2025

INFORME TÉCNICO-EJECUTIVO

DAVID NÚÑEZ FUENTES

ÍNDICE

I.
INFORME EJECUTIVO

01. Propósito y alcance

El propósito de esta pueba de penetración es encontrar el mayor número de vulnerabilidades reproducibles, valorar su criticidad y ver hasta que punto son capaces de comprometer a la víctima.

02. Hallazgos principales

Apartado donde se enumerán los hallazgos con un nivel de criticidad mayor hasta los hallazgos con un bajo nivel de criticidad.

03. Conclusiones y riesgo global

Conclusiones finales sobre la prueba de penetración,argumentando el nivel de compromiso en el que se encuentra el host objetivo y las vulnerabilidades más críticas explotadas.

04. Recomendaciones Prioritarias

Recomendaciones finales sobre el objetivo, dando pautas y consejos para erradicar dichas vulnerabildiadaddes y reforzar la seguridad del host.

01. Propósito y alcance

El presente informe ejecutivo documenta los hallazgos y las acciones realizadas durante una evaluación de seguridad ofensiva (Penetration Test) dirigida principalmente al servidor que aloja la aplicación web **"LosVengadores"**. (10.0.2.21)

Propósito

El objetivo principal de esta evaluación fue determinar el nivel de exposición y el riesgo operacional de la plataforma, identificando vulnerabilidades que pudieran ser explotadas para obtener acceso no autorizado a los sistemas subyacentes.

Específicamente, se buscaron las principales vulnerabilidades recogidas en la OSWAP 10 con el propósito final de realizar una escalada de privilegios para comprometer íntegramente el sistema.

Las principales ataques que se realizaron según las vulnerabilidades del sistema fueron:

CATEGORÍA	NOMBRE	HALLAZGO
A03:2021	**Inyección**	Base de datos **vulnerable a SQLi** en la que se encontraron credenciales expuestas
A04:2021	**Diseño Inseguro**	El código de programación del dominio LosVengadores **no esá correctamente sanitizado** ya que un usuario con bajo privilegio puede modificar dicho código sin ningún tipo de impedimento.
A05:2021	**Mala Configuración de Seguridad**	El subidor de archivos encontrado en el propio código del dominio mencionado p**ermite subir todo tipo de extensiones,** incluyendo la .php, obteniendo así un ataque de ejecución remota de código (RCE)
A06:2021	**Componentes Vulnerables y Obsoletos**	**Pwnkit (CVE-2021-4034)** en el binario pkexec y el uso del **SO Ubuntu 2.4.29** completamente obsoleto.
A07:2021	**Fallos de Identificación y Autenticación**	**Bypass de Autenticación** exitoso en el login de administrador de la página debido a la Inyección SQL mal manejada por la aplicación.

02. Hallazgo crítico princal

Sanitización nula de código:

La incapacidad del código implementado en el servidor web LosVengadores hace que, código oculto dentro del dominio pueda ser modificado y ejecutado fácilmente por cualquier usuario, consiguiendo descubrir y añadir una sección escondida a drede por el programador para subir archivos sin nigún tipo de filtro ni seguridad.

Falla en el Control de Acceso

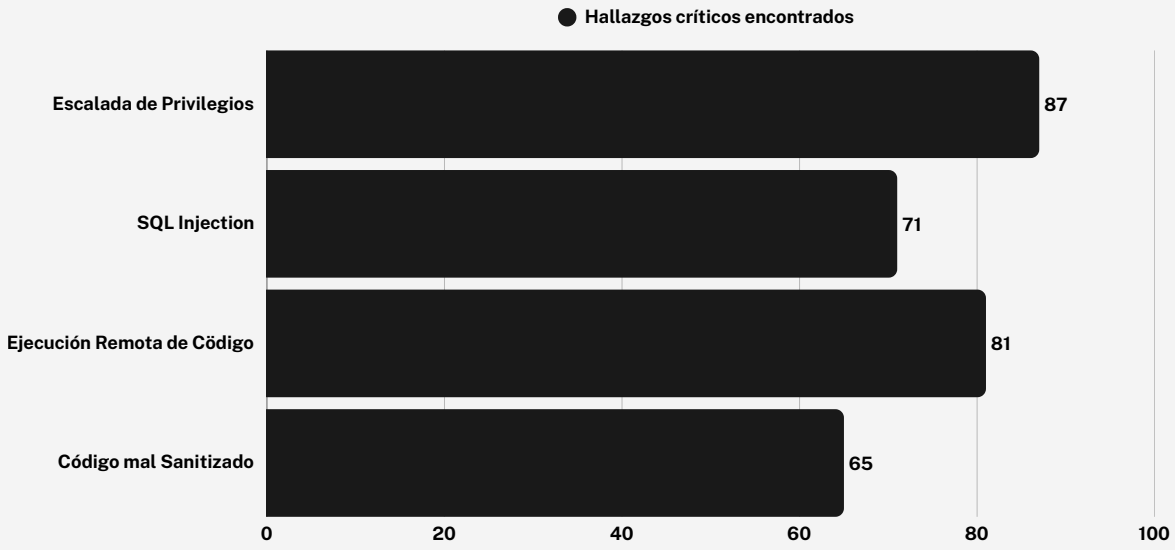
La aplicación web presentaba vulnerabilidades de Inyección SQL, lo que permitió a un atacante eludir el proceso de autenticación inicial y obtener acceso a credenciales sensibles de la base de datos.

Ejecución de Código Remoto

La debilidad en el manejo de archivos en la plataforma de desarrollo del dominio LosVengadores permitió la subida y ejecución de código arbitrario (Reverse Shell), comprometiendo dicho servidor web con privilegios de usuario bajo (www-data).

Escalada de Privilegios Crítica

Finalmente, se identificó un defecto de configuración y software obsoleto (binario pkexec vulnerable) que permitió la Escalada de Privilegios Local con la capacidad de obtener control administrativo absoluto del servidor (root).



“La criticidad de este riesgo requiere una intervención inmediata antes de cualquier otra actividad de desarrollo o mantenimiento.”

