Informe Técnico - Sistema Z

Realizado por: Grupo 06

Cliente: XY

Alcance: Evaluación de seguridad de los servicios y aplicación web del sistema Z.

Resumen Ejecutivo

El objetivo de este pentest fue evaluar la seguridad del servidor y la aplicación web de la empresa SistemaZ. Se identificaron vulnerabilidades críticas en los servicios SSH y MySQL, así como en la aplicación web. Los hallazgos incluyen el uso de versiones obsoletas de software, contraseñas débiles y la exposición de credenciales en texto plano. Se recomienda realizar mejoras inmediatas en la seguridad del servidor y la gestión de contraseñas, además de implementar políticas de control de acceso más robustas.

Alcance de la evaluación

Activos evaluados

- Servicios identificados : SSH, Apache, MySQL
- Aplicación web.

Herramientas utilizadas

- Nmap, Metasploit, ZAP Proxy, MySQL client.

Metodología

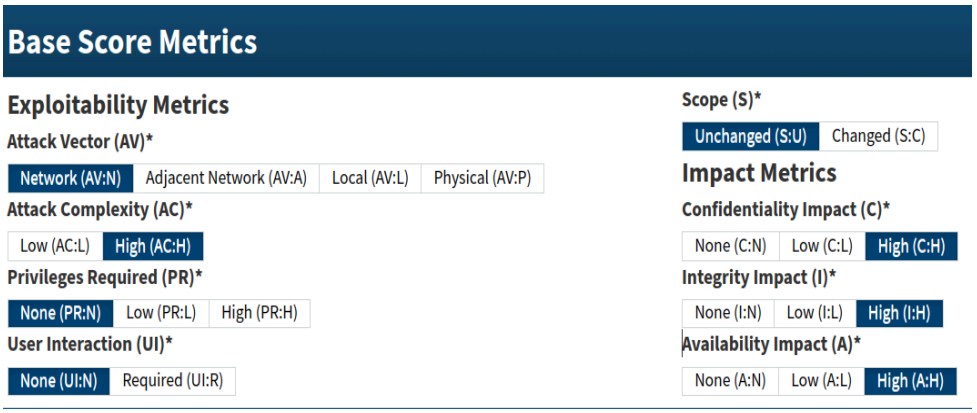
El pentest se realizó en las siguientes fases:

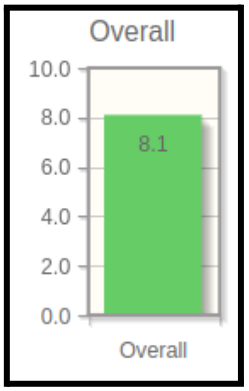
1. Reconocimiento: Identificación del objetivo y sus servicios activos utilizando herramientas como NMAP.
2. Escaneo de vulnerabilidades: Utilización de Metasploit y pruebas manuales para identificar vulnerabilidades en SSH, Apache y MySQL.
3. Explotación: Explotación de vulnerabilidades encontradas para ganar acceso a sistemas críticos.
4. Post-explotación: Acceso y recolección de información sensible de las bases de datos y del sistema.
5. Reporte: Documentación de hallazgos, impacto y recomendaciones.

