reg.ttf') format('truetype');
        font-weight:400;
        font-style:normal
    }

    body{
        font-family: "Lato-Reg";
    }
</style>
Good job admin, you've just permanently banned Admin!
```

Sin dudas el endpoint /ban/user/<username> está siendo accesible sin ser admin. La validación de autenticación/roles en el backend no está configurada o tiene algún fallo.

## 8. Directory Traversal:



- Esta página es vulnerable a la navegación de directorios. Al usar la notación de puntos de Unix, puede acceder a archivos del sistema a los que no debería tener acceso. La codificación HTML es necesaria, ya que la mayoría de los navegadores resuelven los ".." en la URL.  
[https://owasp.org/www-community/attacks/Path\\_Traversal](https://owasp.org/www-community/attacks/Path_Traversal)
- La notación Unix: comienza con el directorio raíz (root directory), indicado por /  
[https://rsg-ecuador.github.io/unix.bioinfo.rsgecuador/content/Curso\\_basico/03\\_Manexo\\_terminal/4\\_directorios.html](https://rsg-ecuador.github.io/unix.bioinfo.rsgecuador/content/Curso_basico/03_Manexo_terminal/4_directorios.html)

```
[marilyn@ASUS-TUF-FX505] ~
$ ls
bin  dev  home  lib  lost+found  mnt  proc  run  srv  tmp  var
boot  etc  init  lib64  media  opt  root  sbin  sys  usr
$
```

- Debemos codificar / como %2F.
- Probe: ../../ en gran cantidad hasta que obtuve "algo" que no fuese acceso denegado: %2F.%2F.%2F

- Inspeccione el HTML desde el navegador y los profesores dejaron una pista:

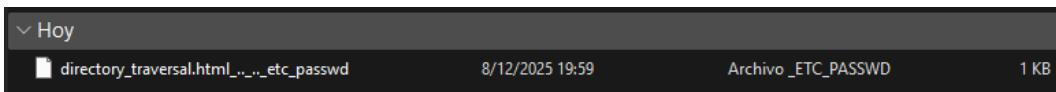
```
<html data-darkreader-mode="dynamic" data-darkreader-scheme="dark" data-darkreader-proxy-injected="true">
... ><head> ... </head> == $0
<body>
    "This page is vulnerable to directory traversal! Using unix dot notation, you can access system files that you shouldn't be able to. Html encoding is necessary here, since most browsers will resolve .'s in the url. "
    <!-- If you're stuck, try appending something similar to %2F..%2F..%2F..%2F..%2Fetc%2Fpasswd to the url -->
<div>
    <div class="vtResetstyles">...</div>
    <div id="vt-augment-container" class="vtzindex vt-augment drawer" style="background: rgb(49, 61, 90); --darkreader-inline-bgimage: initial; --darkreader-inline-bgcolor: var(--darkreader-background-31d5a, #273148); data-darkreader-inline-bgimage data-darkreader-inline-bgcolor"></div>
</div>
<span class="vttooltiptext vthidden">
    "Click to open IoC report"
    <br>
    "with VirusTotal Augment"
</span>
<style></style>
<style class="darkreader darkreader--sync" media="screen"></style>
</body>
</html>
```

<!-- If you're stuck, try appending something similar to %2F..%2F..%2F..%2F..%2Fetc%2Fpasswd to the url  
-->

- Comience a provar y con la siguiente URL se descargó el siguiente archivo (Lo adjunto en "MaterialAdjunto.7z"):

[https://marilynolivera.alexander.net.ar/views/directory\\_traversal.html%2F..%2F..%2Fetc%2Fpasswd](https://marilynolivera.alexander.net.ar/views/directory_traversal.html%2F..%2F..%2Fetc%2Fpasswd)

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
node:x:1000:1000::/home/node:/bin/bash
```



- /etc/passwd es un archivo de configuración del sistema en Unix/Linux que contiene información esencial sobre las cuentas de usuario. Cada línea representa un usuario y tiene 7 campos separados por : → *Nombre : Contraseña : ID de usuario : Grupo principal : Gecos : Directorio de inicio : Shell*  
<https://www.ibm.com/docs/es/aix/7.2.0?topic=files-etcpassword-file>

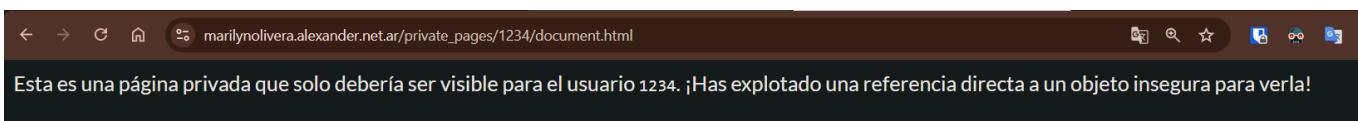
## 9. Insecure Direct Object Reference (IDOR) / Insecure Direct Object Reference

This page exposes an insecure direct object reference.

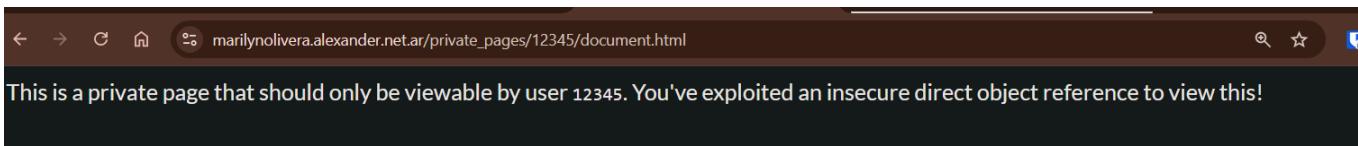
Many cases of this involve a simple user ID in the url, which can be incremented or decremented to view another user's private information.

A properly configured website must use the server to validate that information is only returned for an ID that matches the appropriate user.