02. Hallazgo crítico princal

Escalada de Privilegios Local (ROOT)

- **Mecanismo:** La versión obsoleta del paquete *policykit-1* permite a un usuario de bajo privilegio (www-data) manipular los argumentos del programa y ejecutar código arbitrario con privilegios de root.

- **Impacto Confirmado:** Compromiso total del sistema, permitiendo al atacante el control administrativo absoluto.

Ejecución de Código Remoto (RCE)

- **Mecanismo**:
  a. La función *upload_cv()* **delega toda la responsabilidad de seguridad en el servidor**, ya que no realiza validación en el lado del cliente.
  b. La configuración HTML (<form action="DEPRECATED" method="get">) fue ignorada, y se utilizó la llamada directa a *upload_cv()* para **forzar el envío sin filtros**.
  c. La omisión de validación de tipo de archivo en el servidor permitió la subida de un **archivo .php malicioso**.

- **Impacto Confirmado:** Obtención de una *Reverse Shell* con el usuario *www-data*. Esto proporcionó el acceso inicial al sistema operativo, indispensable para iniciar la escalada a root.

Bypass de Autenticación (SQL Injection)

- **Mecanismo:** La aplicación no sanitiza correctamente los inputs del usuario, permitiendo la inyección de código SQL que alteró la lógica de la consulta de autenticación.

- **Impacto Confirmado:**
  - Acceso al Panel de Control de la aplicación web sin credenciales válidas.
  - Exposición de Credenciales de la base de datos (DB), incluyendo el hash de la contraseña del usuario administrador de la página capitanamerica (6f2f0046544e6821b04c99ec8cdb98f4).

03. Conclusiones y riesgo global

La evaluación de seguridad interna del servidor "LosVengadores" (10.0.2.21) concluye que la plataforma presenta un **Nivel de Riesgo Global CRÍTICO**. La existencia de una cadena de vulnerabilidades probada permite la capacidad de transición de un acceso web básico a la ejecución de código a nivel de sistema (Root).

Conclusiones Clave

- **Fallo de Arquitectura y Configuración:** El servidor está ejecutando componentes de software obsoletos y mal configurados, siendo el más crítico el binario **pkexec** (versión ***0.105***).

- **Ausencia de Defensa en Profundidad:** El compromiso inicial vía **Inyección SQL** y la siguiente obtención de un reverse shell demuestran una **falla significativa en la la seguridad de aquella base de datos**. La cuenta de usuario de bajo privilegio (*www-data*) conservaba los permisos necesarios para interactuar con los servicios clave del sistema operativo, violando el **Principio del Mínimo Privilegio.**

- **Exposición a Credenciales**: La debilidad en la aplicación y de la codificación de credenciales **facilitó el descubrimiento de usuarios y contraseñas sensibles,** lo que aumenta la superficie de ataque a otros servicios.

04. Recomendaciones Prioritarias

Las siguientes acciones deben implementarse de manera inmediata y con la máxima prioridad para mitigar el riesgo **CRÍTICO** de compromiso del sistema. La postergación de estas medidas dejará el servidor expuesto al control total de atacantes con bajo privilegio.

1. Parcheo y Actualización Crítica del Binario pkexec

- **Acción**: Aplicar el parche de seguridad para el gestor de políticas policykit-1 o, preferiblemente, actualizar el paquete completo a una versión que mitigue la vulnerabilidad **Pwnkit.**

- **Justificación**: Esta acción elimina el vector de ataque más peligroso que permite la escalada de privilegios a root.

2. Refuerzo y Reestructuración de la Aplicación Web (Subida de Archivos)

- **Acción**: Desactivar la función de subida de CV hasta que se implemente una validación estricta (Server-Side) en el script ***CV-upload.php***.

- **Justificación**: Es imperativo que el servidor **rechace activamente cualquier archivo** que no cumpla con las políticas de seguridad (ej., bloquear extensiones .php). Esto **cerrará la puerta al acceso inicial** y a la **ejecución remota de código (RCE)**.

3. Rotación Forzosa de Credenciales Sensibles

- **Acción**: Localizar las cadena de texto de las bases de datos y en los archivos de configuración de la aplicación y en cualquier servicio asociado. Se **debe revocar y cambiar inmediatamente por una contraseña robusta.**

- **Justificación**: Aunque la contraseña encontrada en la base de datos nos permitió entrar como adminsitrador al Login del servidor LosVengadores. Su descubrimiento en un contexto de desarrollo o aplicación supone un riesgo de reutilización de credenciales en otros entornos o servicios de la compañía.

II.
INFORME TÉCNICO

01. Sumario de imágenes recopiladas

Glosario de imágenes donde se facilitan cada una de las pruebas encontradas en la prueba de penetración.

02. Herramientas y Entorno

Aquí se listan las herramientas clave y el entorno técnico utilizado para llevar a cabo la evaluación, destacando la necesidad de la compilación cruzada.

03. Fases del Test de Penetración

Breve descripción donde se enumera cada fase de la prueba de penetración con un pequeño resumen de cada hallazgo encontrado.

01. Sumario de Imágenes recopiladas

Sumario:

02. Herramientas y Entorno:

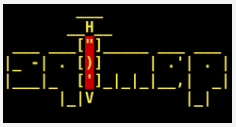
La evaluación requirió un conjunto de herramientas especializadas para las fases de enumeración, explotación web y, críticamente, la adaptación a la arquitectura específica del host víctima.

1. Entorno del Atacante (Kali Linux)

A. Herramientas de Reconocimiento y Escaneo

Herramienta	Aplicación Específica en la Evaluación	Imagen Corporativa
Nmap	Utilizado para el escaneo inicial de puertos y servicios. Se confirmó la apertura de puertos clave como **SSH (22)** y **HTTP (80)**, mapeando la superficie de ataque disponible.	
Dirsearch	Empleadas para la fase de ***fuzzing*** de directorios. El uso de estas herramientas permitió la **localización de rutas sensibles** y archivos de código fuente, lo que condujo al descubrimiento del *script* de subida de archivos y los puntos de entrada para la Inyección SQL.	