Vulnerabilidades detectadas

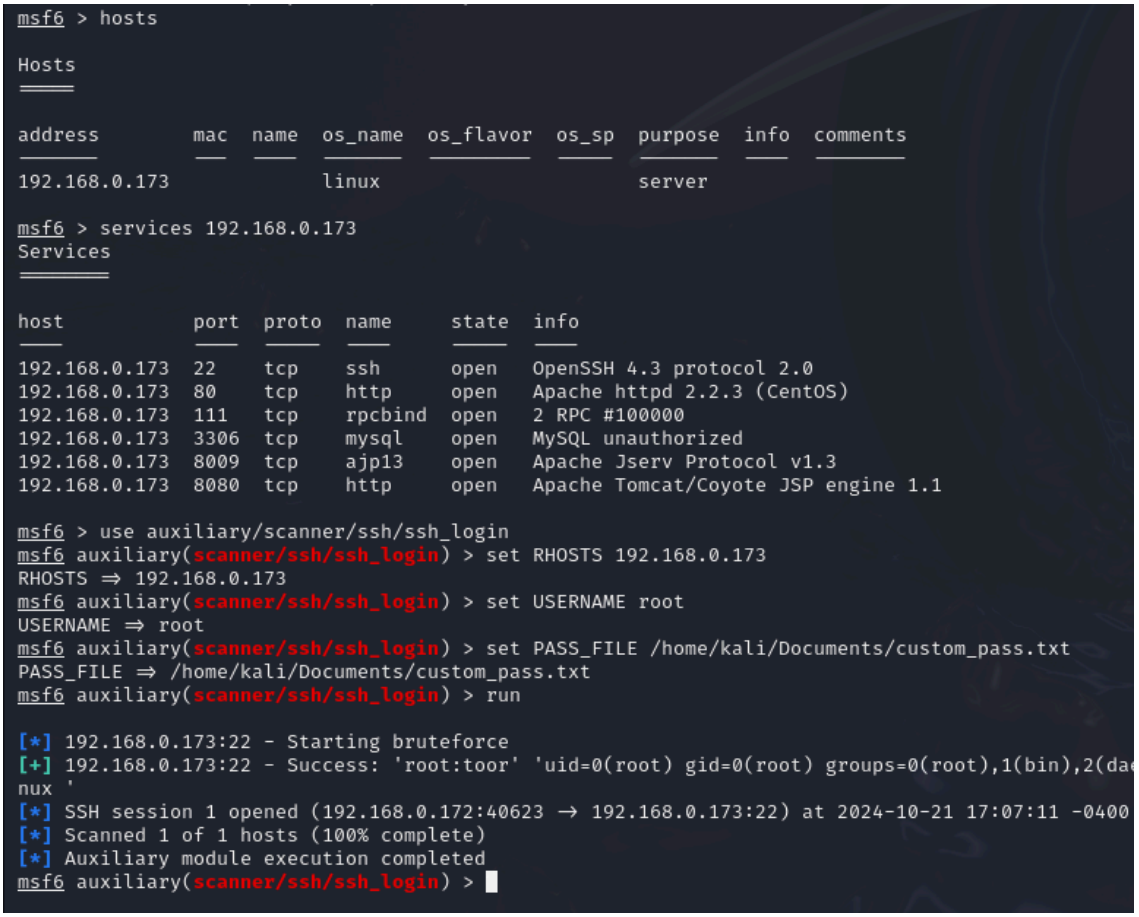
4.1. Vulnerabilidad 1: Versión obsoleta de Software

- Descripción: El servidor SSH está corriendo la versión 4.3 de OpenSSH, que es vulnerable a múltiples ataques conocidos, incluidos ataques de fuerza bruta.
- Impacto: Un atacante puede comprometer el servicio SSH, obtener acceso al servidor y escalar privilegios hasta el nivel de root.
- Prueba de Concepto: Se realizó un ataque de fuerza bruta exitoso utilizando Metasploit (auxiliary/scanner/ssh/ssh_login), accediendo como root con la contraseña toor.
- Riesgo: **Alto.**
- Solución: Actualizar a la última versión estable de OpenSSH.



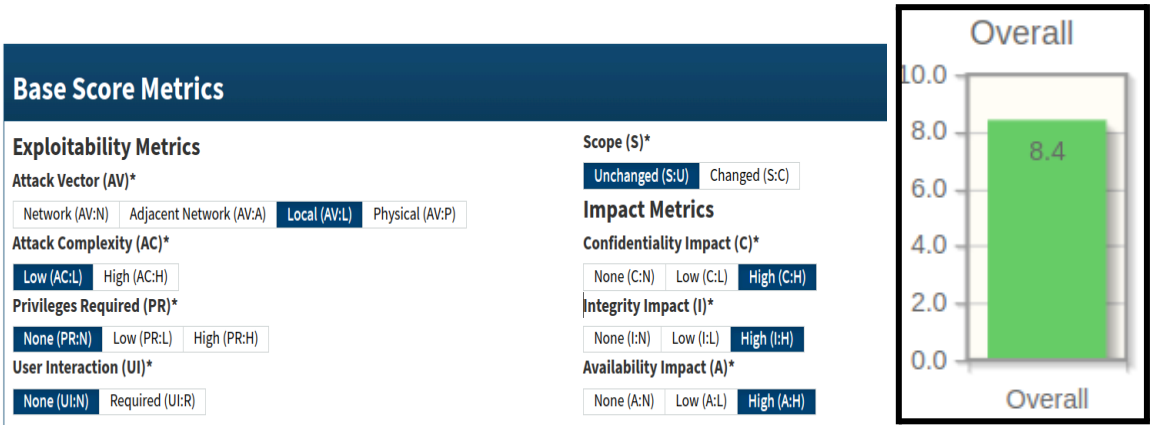


Evidencias



4.2. Vulnerabilidad 2: Credenciales expuestas en el historial de comandos

- Descripción: Las credenciales de la base de datos MySQL (root: Chile2010) se encontraron almacenadas en el archivo .bash_history del sistema. Esto permite a cualquier persona con acceso al servidor visualizar las credenciales sin necesidad de ningún exploit avanzado.
- Impacto: Un atacante con acceso al sistema puede obtener estas credenciales y comprometer la base de datos.
- Prueba de Concepto: Al acceder al archivo /root/.bash_history, se encontró el comando mysql --user=root --password=Chile2010.
- Riesgo: **Alto.**
- Solución: Borrar el archivo de historial o las entradas sensibles. Configurar el shell para no almacenar comandos que incluyan credenciales. Utilizar un archivo de configuración seguro para manejar las contraseñas.