- Esta página expone una referencia directa a un objeto insegura.  
En muchos casos, esto implica un ID de usuario simple en la URL, que puede incrementarse o decrementarse para ver la información privada de otro usuario.  
Un sitio web correctamente configurado debe usar el servidor para validar que la información solo se devuelva para un ID que coincida con el usuario correcto.  
[https://owasp.org/www-project-web-security-testing-guide/latest/4-Web\\_Application\\_Security\\_Testing/05-Authorization\\_Testing/04-Testing\\_for\\_Insecure\\_Direct\\_Object\\_References](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/04-Testing_for_Insecure_Direct_Object_References)
- En la URL se usa un número (123 en la sesión inicial) para determinar la información que se mostrará asociada con el usuario actual, está es visible y puede ser modificada.
- El servidor no realiza el control necesario para evitar que al modificar 123 por 1234 se puede acceder a información de un usuario distinto al mío:



[https://marilynolivera.alexander.net.ar/private\\_pages/1234/document.html](https://marilynolivera.alexander.net.ar/private_pages/1234/document.html)



Esto se produce debido a que se está dando acceso directo a un objeto, basándose en información del usuario actual usada en la URL.

## 10. Injections and remote code execution (Inyecciones y ejecución remota de código)

This tutorial will not directly include RCE and injection examples.

Why?

Due to the possibility of cross-site request forgery, it would actually open up your computer to exploitation.

For example, consider the scenario in which this server was running on localhost:8000, and the path /rce/command would attempt to execute `command` on your computer. The application is not actually secure just because it is running on localhost. If a malicious website were to perform a CSRF POST request to localhost:8000, it could choose the value of command and effectively compromise your computer.

The only safe way to allow this would be to jail the server in a virtual environment, but we will not be doing so in this server since it is intended to be barebones.

To read more on localhost CSRF examples, see here: [https://en.wikipedia.org/wiki/Cross-site\\_request\\_forgery#Example\\_and\\_characteristics](https://en.wikipedia.org/wiki/Cross-site_request_forgery#Example_and_characteristics)

To test RCE and Injection in a sandboxed environment, try <https://hack.me/101347/simple-remote-code-execution.html> and <https://hack.me/102131/very-basic-sql-injection.html>

Note: The directory traversal example in this server is *not* vulnerable due to the Same Origin Policy. While attackers can execute a request on your behalf with CSRF, they cannot read the results or the contents of the pages.

- Este tutorial no incluirá directamente ejemplos de RCE e inyección.

¿Por qué?

Debido a la posibilidad de falsificación de solicitudes entre sitios, esto expondría su equipo a ataques.

Por ejemplo, considere el escenario en el que este servidor se ejecuta en localhost:8000 y la ruta /rce/command intenta ejecutar `command` en su equipo. La aplicación no es realmente segura solo por ejecutarse en localhost. Si un sitio web malicioso realizará una solicitud CSRF POST a localhost:8000, podría elegir el valor `command` y comprometer su equipo.

La única forma segura de permitir esto sería enjaular el servidor en un entorno virtual, pero no lo haremos en este servidor, ya que está diseñado para ser básico.

Para obtener más información sobre ejemplos de CSRF en el host local, consulte aquí:

[https://en.wikipedia.org/wiki/Cross-site\\_request\\_forgery#Example\\_and\\_characteristics](https://en.wikipedia.org/wiki/Cross-site_request_forgery#Example_and_characteristics)

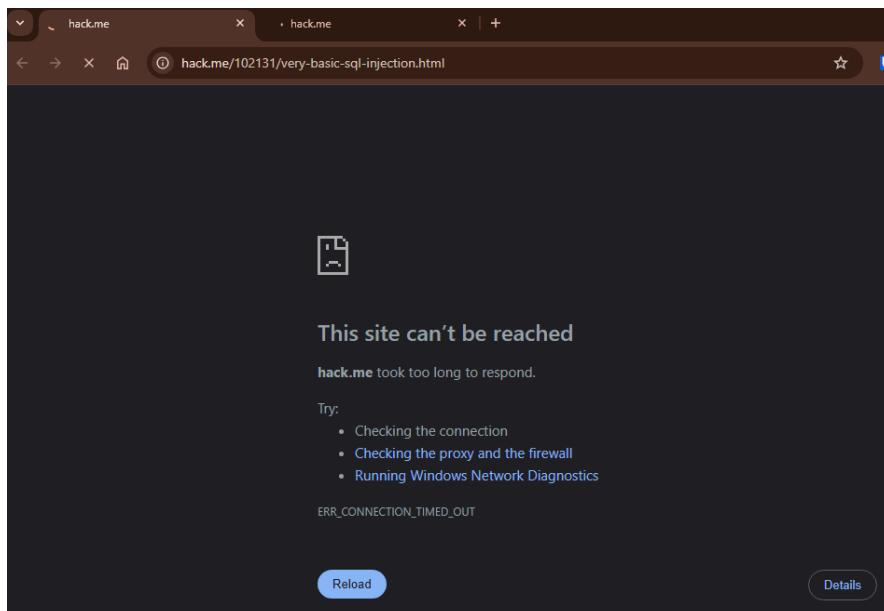
. Para probar RCE e inyección en un entorno aislado, consulte

<https://hack.me/101347/simple-remote-code-execution.html> y

<https://hack.me/102131/very-basic-sql-injection.html>.

Nota: El ejemplo de recorrido de directorio en este servidor *no* es vulnerable gracias a la Política del Mismo Origen. Si bien los atacantes pueden ejecutar una solicitud en su nombre con CSRF, no pueden leer los resultados ni el contenido de las páginas.

Por el momento no están disponibles los sitios para probar RCE e inyección de código:



[https://owasp.org/www-community/attacks/Code\\_Injection](https://owasp.org/www-community/attacks/Code_Injection)

→ Entonces Injections and RCE (remote code execution) es un tipo de ataque en el que se usa a un usuario legítimo, inocentes e inconscientes del efecto de las acciones que está llevando a cabo, para inyectar instrucciones maliciosas que serán ejecutadas porque el servidor confía en el usuario.

En resumidas palabras un atacante usa un usuario/víctima como vector para inyectar y ejecutar código en el servidor.