B. Herramientas de Explotación Web y Credenciales

Herramienta	Aplicación Específica en la Evaluación	Imagen Corporativa
SQLmap	Aunque la explotación principal de SQLi se realizó manualmente a través de Burp Suite, SQLmap **fue usado para la validación automatizada del bypass de autenticación** y la enumeración exhaustiva de la base de datos tras obtener el login.	
Burp Suite	Instrumento central en la explotación de la capa de aplicación. Fue crucial para manipular las requests de login para el **Bypass por SQLi** e interceptar y **alterar peticiones POST** en el formulario de subida de CV, permitiendo la **inyección del payload PHP para el RCE**.	

02. Herramientas y Entorno:

C. Herramienta de Conexión y Persistencia

Herramienta	Aplicación Específica en la Evaluación	Imagen Corporativa
Netcat	Sirvió como el listener principal para la gestión de las conexiones inversas (**Reverse Shells**). Su función fue esencial en un momento crítico: - **Recibir el shell inicial** de bajo privilegio (www-data) tras el RCE.	

03. Fases del Test de Penetración:

La evaluación se ejecutó siguiendo las fases estándar de un test de penetración (pentest), adaptadas al entorno de la aplicación "LosVengadores". La metodología probó la cadena completa de ataque, desde el descubrimiento de vulnerabilidades web hasta el intento de obtención del control total del sistema operativo.

1. Reconocimiento y Acceso Inicial

Esta fase se centró en mapear la superficie de ataque y la tecnología subyacente.

- **Identificación de Servicios**: Se confirmó la presencia de servicios clave como el **Servidor Web** (puerto 80) y **SSH** (puerto 22) abiertos, estableciendo una puerta de entrada potencial.

- **Enumeración de Directorios**: Mediante técnicas de **fuzzing** se identificaron directorios de aplicación expuestos, incluyendo el código fuente de la plataforma web

- **Determinación de SO y Arquitectura**: Se utilizó la enumeración pasiva para identificar la versión del sistema operativo (**Ubuntu 2.4.29**) y el servicio SSH (**OpenSSH 7.6p1**). Ambos con versiones antiguas expuestas a diferentes tipos de ataque.

- **Bypass de Autenticación (SQL Injection**): Se confirmó una vulnerabilidad de **Inyección SQL** en el **login de la aplicación**, que permitió eludir la autenticación y obtener acceso de usuario.

- **Exfiltración de Credenciales:** Se detectó una vulnerabilidad crítica de Inyección SQL (SQLi) en el mecanismo de autenticación del login de la aplicación, que permitió el acceso no autorizado y la exfiltración de información.

- **Ejecución de Código Remoto (RCE)**: Aprovechando una falla en el manejo de la subida de archivos (el script CV-upload.php) y la ausencia de validación, se logró subir y ejecutar una Web Shell o un payload PHP, resultando en la obtención de una Reverse Shell con el usuario de bajo privilegio www-data.

03. Fases del Test de Penetración:

2. Post Explotación

Una vez dentro del sistema, la prioridad fue buscar vectores de escalada de privilegios.

- **Búsqueda de Binarios SUID**: Se identificó el binario /usr/bin/pkexec y su versión 0.105.

- **Análisis de Vulnerabilidad**: Se confirmó que la versión **0.105 de pkexec** es **vulnerable al fallo de Pwnkit (CVE-2021-4034)**, estableciendo el vector de ataque más crítico.

3. Escalada de Privilegios

Esta fase fue la culminación de la cadena de ataque, no interfiriendo en esta última fase por estar fuera de nuestro alcance. Igualmente, cabe recalcar de nuevo el Pwnkit (versión 0.105.20) como falla más vulnerable en la que se nos permite tener la capacidad de accerder a los archivos del sistema como root, obteniendo el control absoluto del sistema.

II.I
PRUEBA DE PENETRACIÓN

01. Reconocimiento y Descubrimiento

Incluye una breve descripción, una fundamentación y el impacto previsto. Es recomendable que el mensaje sea conciso pero concreto.

02. Clasificación de Vulnerabilidades

Incluye una breve descripción, una fundamentación y el impacto previsto. Es recomendable que el mensaje sea conciso pero concreto.

01. Reconocimiento y Descubrimiento

La prueba de penetración se empieza realizando un *Nmap* en busca de los servicios y puestros que se encuentran activos en nuestra máquina víctima. (*10.0.2.21*)

Como ya dijimos anteriormente, nos encontramos ante **2 servicios activos** claramente diferenciados:

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 64 OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a5:7a:b3:b1:dc:63:c1:97:65:83:ca:c1:02:81:2f:46 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDXaJZsOs4o2kxsV8IRhBzSfSM78OMM6RkGCNsUOsGEAUqjOkjbcJpllsF6E
lHd9+nuGA2IbdIx12t7GRGMgpxpQI+7O/To0q0fPrF1r+ZE6B2NSgw4mJYqRZjQD+aM3tORjMfGYM6weic/wE2LD1QERVxdquaH
|   256 94:00:f3:7f:81:b2:67:03:c0:5e:33:77:92:db:3d:38 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAYkQvFNQaF9gnJCJg5jAnSN/
|   256 29:df:a0:81:84:cb:1c:9d:0e:ed:d2:94:ee:0f:02:c6 (ED25519)
| ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINqAfYW/Y39/r9OZb5xHCXgHRFsAOxkKRvKF936ShKq4
80/tcp open  http    syn-ack ttl 64 Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Infinite war - Los Vengadores 4.5
| http-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
MAC Address: 08:00:27:FE:0D:16 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Imagen 1: Identificando servicios