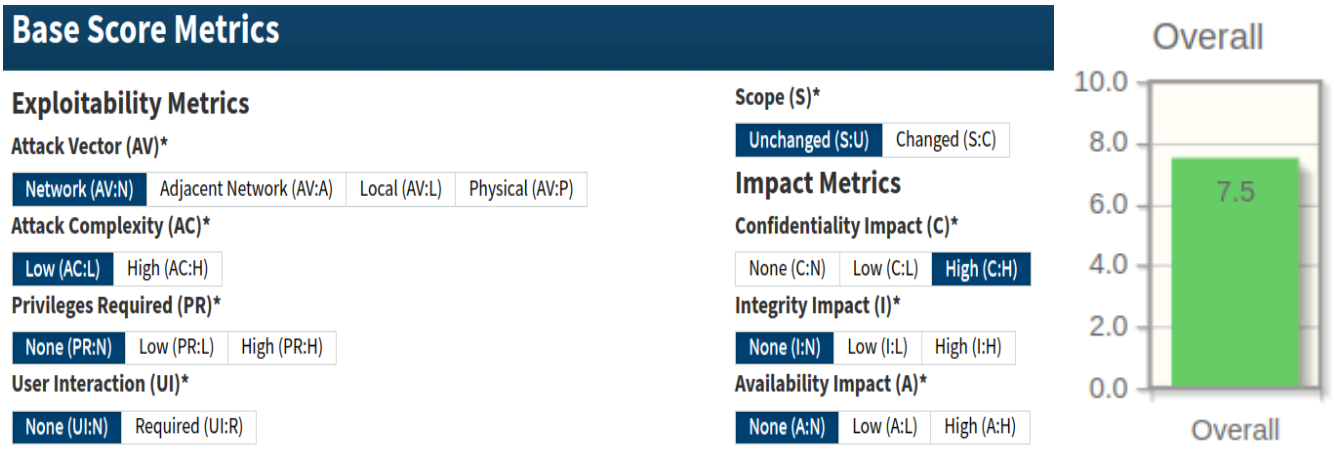


Evidencias

```
s
cat /root/.bash_history
mysql
mysql --user=root --password=chile2010
mysql --user=root --password=YES
mysql --user=root --password=Chile2010
exit
```

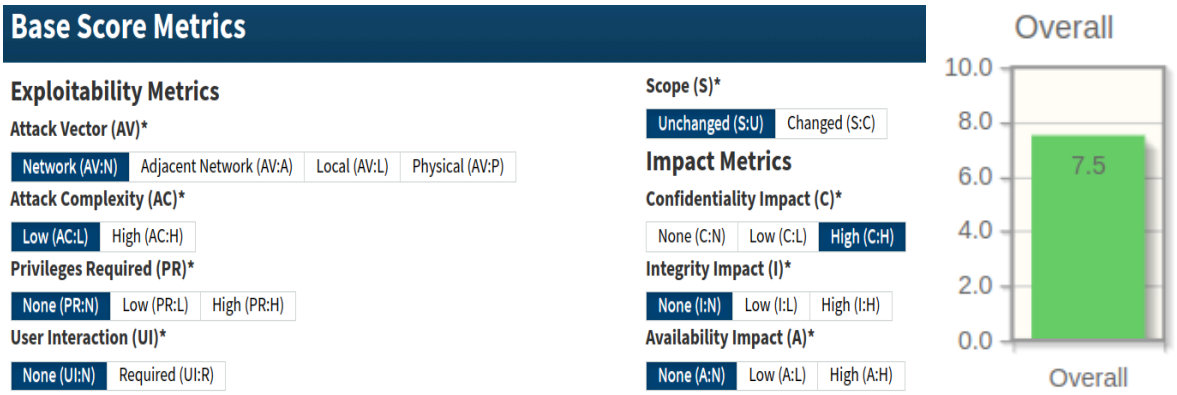
4.3. Vulnerabilidad 3: Contraseñas débiles en usuarios de la base de datos