## 11. Mixed topics: Temas mixtos

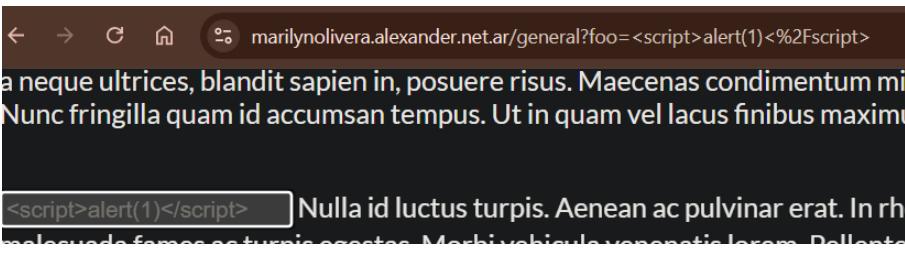
- Esta cuenta con un mix de los temas ya vistos.
- Está llena de texto de relleno.
- La URL recibe un parámetro <https://marilynolivera.alexander.net.ar/general?foo=a>
- Cuenta con un cuadro de texto que viene que viene por defecto

Mauris suscipit tellus maximus eros congue, quis consectetur lorem malesuada. Nullam auctor ullamcorper purus. Etiam volutpat est a justo dictum elementum. Ut pulvinar elit convallis, porta quam nec, tempus orci. Aliquam id luctus purus. Praesent consectetur nulla ut lectus consectetur imperdiet. Donec urna libero, tristique quis arcu non, imperdiet ullamcorper dolor. Quisque tempor dui non risus pellentesque, id bibendum erat accumsan. Sed eleifend pretium nulla. Integer iaculis et metus in porttitor. Aenean a neque ultrices, blandit sapien in, posuere risus. Maecenas condimentum mi quis fringilla suscipit. Aliquam dapibus finibus turpis a ullamcorper. a Nunc fringilla quam id accumsan tempus. Ut in quam vel lacus finibus maximus.

a Nulla id luctus turpis. Aenean ac pulvinar erat. In rhoncus sit amet leo et consectetur. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Morbi vehicula venenatis lorem. Pellentesque justo elit, suscipit vel luctus viverra, suscipit quis sapien. Nam lacinia tempor ipsum nec rutrum. Proin et maximus felis, ut consequat libero. Proin eget ornare nunc. Integer at iaculis augue. Phasellus et mauris odio. Morbi a tellus eget nisi dictum semper. Nullam dictum augue non sapien sagittis, eget ullamcorper massa eleifend. Proin quis euismod orci, ut laoreet purus. Nulla iaculis venenatis euismod.

### Reflected XSS

- Prueba con: <script>alert(1)</script>: %3Cscript%3Ealert(1)%3C%2Fscript%3E



La información se refleja dentro del cuadro.

- Análisis del HTML: podemos cerrar la definición del placeholder con " y agregar el script entonces".

```
...    <input placeholder="<script>alert(1)</script>"> == $0
    " Nulla id luctus turpis. Aenean ac pulvinar erat. In rh
    Pellentesque habitant morbi tristique senectus et netus
    egestas. Morbi vehicula venenatis lorem. Pellentesque ju
    viverra, suscipit quis sapien. Nam lacinia tempor ipsum
    ut consequat libero. Proin eget ornare nunc. Integer at
    odio. Morbi a tellus eget nisi dictum semper. Nullam dic
    ullamcorper massa eleifend. Proin quis euismod orci, ut
    venenatis euismod. "
```

- Analizando el código encontre al menos 4 usos del parámetro:

```
<div class="{{payload}}>
  Sed ut placerat ligula, et feugiat dui. Aenean sapien est, varius et pharetra et, mollis a turpis. Quisque
</div>
<br><br>
Nulla dictum nisl sit amet purus commodo, non scelerisque lectus mollis. Phasellus augue felis, pretium at
<br><br><br>
Sed vitae dui vulputate, porttitor libero non, lacinia dolor. Morbi semper massa eu vulputate posuere. Pell
<br><br>
Mauris suscipit tellus maximus eros congue, quis consectetur lorem malesuada. Nullam auctor ullamcorper pur
<br><br>
<p>
<input placeholder="{{payload}}>
  Nulla id luctus turpis. Aenean ac pulvinar erat. In rhoncus sit amet leo et consectetur. Pellentesque habit
</p>

<b>
  Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Cras eu dui a arcu
</b>
<br><br>
  Nullam mattis arcu sapien, sed mollis urna volutpat eget. Fusce mauris ante, scelerisque quis placerat vive
  Donec leo leo, consequat id purus in, placerat tempor massa. Duis elementum quam ut convallis ullamcorper. I
<div name='{{payload}}>Ut porta erat non condimentum sodales. Vivamus et ultricies sem. Aenean sit amet
<br><br>
  Duis ut vehicula nunc. Nullam sagittis interdum lectus eu tempus. Duis lacinia facilisis leo. Cras nec orci
<br><br>
<div id="{{payload}}>In massa urna, malesuada eget tristique sed, fringilla nec turpis. Nullam semper orci
<br><br>
  Aliquam arcu massa, mollis in felis sit amet, ultricies tristique ipsum. Aliquam scelerisque ornare aliquet
<br><br>
  Nunc at sapien nulla. Donec eget suscipit est. Vestibulum mollis lacus libero, sit amet sollicitudin augue
<br><br>
  Phasellus malesuada finibus eros nec ultricies. Quisque ut lorem molestie, interdum ante vitae, varius quam
<br><br>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc imperdiet sapien lacus, a egestas est vulputate
<br>
  Nullam finibus, purus non accumsan sagittis, quam urna lacinia massa, et gravida turpis leo id ipsum. Maece
<br>
  Pellentesque cursus metus tortor, a pulvinar lectus semper eu. Vestibulum vehicula justo ac ipsum iaculis f
```

En 3 de estas se usan {} (doble llave) esto significa que el payload es codificado a html automáticamente, osea que la " no cierra la configuración del atributo.

En un caso se usan {{{} }} (triple llave) esto significa que el payload se mantendrá tal como es ingresado, lo que sí me permitirá cerrar la definición del atributo.

Entonces ¿Por qué no funcionó?  
Porque en donde se usan 3 corchetes se usan comillas simples ‘ ‘ y yo envié “ para cerrar la cadena.