Una vez tenemos estos dos servicios abiertos lo que haremos será empezar añadiendo dicha IP víctima (*10.0.2.21*) a nuestro directorio */hosts* para que no tengamos problemas con la resolución DNS y nos permita entrar con total tranquilidad a dicha página web. Pero antes de eso utilizaremos la herramienta *Dirsearch* para ver que directorios a parte del principal puede encontrar que pueda servirnos de utilidad:

```
[08:20:58] 403 -  274B  - /.htaccess_extra
[08:20:58] 403 -  274B  - /.htaccess_orig
[08:20:58] 403 -  274B  - /.htaccess_sc
[08:20:58] 403 -  274B  - /.htaccessOLD
[08:20:58] 403 -  274B  - /.htaccessOLD2
[08:20:58] 403 -  274B  - /.htaccessBAK
[08:20:58] 403 -  274B  - /.htm
[08:20:58] 403 -  274B  - /.html
[08:20:59] 403 -  274B  - /.htpasswds
[08:20:59] 403 -  274B  - /.htpasswd_test
[08:20:59] 403 -  274B  - /.httr-oauth
[08:20:59] 403 -  274B  - /.php
[08:21:08] 301 -  311B  - /javascript  →  http://10.0.2.21/javascript/
[08:21:12] 301 -  311B  - /phpmyadmin  →  http://10.0.2.21/phpmyadmin/
[08:21:12] 200 -    3KB - /phpmyadmin/doc/html/index.html
[08:21:12] 200 -    3KB - /phpmyadmin/
[08:21:12] 200 -    3KB - /phpmyadmin/index.php
[08:21:14] 200 -   40B  - /search.php
[08:21:14] 403 -  274B  - /server-status
[08:21:14] 403 -  274B  - /server-status/
```

Imagen 2: Enumerando directorios

Nos aparecen **varios directorios de utilidad** asique nos los guardaremos para tenerlos en cuenta más adelante.

01. Reconocimiento y Descubrimiento

1. **Dentro de la Web:**

Dentro de la página web vemos a simple vista varias secciones que nos llaman bastante la atención y en las que **son bastante probables realizar ataques** para comprometer la seguridad de la página:

Imagen 3: Identificando paneles

De entre todas estas secciones la primera en la que sacamos la primera vulnerabilidad es en la sección de *BUSCADOR*, donde nos damos cuenta que es **vulnerable a un ataque de SQLi.** ¿Cómo lo sabemos?. Básicamente porque al ejecutar el script ***‘ OR ‘ 1’=’1*** nos refleja que la conectividad con la base de datos es correcta bajo el mensaje:

Imagen 4: Panel vulnerable a SQLi

Una vez sabemos ésto nuestro próximo paso será utilizar la herramienta *SQLMap* para ver que tipo de ataque es el vulnerable.

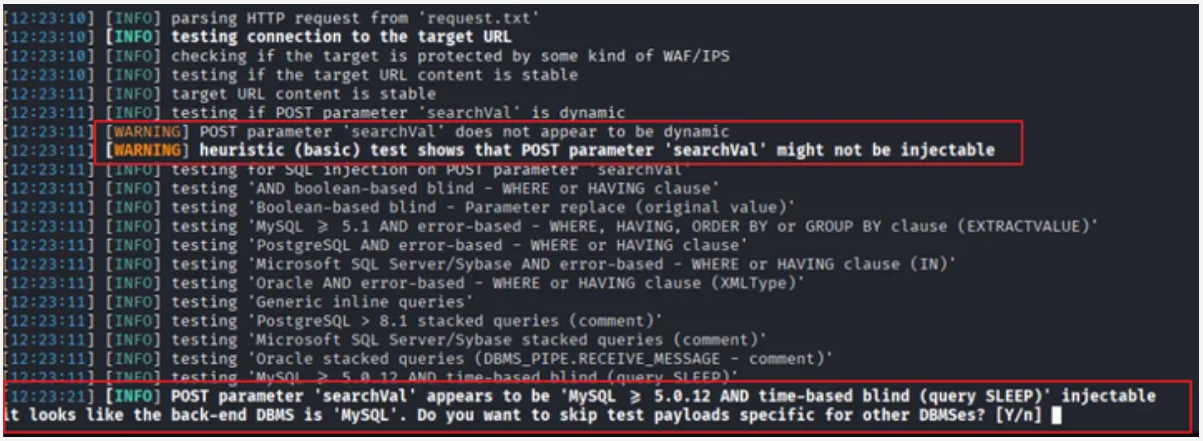


Imagen 5: SQL Injection

01. Reconocimiento y Descubrimiento

2. Explorando SQLMap:

En la imagen anterior nos refleja que, exactamente **se trata de una base de datos** en la cual el parámetro searchval es **vulnerable a *SQL injection***, específicamente **SQLi Time-Based Blind** (*Inyección SQL ciega basada en tiempo*).

Una vez sabemos ésto nuestro próximo será tratar de explotar esta base de datos con un payload específico para que nos muestre que tipo de información se encuentra en dicha tabla.

Una vez ejecutamos los payloads pertinentes nos despliega información altamente crítica a nivel de vulnerabilidad:

```
Database: vengadores
Table: users
[1 entry]
+----+-------------+---------------+----------------------------------------------------+
| id | email       | username      | password                                           |
+----+-------------+---------------+----------------------------------------------------+
| 0  | info@gba.us | capitanamerica | 6f2f0046544e6821b04c99ec8cdb98f4 (adamantium) |
+----+-------------+---------------+----------------------------------------------------+
```

Imagen 6: SQLMap - users

```
Database: vengadores
Table: fundadores
[5 entries]
+-----------------+-----------------------------------------------+
| heroe           | consigna                                      |
+-----------------+-----------------------------------------------+
| avispa          | fcdddc8c109hskw94e689d9852fcf                 |
| hombre_hormiga  | fcdddc8c109a6dc18814e689d9852fcf              |
| Hulk            | fc44ddc8c144hskw94e689d9852fcf                |
| ironman         | 93952a75e6131a269680f15a082c2a28              |
| thor            | fcc8c10bc39a6dc18814e689d9852fcf (guapeton)   |
+-----------------+-----------------------------------------------+
```

Imagen 7: SQLMap - fundadores

Nos muestra diferentes credenciales y usarios de diferentes tablas en las cuales podemos utilizar para intentar loguearnos en algún servicio tanto fuera como dentro de la web para intentar acceder a información comprometida y así poder seguir escalando dentro del host víctima (*10.0.2.21*)

