Enumeración de Usuarios con enum4linux

Después de la fuerza bruta, se utilizó **enum4linux** para intentar obtener información del sistema objetivo.

Acción realizada:

1. Ejecuté el siguiente comando en la terminal:

enum4linux -a boa.up.railway.app

Resultados:

- Sin embargo, el servidor no permitió conexiones de sesión vacías, por lo que no se pudieron realizar más pruebas con esta herramienta.

Búsqueda de Directorios Ocultos con dirb

Luego, utilicé **dirb** para buscar directorios y archivos ocultos en el servidor objetivo.

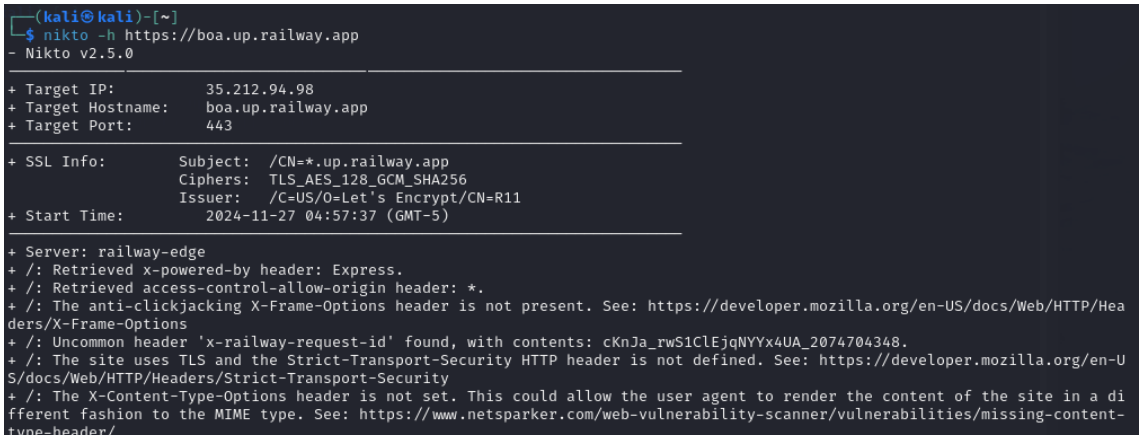
Acción realizada:

1. Ejecución del siguiente comando:

dirb https://boa.up.railway.app/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -r

se encontraron 2 archivos interesantes para analizar

Uso de nikto para ver las vulnerabilidades generales y configuraciones generales



X-Frame-Options no presente:

Problema: Ausencia de esta cabecera permite que el sitio pueda ser cargado en un iframe por sitios maliciosos, lo que lo hace vulnerable a ataques de clickjacking.

Solución: Agregar la cabecera X-Frame-Options con valores como DENY o SAMEORIGIN para prevenir estos ataques.

Strict-Transport-Security (HSTS) no configurado:

Problema: La falta de esta cabecera significa que los navegadores no están obligados a usar conexiones seguras (HTTPS).

Solución: Configurar HSTS en el servidor para forzar el uso de HTTPS.

X-Content-Type-Options no configurado:

Problema: Esto podría permitir a los navegadores interpretar los archivos de manera incorrecta, aumentando el riesgo de ataques.

Solución: Configurar la cabecera X-Content-Type-Options a nosniff para evitar que los navegadores "adivinen" el tipo de contenido.

Vulnerabilidad a BREACH:

Problema: El uso de la codificación deflate en la cabecera Content-Encoding puede hacer que el servidor sea vulnerable al ataque BREACH, el cual permite exfiltrar datos sensibles cuando la página tiene contenido dinámico.

Solución: Considera deshabilitar la compresión en las respuestas HTTP para solicitudes sensibles. También podrías usar técnicas de mitigación como fragmentar la información sensible o evitar el uso de compresión.

Uso de Certificado Wildcard:

Problema: El uso de certificados wildcard no es inherentemente inseguro, pero debe gestionarse correctamente para evitar exponer subdominios de manera inadvertida.

Solución: Asegúrate de que el uso del wildcard no exponga subdominios no necesarios.

Desarrollo y Ejecución de un Script de Fuerza Bruta

Primero, se desarrolló un script en Python para intentar una prueba de fuerza bruta sobre el endpoint de autenticación de la aplicación.

Acciones realizadas:

1. Se definió el endpoint: https://backendfitmrp-production.up.railway.app/api/auth/login.
2. Se creó una lista de posibles nombres de usuario y contraseñas.
3. Se programó el script para enviar solicitudes POST con combinaciones de usuario/contraseña.

Paso 4: Escaneo con Metasploit

Finalmente, usé Metasploit para explorar posibles vulnerabilidades y configuraciones del servidor.

Acción realizada:

1. Inicié Metasploit con:

bash

Copiar código

msfconsole

2. Utilicé el módulo auxiliary/scanner/http/robots_txt para buscar restricciones en el archivo robots.txt:

use auxiliary/scanner/http/robots_txt

set RHOSTS boa.up.railway.app

set RPORT 443

run

encontramos rutas del backend como ser