- Reacomodo y mandó la nueva cadena:

holis'><script>alert(1)</script> → holis%27%3E%3Cscript%3Ealert(1)%3C%2Fscript%3E

[https://marilynlivera.alexander.net.ar/general?foo=holis%27%3E%3Cscript%3Ealert\(1\)%3C%2Fscript%3E](https://marilynlivera.alexander.net.ar/general?foo=holis%27%3E%3Cscript%3Ealert(1)%3C%2Fscript%3E)



### Fuzzing:

<https://github.com/ffuf/ffuf>

Resultados en comprimido de archivos adjuntos en carpeta “Mixed-topics-Fuzzing”

→ Implementación de la misma herramienta que use para el punto de Fuzzing, para probar si este host es vulnerable al ingreso de caracteres especiales.

- Como parte de la URL:

- Caracteres especiales:

ffuf -w "/usr/share/seclists/Fuzzing/special-chars + urlencoded.txt" -u

<https://marilynlivera.alexander.net.ar/generalFUZZ> > MixedtopicsURLCaracteresEspeciales+urlcoded.txt

```
(marilyn@ASUS-TUF-FX505:~)
$ ffuf -w "/usr/share/seclists/Fuzzing/special-chars + urlencoded.txt" -u https://marilynlivera.alexander.net.ar/generalFUZZ > MixedtopicsURLCaracteresEspeciales+urlcoded.txt
          ^__^
         /  \_ \
        o   o_-
        ||----w |
        ||     ||
v2.1.0-dev

:: Method      : GET
:: URL         : https://marilynlivera.alexander.net.ar/generalFUZZ
:: Wordlist    : FUZZ: /usr/share/seclists/Fuzzing/special-chars + urlencoded.txt
:: Follow redirects: false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
:: Progress: [60/60] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 1 ::


```

No se obtuvieron resultados.

- Como parámetros:

- Command-injection:

```
echo%20AGIYMZ$((69%2B52))$(echo%20AGIYMZ)AGIYMZ
%20echo%20TDJHRY$((30%2B41))$(echo%20TDJHRY)TDJHRY
;echo%20MPCSBG$((54%2B42))$(echo%20MPCSBG)MPCSBG
&echo%20NWMZCF$((57%2B72))$(echo%20NWMZCF)NWMZCF
|echo%20TJEGSE$((27%2B57))$(echo%20TJEGSE)TJEGSE
||echo%20ANSBHE$((26%2B89))$(echo%20ANSBHE)ANSBHE
&&echo%20PVJXOS$((12%2B1))$(echo%20PVJXOS)PVJXOS
```

ExamenPractico			
	Nombre	Fecha de modificación	Tipo
ExamenPractico	command-injection-commix.txt	19/9/2025 02:28	Text Document 919 KB
marilyn	country-codes.txt	19/9/2025 02:28	Text Document 2 KB
	country-codes-lower-case.txt	19/9/2025 02:28	Text Document 1 KB

ffuf -w "/usr/share/seclists/Fuzzing/command-injection-commix.txt" -u

<https://marilynlivera.alexander.net.ar/general?foo=FUZZ> >

MixedtopicsParametro-command-injection-commix.txt

- Filtre del resultado los caracteres que dieron resultado:

grep -i "status. (500|403|404|400|302)" MixedtopicsParametro-command-injection-commix.txt

```
(marilyn@ASUS-TUF-FX505:~)
$ grep -i "status: (500|403|404|400|302)" MixedtopicsParametro-command-injection-commix.txt

(marilyn@ASUS-TUF-FX505:~)
$
```

No dio resultados.

CSRF: Podemos ver que tiene la protección activada.

Name	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
general?foo=(^._.^)	▼ General						
foobar1.js							
image1.png							
vcf15cbe7772f49c399c6a5bab...							
csrf_protected_form							
Lato-Reg.ttf							
rum							
gti-vt-icon2.png	▼ Response Headers						

## Authentication Bypass:

Name	Status	Type	Initiator	Size	Time
general?foo=(^._.^)	200	document	Other	5.5 kB	202 ms
foobar1.js	200	script	general?foo=(^._.^):22	0.6 kB	228 ms
image1.png	200	png	general?foo=(^._.^):39	0.9 kB	224 ms
vcf15cbe7772f49c399c6a5babf22c1241717689176015	200	script	general?foo=(^._.^):83	7.0 kB	29 ms
csrf_protected_form	200	xhr	general?foo=(^._.^):28	0.5 kB	205 ms
Lato-Reg.ttf	200	font	general?foo=(^._.^):84	52.6 kB	508 ms
rum	204	xhr	vcf15cbe...:1	0.5 kB	102 ms
gti-vt-icon2.png	200	png	virustotal.js:229	17.9 kB	6 ms
js.js	200	script	content.js:43	1.4 kB	2 ms
dom.js	200	script	content.js:43	2.0 kB	4 ms
js.js	200	script	content.js:43	1.4 kB	2 ms
rum	204	ping	vcf15cbe...:1	0.5 kB	90 ms

Estas son las peticiones de red.

- [foobar1.js](#) es usada desde el html, no hay situación similar a la anterior.

```
<script src="/static/foobar1.js"></script>
```

- [js.js](#) [dom.js](#) y [js.js](#) son funciones que no pertenecen a la aplicación.
- No se encontraron funciones sin uso que podrían permitir Authentication Bypass.

## Directory Traversal:

- Prueba de: %2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd  
No dio resultados.

Con el nivel de exploración realizado, no encontré más vulnerabilidades evidentes.

## Intento 1 de resolución del examen con Oracle y Coolify:

Cuento con acceso a un servidor en la nube de Oracle, con docker previamente instalado. El servidor es de mi pareja, que lo usamos para proyectos personales, él también tiene un dominio por eso este lleva su nombre. En el voy a instalar Coolify, un VPS (Servidor Privado Virtual) que te da la interfaz y la simplicidad de un Heroku, pero utilizando tu propio servidor en lugar de la infraestructura de ellos. Este además se puede conectar con github actions por medio de un Webhook y un Token de la API que Coolify proporciona.