01. Reconocimiento y Descubrimiento

3. Login en LosVengadores:

Una vez hemos obtenido en la base de datos diferentes credenciales probamos con la que tiene más probabilidad de poder acceder en el panel de *LOGIN*:

- **Usuario**: *capitanamerica*
- **Contraseña**: *adamantium*

Y efectivamente **conseguimos loguearnos en la página web** en la que, a simple vista no tiene mucho más que una especie de buscador:

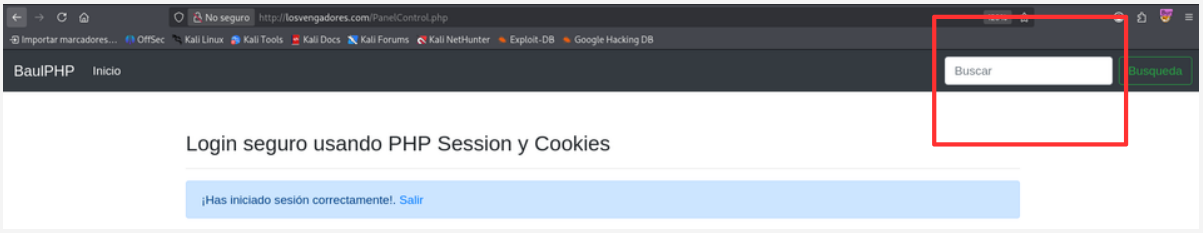


Imagen 8: Login exitoso

Volviendo de nuevo a la página web principal y accediendo a la fuente de la página (*F12*) nos cercioramos que hay parte del código comentado de lo que parece ser un subidor de archivos llamado ***<!-- CV Upload – >***

Si observamos el código fuente del formulario, se puede ver que llama a una función llamada ***"upload_cv()"***

```
<!-- CV Upload -->

<div class="col-sm-12 col-md-6">
          <form action="DEPRECATED" method="get">
            <input id="cv" name="cv" type="file" placeholder="Your CV" />
            <button class="btn tm-btn-submit" onclick="upload_cv()">Submit</button>
          </form>
        </div>
```

Imagen 9: <! - - CV Upload - - >

Además un poco más abajo vemos como una sección llamada "***EL SECRETO***" tiene una fuente js que te lleva a un recurso ***cv_upload.min.js***

```
<!--ELSECRETO-->
<section id="secreto" class="tm-section-pad-top tm-parallax-2">…</section>
<script src="js/jquery-1.9.1.min.js"></script>
<script src="slick/slick.min.js"></script>
<script src="magnific-popup/jquery.magnific-popup.min.js"></script>
<script src="js/easing.min.js"></script>
<script src="js/jquery.singlePageNav.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/cv_upload.min.js"></script>
<script>…</script>
```

Imagen 10: js/cv_upload.min.js

01. Reconocimiento y Descubrimiento

4. upload_cv():

Si nos metemos en dicha función podemos ver como trabaja dicho script a la hora de poder subir un formulario a la web:

```
function upload_cv(){

        // Logged users
        let cv = document.getElementById('cv').files[0];
        let formData = new FormData();

        formData.append("cv",cv);
        fetch('CV-upload.php',{"method": "POST", body: formData});
}
```

Imagen 11: function upload_cv()

La función ***upload_cv()*** está implementada en el lado del cliente (*navegador*) y es **responsable de iniciar la subida de archivos del usuario**. Esta función es clave para la vulnerabilidad de *Ejecución de Código Remoto (RCE)* debido a su diseño minimalista y a la ausencia de validaciones de seguridad.

Es decir, la función actúa como un disparador que **delega completamente la responsabilidad de seguridad** en el script de destino (*CV-upload.php*). Este diseño permite que un atacante (en este caso nosotros) pueda:

- **Eludir** cualquier potencial filtro del lado del cliente.
- **Subir un archivo** con extensión peligrosa (ej., .php) al sistema.
- E**jecutar el archivo subido**, resultando en la obtención de una *Reverse Shell (RCE*), lo que fue el punto de partida para la escalada de privilegios.

Una vez sabemos la extrema vulnerabilidad que tiene dicha función el próximo paso será, con la información que tenemos ahora, ver como podemos ejecutar dicho código en el código fuente de la página principal.

Para realizar ésto tendremos que entender la función que se realiza en la parte comentada mencionada anteriormente (*véase Imagen 9: - - CV Upload - - >*)

01. Reconocimiento y Descubrimiento

5. <!-- CV-Upload – – >:

El código fuente del formulario de subida de archivos revela un **diseño que anula el comportamiento estándar de HTTP,** delegando toda la responsabilidad del envío y la validación a una función JavaScript del lado del cliente

La combinación de un action inválido y la ejecución de la función ***upload_cv()*** vía **onclick** demuestra que la aplicación **está diseñada para evitar los mecanismos de envío de formularios nativos.**

Esto tiene dos consecuencias críticas:

- **Omisión de Controles HTML**: La aplicación no utiliza los controles básicos de seguridad que el estándar HTML podría ofrecer.

- **Exposición al Proxy**: El atacante puede saltarse la función ***upload_cv()*** (*el JS*) y el onclick simplemente **interceptando la petición POST** de Burp Suite y modificando el nombre o tipo del archivo a una extensión peligrosa, lo que **confirma la vulnerabilidad de RCE (Ejecución de Código Remoto).**

Todo este tipo de información juntado también a la de la función **upload_cv()** nos da la clave necesaria para **poder realizar sin mayor problema una Reverse Shell**, ya que, como vimos anteriormente, la aplicación no tiene filtros complejos vinculados al evento ***onclick*** del botón o al evento ***onsubmit*** del formulario. SI fuese así tendríamos que simular el clic. Pero como la única lógica está en ***upload_cv()***, al ejecutarla directamente, garantizas que el envío (*fetch*) se realiza inmediatamente después de que el archivo es seleccionado y ejecutado.

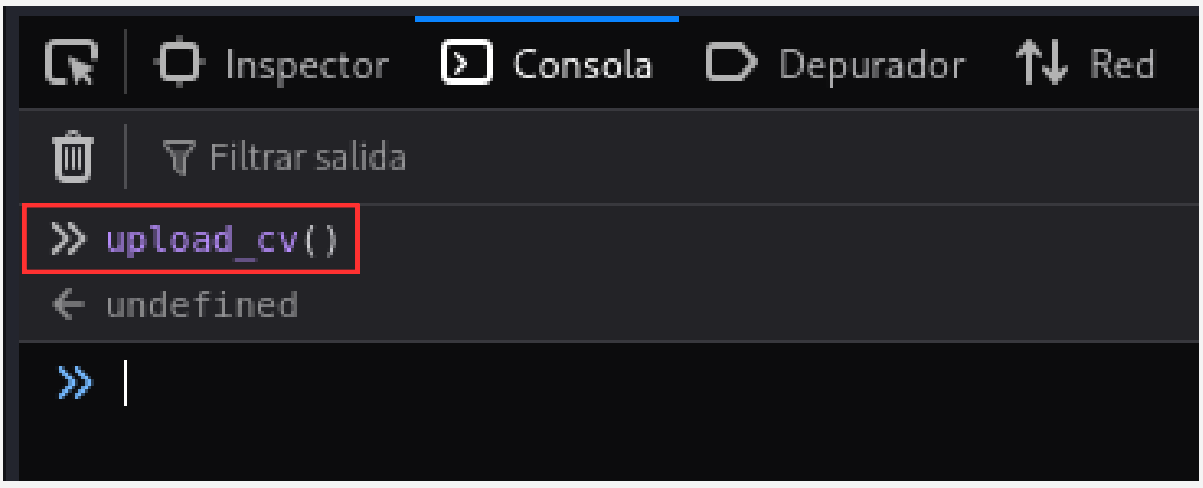


Imagen 12: Desde consola: upload_cv()

01. Reconocimiento y Descubrimiento

6. RevereShell.php:

Una vez sabemos que es lo que hace exactamente las funciones y sabiendo que **no hay ninguna sanitización de subida de archivos.** Creamos nuestro archivo ***.php*** en nuestra Kali para poder ejecutar una *Reverse Shell* dentro de la web y así poder obtener uno de los mayores ataques en los test de penetración, el **Remote Control Excute (RCE)**.

Una vez tenemos nuestro archivo ***ReverseShell.php,*** nos hemos logeado en nuestro usuario y hemos descomentado el código ***<!-- CV-Upload – – >*** lo único que nos falta es añadir nuestro ***ReverseShell.php*** en la casilla de subida y ejecutar el ***cv_upload()*** desde la consola de herramientas de programador. Pero no sin antes abrir nuestro **Burpsuite** para poder interceptar la request y ver que tipo de respuesta nos ofrece.

Imagen 13: Subiendo el ReverseShell.php)

01. Reconocimiento y Descubrimiento

6. Burpsuite:

Ahora, lo único que tenemos que hacer es, una vez hemos ejecutado la petición desde la consola de herramientas del programador es **poner el Burpsuite interceptando la petición** para que, una vez nos lo intercepte, ver el tipo de respuesta que nos realiza la propia página web

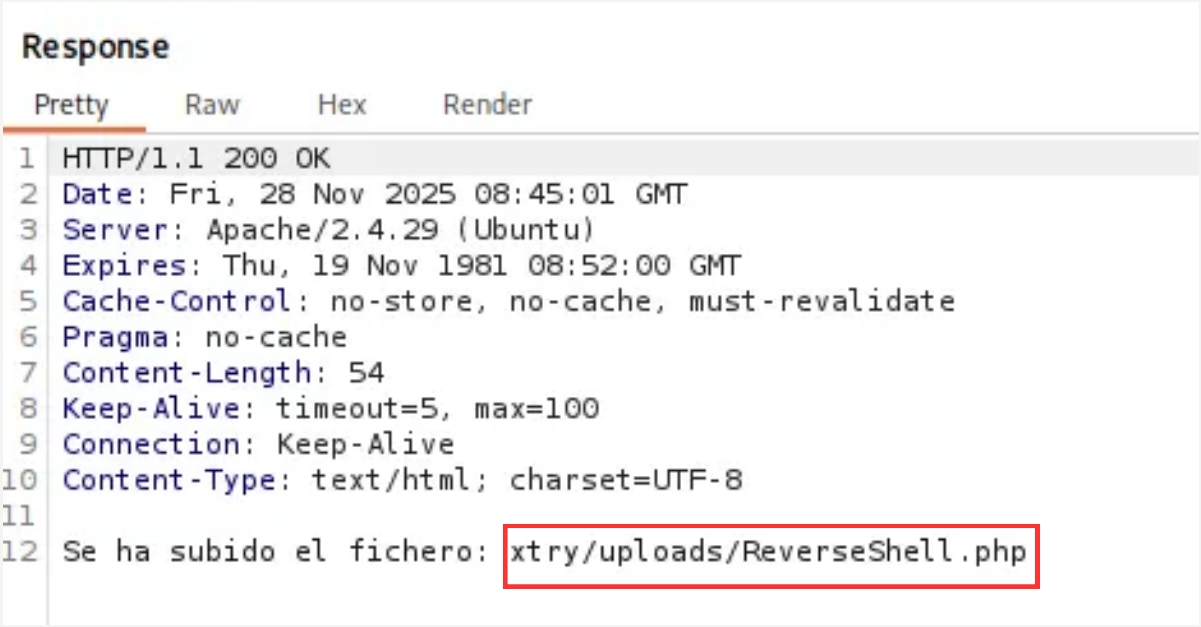


Imagen 14: Interceptando en BurpSuite

7**. Netcat:**

Una vez sabemos donde se encuentra nuestra ***ReverseShell.php*** deberemos poner nuestro *netcat* a la escucha en el puerto correspondiente.