Para hacer el pipeline de CI/CD usaré GitHub Actions. Este tiene su propio registro de contenedores ghcr.io GitHub Container Registry, se puede comunicar con Coolify y permite automatizar escaneos e integrar varias de las herramientas que vimos (Snyk, OWASP ZAP, Dependabot, SonarQube y más).

- Instalación de Coolify instancia de servidor de Oracle: <https://coolify.io/docs/get-started/installation>  
<https://coolify.io/docs/knowledge-base/server/firewall>

```
curl -fsSL https://cdn.coollabs.io/coolify/install.sh | sudo bash
```

```
ubuntu@UbuntuOracleCloudVPS01: $ curl -fsSL https://cdn.coollabs.io/coolify/install.sh | sudo bash
Welcome to Coolify Installer!
This script will Install everything for you. Sit back and relax.
Source code: https://github.com/coollabsio/coolify/blob/v4/x/scripts/install.sh
Using default Registry URL: ghcr.io
-----
Operating System | ubuntu 24.04
Docker           | 27.0
Coolify          | 4.0.0-beta.452
Helper           | 1.0.12
Realtime         | 1.0.10
Docker Pool      | 10.0.0.0/8 (size 24)
Registry URL     | ghcr.io
-----
1. Installing required packages (curl, wget, git, jq, openssl).
2. Check OpenSSH server configuration.
- OpenSSH server is installed.
- SSH PermitRootLogin is enabled.
3. Check Docker Installation.
- Docker is installed.
4. Check Docker Configuration.
- Network pool configuration: 10.0.0.0/8/24
- To override existing configuration: DOCKER_POOL_FORCE_OVERRIDE=true
-----
```

Creating new Docker configuration with network pool: 10.0.0.0/8/24  
- Configuration updated - restarting Docker daemon...  
- Docker daemon restarted successfully  
5. Download required files from CDN.  
6. Setting up environment variable file  
- No .env file found, copying .env.production to .env  
7. Checking and updating environment variables if necessary...  
- Updated value of APP\_ID as the current value was empty  
- Updated value of APP\_KEY as the current value was empty  
- Updated value of DB\_PASSWORD as the current value was empty  
- Updated value of REDIS\_PASSWORD as the current value was empty  
- Updated value of PUSHER\_APP\_ID as the current value was empty  
- Updated value of PUSHER\_APP\_KEY as the current value was empty  
- Updated value of PUSHER\_APP\_SECRET as the current value was empty  
- Added DOCKER\_ADDRESS\_POOL\_BASE and it's value as the variable was missing  
- Added DOCKER\_ADDRESS\_POOL\_SIZE and it's value as the variable was missing  
8. Checking for SSH key for localhost access.  
- Generating SSH key.  
9. Installing Coolify (4.0.0-beta.452)  
- It could take a while based on your server's performance, network speed, stars, etc.  
- Please wait.  
- Until then, here's a joke for you:  
Debugging is like being the detective in a crime movie where you're also the murderer at the same time.

```
-----
```

Your instance is ready to use!

You can access Coolify through your Public IPV4: <http://129.153.172.47:8000>

If your Public IP is not accessible, you can use the following Private IPs:

<http://172.18.0.1:8000>  
<http://100.114.185.60:8000>  
<http://10.0.1.1:8000>  
<http://10.0.2.1:8000>  
<http://fd7a:115:scale0::c901:b948:8000>  
<http://fdaa:cida:d3dc::1:8000>

WARNING: It is highly recommended to backup your Environment variables file (/data/coolify/source/.env) to a safe location, outside of this server (e.g. into a Password Manager).

```
ubuntu@UbuntuOracleCloudVPS01: $ 
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
25dae569d69	ghcr.io/coollabsio/sentinel:0.0.18	"/app/sentinel"	24 minutes ago	Up 24 minutes (healthy)	8000/tcp, 8443/tcp, 9000/tcp, 0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp	coolify-sentinel
ce342c9cc5	ghcr.io/coollabsio/coolify:4.0.0-beta.452	"docker-php-serveri..."	25 minutes ago	Up 24 minutes (healthy)	8000/tcp, 8443/tcp, 9000/tcp, 0.0.0.0:6001-6002->6001-6002/tcp, [::]:6001-6002->6001-6002/tcp	coolify
6104270b49d	ghcr.io/coollabsio/coolify-realtime:1.0.10	"/bin/sh /socketi-ent..."	25 minutes ago	Up 25 minutes (healthy)	0.0.0.0:6001-6002->6001-6002/tcp, [::]:6001-6002->6001-6002/tcp	coolify-realtime
67e116c178	postgres:15-alpine	"docker-entrypoint.s..."	25 minutes ago	Up 25 minutes (healthy)	5432/tcp	coolify-db
30c885f4019f	redis:7-alpine	"docker-entrypoint.s..."	25 minutes ago	Up 25 minutes (healthy)	6379/tcp	coolify-redis

Configure nombre, email y contraseña:

The screenshot shows the Coolify setup interface across three main sections:

- Welcome to Coolify:** A dark-themed page with a "Welcome to Coolify" header and a "Let's go!" button. It includes a "Skip Setup" link and a progress bar indicating "Configuración completa! (Completa)".
- Elige el tipo de servidor:** A step titled "Elige el tipo de servidor". It asks to select where to implement applications and databases. It shows three options:
  - Esta máquina**: "Implementar en el servidor que ejecuta Coolify. Ideal para pruebas y configuraciones de un solo servidor."
  - Servidor remoto**: "Conéctese a través de SSH a cualquier servidor: VPS en la nube, hardware o infraestructura doméstica."
  - Nube de Hetzner**: "Implemente servidores directamente desde su cuenta de Hetzner Cloud."
- Configuración del proyecto:** A step titled "Configuración del proyecto". It asks to create the first project for organizing applications, databases, and services. It includes a "Crear 'Mi primer proyecto'" button and a "DETALLES TÉCNICOS" section with sub-sections for "Organización del proyecto", "Entorno", and "Acceso del equipo".

- Integro Coolify a GitHub Actions: <https://coolify.io/docs/applications/ci-cd/github/github-actions>

- Configuración en Coolify

1. Creé en Coolify un nuevo recurso que espera recibir una imagen de docker lista, desde la notificación de GitHub, para desplegarla en el servidor de oracle.

The screenshot shows the Coolify interface for creating a new resource. On the left sidebar, 'Projects' is selected. In the main area, under 'Docker Based', the 'Docker Image' option is highlighted with a green border. The 'Image Name' field contains 'ghcr.io/marilynolivera/vulnerable-web-app'. Below it, the 'Tag (optional)' field has 'latest' selected. To the right, there is a 'SHA256 Digest (optional)' field with the value '59e02939b1bf39f16c93138a28727'.

→ Se define el nombre de la imagen, el nombre de la imagen debe seguir la estructura estándar de los registros de contenedores: registry.hostname / owner / repository-name\

Algo como: ghcr.io/ginodevops/vulnerable-web-app

El nombre debe ir todo en minuscula, le puse mi nombre ya que hice un fork:  
ghcr.io/marilynolivera/vulnerable-web-app

The screenshot shows the Coolify interface for creating a new application. On the left sidebar, 'Projects' is selected. In the main area, the 'Create a new Application' screen is shown with 'Docker Image' selected. The 'Image Name' field contains 'ghcr.io/marilynolivera/vulnerable-web-app'. Below it, the 'Tag (optional)' field has 'latest' selected. To the right, there is a 'SHA256 Digest (optional)' field with the value '59e02939b1bf39f16c93138a28727'. Below this, the 'Configuration' tab is selected, showing deployment details like port mappings (80 to 3000-3000) and network aliases. The 'Webhooks' section is also visible.

2. Webhook: Esta URL es la dirección a la que GitHub Actions enviará la señal POST para decirle a Coolify “¡Nueva imagen lista, despliega!”.

The screenshot shows the Coolify interface for configuration. On the left sidebar, 'Projects' is selected. In the main area, the 'Configuration' tab is selected. Under 'Webhooks', there is a section for 'Manual Git Webhooks' with entries for GitHub, GitLab, Bitbucket, and Gitea, each with their respective webhook URLs and secret keys.

<https://coolify.alexander.net.ar/api/v1/deploy?uuid=f8ccc4skwc4cgcs8408k8ow4&force=false>

- Así quedo configurado el proyecto en coolify:

3. Cree un token API de Coolify: <https://coolify.io/docs/api-reference/authorization>

Con estos datos ya se puede hacer la implementación en github action, es necesario enviar una solicitud GET a ese punto final del webhook (autenticado con el token) para activar la implementación con este formato:

```
curl --request GET "${% raw %}{{ secrets.COOLIFY_WEBHOOK }}% endraw %" --header "Authorization: Bearer ${% raw %}{{ secrets.COOLIFY_TOKEN }}% endraw %"
```

4. Agregar los datos en variables secretas del repositorio: COOLIFY\_TOKEN COOLIFY\_WEBHOOK

- Construcción del pipeline:

5. Se debe definir un token más, un Personal Access Token (PAT) de GitHub que se usa para autenticarse en GitHub Container Registry(GHCR), necesario para subir imágenes Docker a mi registro privado/público: un GH\_TOKEN

6. Prueba de funcionamiento, me baso en un template de ejemplo que provee Coolify y lo adapto:  
<https://github.com/andrasbacsai/github-actions-with-coolify/blob/main/.github/workflows/build.yaml>

Asigne el nombre a la imagen que configure en coolify:

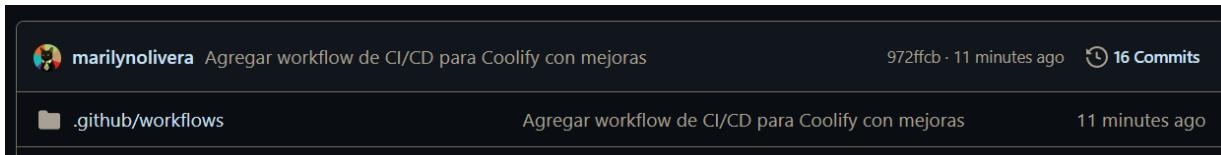
```
.github > workflows > deploy.yml
1 # For more details, read this: https://coolify.io/docs/github-actions
2 name: Despliegue app - Examen Practico Hackademy
3 on:
4   push:
5     branches: ["main"]
6   workflow_dispatch: #permite ejecucion manual
7
8 env:
9   REGISTRY: gcr.io
10  IMAGE_NAME: "marilynlivera/vulnerable-web-app"
11
12 jobs:
13   amd64:
14     runs-on: ubuntu-latest
15     permissions:
16       contents: read
17       packages: write
18     steps:
19       - uses: actions/checkout@v3
20       - name: Login to gcr.io
21         uses: docker/login-action@v2
22         with:
23           registry: ${env.REGISTRY}
24           username: ${github.actor}
25           password: ${secrets.GH_TOKEN}
26       - name: Build image and push to registry
27         uses: docker/build-push-action@v4
28         with:
29           context: .
30           file: Dockerfile
31           platforms: linux/amd64
32           push: true
33           tags: ${env.REGISTRY}/${env.IMAGE_NAME}:latest
34       - name: Deploy to Coolify
35         run:
36           curl --request GET '${secrets.COOLIFY_WEBHOOK}' --header 'Authorization: Bearer ${secrets.COOLIFY_TOKEN}'
```

Este pipeline automatiza el despliegue:

- Se sube código a GitHub
- Se construye una imagen de Docker
- Se sube al registro de GitHub (GHCR)
- Automáticamente se notifica a Coolify
- Automáticamente Coolify despliega la nueva versión

Lo subí, en ese momento no tenía habilitado la ejecución de workflows, lo habilité, decidí cambiarle el nombre que venía en el ejemplo de coolify (Build Static Image) a uno relacionado con el examen y habilitar la opción de ejecución manual:

```
● PS D:\Repos\Hackademy\ExamenPractico\vulnerable-web-app> git add .github/workflows/deploy.yml
● PS D:\Repos\Hackademy\ExamenPractico\vulnerable-web-app> git commit -m "Agregar workflow de CI/CD para Coolify con mejoras"
[master 972ffcb] Agregar workflow de CI/CD para Coolify con mejoras
 1 file changed, 1 insertion(+), 1 deletion(-)
● PS D:\Repos\Hackademy\ExamenPractico\vulnerable-web-app> git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 462 bytes | 462.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/marilynlivera/vulnerable-web-app.git
 3c00a66..972ffcb  master -> master
```



Lo corrí y luego de corregir error de sintaxis salió okey:

The screenshot shows the Coolify interface for the workflow. On the left, there is a summary table with one job named 'amd64'. The job status is 'Success' with a duration of '1m 6s'. On the right, there is a detailed log of the workflow steps. The log shows the following steps:

- Manually triggered 1 minute ago
- Status: Success
- Total duration: 1m 6s
- Artifacts: -
- Re-run all jobs

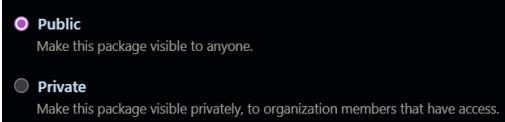
The log details the execution of the 'deploy.yml' file, specifically the 'amd64' job. It lists the following steps:

- Set up job
- Run actions/checkout@v3
- Login to ghcr.io
- Build image and push to registry
- Deploy to Coolify
- Post Build image and push to registry
- Post Login to ghcr.io
- Post Run actions/checkout@v3
- Complete job

Pero no levantó la imagen en mi host, comencé a ver por qué y a corregir algunos errores:

- En el template de ejemplo de coolify cuando se establece que cierra cada vez que se haga push se usa la rama main pero acá usamos la rama master:

```
push:
  branches: ["master"]
```
- Después de analizar me encontré con que la imagen en GHCR se estaba guardando como privada, lo que pensé erróneamente que no era así (pensé que si el repositorio era público eran públicas, pero ahora son todas privadas por default), decidí hacerla pública para el examen:



### Prueba desde mi consola:

```
PS C:\Users\maril> docker pull ghcr.io/marilynolivera/vulnerable-web-app:latest
Error response from daemon: error from registry: unauthorized
unauthorized
PS C:\Users\maril> docker pull ghcr.io/marilynolivera/vulnerable-web-app:latest
latest: Pulling from marilynolivera/vulnerable-web-app
9935d9c62ace: Pull complete
a1cdcc6c1b07: Pull complete
db8fb9db2fe: Pull complete
91a71170aa27: Pull complete
lc151cd1b3ea: Pull complete
645c20ec8214: Pull complete
e705a4cf4fd31: Pull complete
c877b722db6f: Pull complete
fb9d93995f40: Pull complete
146bd6a88618: Pull complete
db0efb88e806: Pull complete
4f4fb706ef54: Pull complete
Digest: sha256:d813038c6a8b6a57448af732ec0b563370a73b579a3e54c3a4b8b7af71b72edd
Status: Downloaded newer image for ghcr.io/marilynolivera/vulnerable-web-app:latest
ghcr.io/marilynolivera/vulnerable-web-app:latest
```

amd64
succeeded now in 32s
> <input checked="" type="checkbox"/> Set up job
> <input checked="" type="checkbox"/> Run actions/checkout@v3
> <input checked="" type="checkbox"/> Login to ghcr.io
> <input checked="" type="checkbox"/> Build image and push to registry
> <input checked="" type="checkbox"/> Deploy to Coolify
> <input checked="" type="checkbox"/> Post Build image and push to registry
> <input checked="" type="checkbox"/> Post Login to ghcr.io
> <input checked="" type="checkbox"/> Post Run actions/checkout@v3
> <input checked="" type="checkbox"/> Complete job

- La imagen fue descargada y el contenedor ejecutado pero constantemente se restauraba. En los logs de Coolify se reportaba que un archivo tenía un formato incorrecto:

CENTER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS	PORTS
05d4ba593cfa	ghcr.io/marilynolivera/vulnerable-web-app:latest	"docker-entrypoint.s_"	4 minutes ago	Restarting (255)	Less than a second ago	
cca47e49813e	traefik:v3.6	"entrypoint.sh --pl_"	55 minutes ago	Up 55 minutes	(healthy)	0.0.0.0:8080->8080/tcp, 0.0.0.0:443->443/udp, [::]:443->443/udp
25dae36e9d09	ghcr.io/coollabsio/sentinel:0.0.18	"/app/sentinel"	29 hours ago	Up 23 hours	(healthy)	
ce342c29cfc5	ghcr.io/coollabsio/coolify:4.0.0-beta.452	"docker-php-serversi_"	29 hours ago	Up 29 hours	(healthy)	8000/tcp
6104270b4a9d	ghcr.io/coollabsio/coolify-realtime:1.0.10	"bin/sh /socketi-ent_"	29 hours ago	Up 29 hours	(healthy)	0.0.0.0:8080->8080/tcp
67e110c9d178	postgres:15-alpine	"docker-entrypoint.s_"	29 hours ago	Up 29 hours	(healthy)	5432/tcp
38c885f4019f	redis:7-alpine	"docker-entrypoint.s_"	29 hours ago	Up 29 hours	(healthy)	6379/tcp
c905ela0ff34	cloudflare/cloudflared:latest	"cloudflared --no-aut_"	7 weeks ago	Up 29 hours		
		cloudflared				

exec /usr/local/bin/docker-entrypoint.sh: exec format error

```
ubuntu@UbuntuOracleCloudVPS01:~$ uname -m
aarch64
ubuntu@UbuntuOracleCloudVPS01:~$ docker inspect ghcr.io/marilynolivera/vulnerable-web-app:latest | grep Architecture
"Architecture": "amd64",
```

El escenario actual es que la instancia gratuita de Oracle Cloud está usando una arquitectura de CPU ARM64 (aarch64), mientras que el Dockerfile de la aplicación vulnerable está diseñado para procesadores AMD64. Por esto voy a optar por usar una instancia creada para la arquitectura AMD64 en GCP.

## Implementación de workflow ZAP proxy

→ Como zap era ideal para la situación de la vulnerabilidad numero 6 pero interpretado erróneamente el enunciado, también me incentivó la detección y análisis de todas las vulnerabilidades de la lista, así que lo implemente y lo deje: Voy a usar una plantilla que arme en la práctica de DAST (analizaba en modo baseline <http://testphp.vulnweb.com>). Adapto: target: <https://marilynlivera.alexander.net.ar/> y el tipo de análisis (uses) será el que incluye las pruebas de fuzzing: zaproxy/action-full-scan@v0.13.0

The screenshot shows the GitHub Actions interface for a workflow named 'ZAP Security Scan'. The workflow file is located at '.github/workflows/zap-proxy-analysis.yml' and contains the following configuration:

```

name: ZAP Security Scan
on:
  push:
    branches: [ main ]
  workflow_dispatch: #permite ejecucion manual
jobs:
  zap_scan:
    runs-on: ubuntu-latest
    name: ZAP Security Scan
    permissions:
      contents: write
      issues: write
      actions: read
      security-events: write
    steps:
      - name: Checkout
        uses: actions/checkout@v4
      - name: ZAP Scan
        uses: zaproxy/action-full-scan@v0.13.0
        with:
          token: ${{ secrets.GITHUB_TOKEN }}
          target: 'https://marilynlivera.alexander.net.ar'
          rules_file_name: '.zap/rules.tsv'
          cmd_options: '-a'
      - name: Upload ZAP Report
        uses: actions/upload-artifact@v4
        with:
          name: zap-security-report
          path: "*_*.html"

```

The workflow has been triggered via push now by 'marilynlivera' and completed successfully in 40s. The build step was successful. The workflow summary shows the status as 'Success'.

→ Ejecutar el workflow de el ZAP:

The screenshot shows the GitHub Actions interface for the 'ZAP Security Scan' workflow. It displays a single workflow run with the following details:

- Run workflow** button
- ZAP Security Scan** (Workflow name)
- zap-proxy-analysis.yml** (Workflow file)
- 1 workflow run**
- This workflow has a workflow\_dispatch event trigger.**
- ZAP Security Scan** (Job name)
- ZAP Security Scan #1: Manually run by marilynlivera** (Run details)
- now** (Event)
- In progress** (Status)
- master** (Branch)
- ...** (More options)

- Espere que termine :). En la primera ejecución tube un error por no tener activados los issues para ese repositorio:

The screenshot shows the GitHub repository interface for the 'ZAP Security Scan' workflow. It displays a workflow run with the following details:

- Manually triggered 21 minutes ago**
- marilynlivera > 2fa0113 master**
- Status: Failure**
- Total duration: 3m 50s**
- Artifacts: -**
- zap-proxy-analysis.yml** (Workflow file)
- Annotations: 1 error**
- ZAP Security Scan** (Job name)
- Issues: Issues have been disabled in this repository. - <https://docs.github.com/en/issues/issues-with-create-an-issue>**

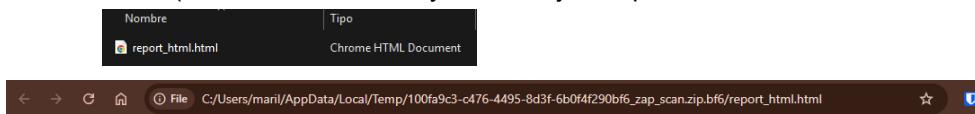
→ Lo volvi a ejecutar:

The screenshot shows the GitHub repository interface for the 'ZAP Security Scan' workflow. It displays a workflow run with the following details:

- Manually triggered 1 hour ago**
- marilynlivera > 2fa0113 master**
- Status: Success**
- Total duration: 3m 11s**
- Artifacts: 2**
- zap-proxy-analysis.yml** (Workflow file)
- ZAP Security Scan** (Job name)
- Artifacts produced during runtime:**
  - zap-security-report**: Size 31 KB, Digest sha256:a93c2ed5c7bd0d99ded05bf57802a64bb119a084...
  - zap\_scan**: Size 63 KB, Digest sha256:c183c8937cb3780ffef11d918f09a1478d573a...

## Generó dos reportes:

- zap-security-report en formato HTML o JSON contiene el resumen de todas las vulnerabilidades que ZAP encontró (como XSS, CSRF, Inyecciones y la exposición de información debido al Fuzzing).



### ZAP by Checkmarx ZAP Scanning Report

Sites: <http://marilynolivera.alexander.net.ar> <https://marilynolivera.alexander.net.ar>

Generated on Sun, 7 Dec 2025 18:58:39

ZAP Version: 2.16.1

ZAP by [Checkmarx](#)

#### Summary of Alerts

Risk Level	Number of Alerts
High	2
Medium	4
Low	9
Informational	14
False Positives	0

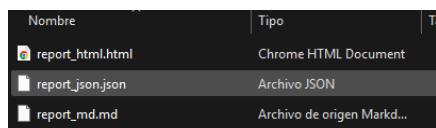
#### Summary of Sequences

For each step: result (Pass/Fail) - risk (or highest alert(s) for the step, if any).

#### Alerts

Name	Risk Level	Number of Instances
Cross Site Scripting (DOM Based)	High	4
Cross Site Scripting (Reflected)	High	5
CSP: Failure to Define Directive with No Fallback	Medium	1
Content Security Policy (CSP) Header Not Set	Medium	11
Missing Anti-clickjacking Header	Medium	11
Proxy Disclosure	Medium	25

- zap\_scan: es el archivo de sesión o el reporte más detallado de la ejecución, contiene todos los detalles de los requests y responses que ZAP usó para probar la aplicación. También incluye el mismo reporte-html.html.



(Adjunto archivos en la entrega “zap\_materialAdjunto.7z”)

Además el pipeline sube el reporte a los Issues de Github:

- Por desgracia no se destaca la vulnerabilidad Fuzzer en la pagina específica

<https://marilynolivera.alexander.net.ar/fuzzing/fuzz.html>, sin en la csrf puesto que expone datos confidenciales, ademas da informacion acerca de un para mas pero no de la que necesito.

Si leíste hasta acá, gracias, me entusiasma la ciberseguridad, hace poco entré a este mundo y estoy en proceso de crecimiento y aprendizaje constante.