Entraremos en dicha carpeta (xtry/uploads/ReverseShell.php) y ya tendremos acceso en nuestra propia terminal para poder ejecutar sin ningún problema la **Ejecución Remota de Código** desde nuestra Kali Linux

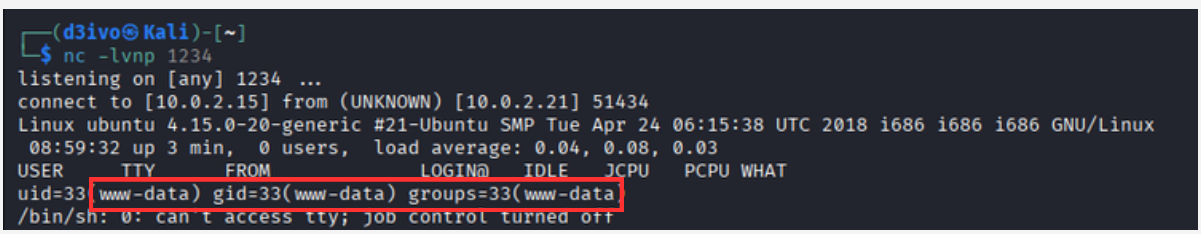


Imagen 15: Escuchando en Netcat

01. Reconocimiento y Descubrimiento

6. RCE y pkexec:

Una vez tenemos acceso al sistema y vemos que tipo de usuario somos (*www-data*) **podemos intentar realizar una escalada de privilegios**.

Tras obtener un reverse shell con privilegios de *www-data*, la fase de enumeración local se enfocó en identificar binarios del sistema que pudieran tener permisos especiales, violando el ***Principio del Mínimo Privilegio***.

El binario */usr/bin/pkexec* fue rápidamente identificado en la lista de *binarios SUID*. Este binario es la utilidad principal para la **gestión de políticas de seguridad** *(PolicyKit*) y **permite a los usuarios ejecutar comandos como si fueran otros usuarios** (mayormente root).

```
www-data@ubuntu:/usr/bin$ dpkg -l policykit-1
dpkg -l policykit-1
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version      Architecture Description
+++-==============-============-============-=================================
ii  policykit-1    0.105-20     i386         framework for managing administra
www-data@ubuntu:/usr/bin$
```

Imagen 16: Versión del pkexec

Se procedió a verificar la versión exacta del paquete asociado y se confirmó como la **versión 105.20**, con una **arquitectura de 32bits**.

```
www-data@ubuntu:/usr/bin$ ls -la pkexec
ls -la pkexec
-rwsr-xr-x 1 root root 21864 Mar 27  2018 pkexec
```

Imagen 17: Permisos del pkexec

La **versión 105.20** del binario */usr/bin/pkexec* es **susceptible a la vulnerabilidad conocida como Pwnkit.**

Esta vulnerabilidad es un **fallo de corrupción de memoria** (*buffer overflow*) que existe en ***pkexec***. El defecto **permite a un atacante inyectar y ejecutar código arbitrario con privilegios de root.**

A pesar de no haber ejecutado el exploit finalizado en el informe (porque está fuera de alcance), el análisis técnico confirma que **el sistema es completamente vulnerable**, dejando solo la fase de ejecución pendiente para la obtención del root shell.

02. Hallazgos Críticos y Vulnerabilidades

Vulnerabilidad Crítica

Escalada de Privilegios Local (ROOT)

- **Identificador**: Pwnkit (CVE-2021-4034)
- **Vector de Ataque**: Defecto en el binario */usr/bin/pkexec* (*versión 105.20*).
- **Mecanismo:** La versión obsoleta del paquete *policykit-1* permite a un usuario de bajo privilegio (www-data) manipular los argumentos del programa y ejecutar código arbitrario con privilegios de root.
- **Impacto Confirmado:** Compromiso total del sistema, permitiendo al atacante el control administrativo absoluto.

FILA	CONTENIDO
ID	WSTG-ATHZ-03
CVWE	CWE-787: Out-of-bounds Write
CVSS / VECTOR	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
CVE	CVE-2021-4034
IMPACTO	7.80 **(HIGH)**
DESCRIPCÓN	La vulnerabilidad Pwnkit en el binario pkexec (marcado como SUID) permite que un usuario con privilegios mínimos **manipule los parámetros de entrada del programa**. Esto causa que el sistema ejecute código malicioso **a nivel de root**, logrando así la escalada total de privilegios en la máquina.
SOLUCIÓN	La solución para Pwnkit es el **parcheo inmediato** del paquete **policykit-1** (para eliminar la vulnerabilidad) o, como medida temporal de emergencia, **eliminar el permiso SUID** del binario pkexec. Dada la antigüedad del sistema, la **migración a un sistema operativo moderno** es la única solución definitiv

02. Hallazgos Críticos y Vulnerabilidades

Vulnerabilidad Alta

Ejecución de Código Remoto (RCE)

- **Vector de Ataque**: Subida de archivos sin validación en el script *CV-upload.php*.
- **Mecanismo**:
  a. La función *upload_cv()* **delega toda la responsabilidad de seguridad en el servidor**, ya que no realiza validación en el lado del cliente.
  b. La configuración HTML (<form action="DEPRECATED" method="get">) se ignoró, y se utilizó la llamada directa a *upload_cv()* para **forzar el envío sin filtros**.
  c. La omisión de validación de tipo de archivo en el servidor permitió la subida de un **archivo .php malicioso**. (ReverseShell.php)
- **Impacto Confirmado:** Obtención de una *Reverse Shell* con el usuario *www-data*. Esto proporcionó el acceso inicial al sistema operativo, indispensable para iniciar la escalada a root.

FILA	CONTENIDO
ID	WSTG-INPV-19
CVWE	CWE-20: Improper Input Validation
CVSS / VECTOR	CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H
CVE	CVE-2017-15715
IMPACTO	8.1 **(HIGH)**
DESCRIPCÓN	Permite a un atacante **evadir las restricciones de seguridad** de **Apache HTTP Server** (v. 2.4.0 a 2.4.29). Esto causa que los filtros de seguridad del servidor se anulen, permitiendo la **ejecución de código remoto (RCE)** y el compromiso del servidor web.
SOLUCIÓN	La solución para CVE-2017-15715 es actualizar de inmediato Apache HTTP Server a la versión 2.4.30 o posterior para corregir el fallo de manejo de nombres de archivo y prevenir la Ejecución de Código Remoto (RCE).

02. Hallazgos Críticos y Vulnerabilidades

Vulnerabilidad Alta

Bypass de Autenticación (SQL Injection)

- **Vector de Ataque**: Campos de entrada del formulario de login de la aplicación.
- **Mecanismo:** La aplicación no sanitiza correctamente los inputs del usuario, permitiendo la inyección de código SQL que alteró la lógica de la consulta de autenticación.
- **Impacto Confirmado:**
  - Acceso al Panel de Control de la aplicación web sin credenciales válidas.
  - Exposición de Credenciales de la base de datos (DB), incluyendo el hash de la contraseña del usuario capitanamerica (6f2f0046544e6821b04c99ec8cdb98f4).

FILA	CONTENIDO
ID	WSTG-INPV-05
CVWE	CWE-89: 'SQL Injection'
CVSS / VECTOR	CVSS: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
CVE / OSWAP	A03:2021 – Inyección
IMPACTO	7.8 **(HIGH)**
DESCRIPCÓN	La Inyección SQL es un ataque que consiste en insertar código SQL malicioso a través de la entrada de datos de la aplicación. El ataque permite al atacante leer, modificar o eliminar datos sensibles de la base de datos y, en casos graves, incluso ejecutar comandos de administración o acceder al sistema operativo del servidor.
SOLUCIÓN	La solución para Inyección SQL es el uso de consultas parametrizadas (Prepared Statements) en todo el código. Esto garantiza que la entrada del usuario sea tratada siempre como datos y nunca como código SQL, eliminando la vulnerabilidad.

II.II
CONCLUSIÓN
Y RECOMENDACIONES DE MITIGACIÓN

01. Conclusión General

Conclusión final de los hallazgos encontrados en el servidor basado en un análisis exhaustivo en la prueba de penetración

02. Recomendaciones

Recomendaciones de mitigación finales para parchear, actualizar o sanitizar las vulnerabilidades encontradas en la prueba de penetración.

01. Conclusiones Generales

El resultado de la evaluación de seguridad sobre el host de "LosVengadores" establece un **Nivel de Riesgo Operacional CRÍTICO e inaceptable**. Se ha demostrado de manera concluyente una cadena de explotación que permite a un atacante, partiendo de un acceso público, obtener el control total del servidor.

Factores Críticos Identificados:

- **Falla de Arquitectura**: La dependencia de un sistema operativo obsoleto (Ubuntu 2.4.29 con arquitectura i686) y la versión vulnerable del binario pkexec (v. 105.20) garantiza que la escalada de privilegios sea posible.

- **Falla de Validación en la Aplicación:** Las vulnerabilidades de Inyección SQL y la no validación de Archivos del lado del servidor (RCE) en la función **upload_cv()** evidencian graves defectos en el código de la aplicación que proporcionan un acceso inicial fácil al sistema operativo.

- **Ausencia del Principio del Mínimo Privilegio:** El usuario comprometido (*www-data*) tenía la capacidad de interactuar con binarios SUID como pkexec, lo que demuestra una falla en la escalada de privilegios del sistema.

02. Recomendaciones de Mitigación

La solución requiere un enfoque de mitigación en tres capas: Sistema Operativo, Configuración, y Aplicación Web.

A. Prioridad CRÍTICA: Fallos de Infraestructura y Sistema

La vulnerabilidad en el sistema operativo permite el control absoluto del servidor.

Hallazgo	Recomendación de Mitigación
Pwnkit (pkexec v. 105.20) y Kernel Obsoleto	**Migración Obligatoria e Inmediata:** La plataforma con **kernel 2.4.29** es insalvable. El servidor debe migrarse a una **distribución moderna y con soporte** para eliminar cientos de CVEs conocidos.

Pwnkit (Mitigación Temporal)
Si la migración es imposible a corto plazo, eliminar el flag SUID del binario pkexec para que los usuarios sin privilegios no puedan ejecutarlo con permisos de root

- P**rincipio del Mínimo Privilegio:** Asegurar que el usuario de bajo privilegio (www-data) no tenga permisos de ejecución sobre binarios del sistema que no necesite.

B. Prioridad ALTA: Ejecución de Código y Autenticación (RCE & SQLi)

Estos fallos permiten el acceso inicial al sistema y el bypass de seguridad.

Hallazgo	Recomendación de Mitigación
RCE por Subida de Archivos No Restringida	La función CV-upload.php debe ser modificada para **Rechazar** cualquier archivo que no esté en una lista blanca.
SQL Injection	Reescribir cualquier código de la aplicación que interactúe con la base de datos (DB) para utilizar consultas parametrizadas. para que la entrada del usuario sea tratada siempre como **datos** y nunca como **código SQL**.
Credenciales Expuestas	**Rotación de Credenciales:** Cambiar inmediatamente todas las contraseñas, especialmente la del usuario capitanamerica, y las credenciales de conexión a la base de datos.

02. Recomendaciones de Mitigación

C. Prioridad MEDIA: Fallos de Servidor

Hallazgo	Recomendación de Mitigación
Apache obsoleto	**Actualizar Apache HTTP Server:** Actualizar el servidor web a la **versión 2.4.58 o superior**. Esto corregirá la vulnerabilidad de Denegación de Servicio (DoS) en el módulo mod_macro.
Falta de Detección de Intrusos	**Implementar Logs y Monitoreo:** Activar un sistema de registro y monitoreo centralizado (OWASP A09). Esto debe incluir monitoreo de integridad de archivos y alertas sobre ejecuciones inesperadas de *shells* (sh o bash) por parte del usuario www-data.
Información Expuesta	**Reconfiguración de directorios:** Asegurar que los archivos de configuración, copias de seguridad, y directorios críticos no sean accesibles a través del navegador.