- Descripción: Se identificaron varias cuentas de usuario en la base de datos xy_sistemaz con contraseñas débiles como ‘1234’,’ password’, o ‘pedro’.
- Impacto: Un atacante puede explotar estas credenciales fácilmente mediante ataques de diccionario o fuerza bruta, comprometiendo las cuentas de los usuarios.
- Prueba de Concepto: A través de MySQL, se accedió a la tabla usuario y se obtuvieron las contraseñas en texto plano de todos los usuarios.
- Riesgo: **Media.**
- Solución: Implementar políticas de contraseñas más seguras que requieran combinaciones complejas de caracteres (mayúsculas, minúsculas, números y símbolos). Forzar el cambio de contraseñas para los usuarios afectados.



4.4. Vulnerabilidad 4: Contraseñas almacenadas en texto plano

- Descripción: Las contraseñas de los usuarios en la base de datos xy_sistemaz están almacenadas en texto plano, lo que permite a cualquier persona con acceso a la base de datos leerlas sin ninguna protección.
- Impacto: Un atacante puede obtener y utilizar estas contraseñas para comprometer las cuentas de los usuarios en el sistema.
- Prueba de Concepto: Al realizar una consulta SQL (SELECT * FROM usuario), se mostraron las contraseñas en texto claro.
- Riesgo: **Alta.**
- Solución: Implementar algoritmos de hash seguros como bcrypt o Argon2 para almacenar las contraseñas en lugar de guardarlas en texto plano.

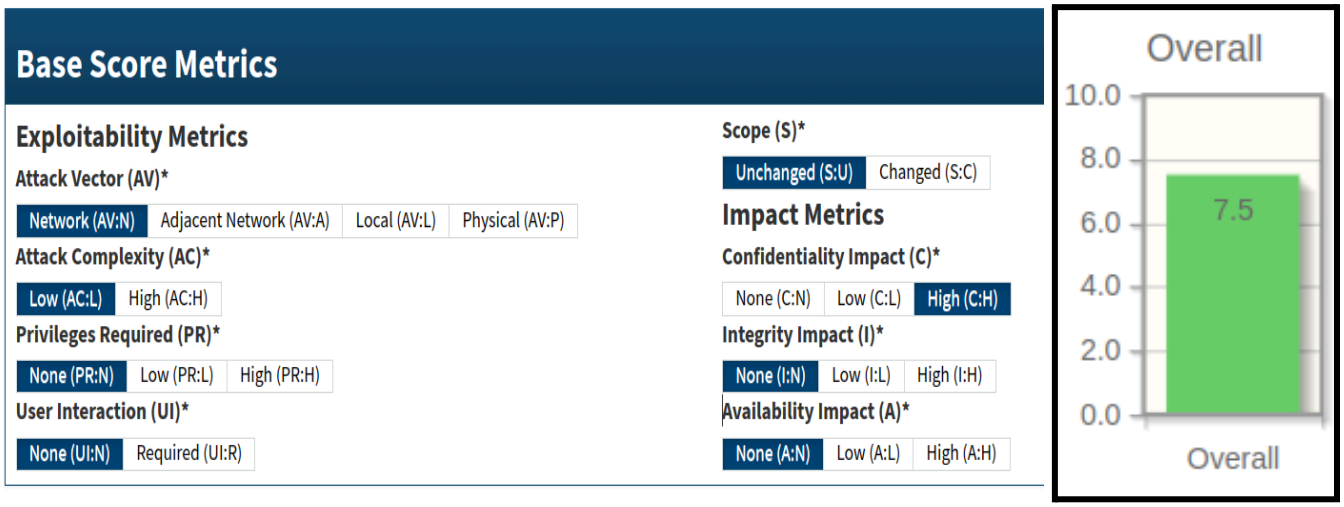


Evidencias

```
[root@localhost ~]# mysql --user=root --password=Chile2010
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
| xy_sistemaz        |
+--------------------+
4 rows in set (0.01 sec)

mysql> USE xy_sistemaz
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-------------------+
| Tables_in_xy_sistemaz |
+-------------------+
| expediente        |
| usuario           |
+-------------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM usuario;
+----+----------+-----------------+-----------+
| id | clave    | nombre          | usuario   |
+----+----------+-----------------+-----------+
|  1 | clave01  | Juan Perez      | juanperez |
|  2 | clave    | Sebastian Gonzalez | sgonzalez |
|  3 | pedro    | Pedro Gomez     | pedro     |
|  4 | Ja7esUJ6 | Victor Gomez    | vgomez    |
|  5 | password | Cesar Lopez     | clopez    |
|  6 | ceH5vux7 | Cristina Vargas | cvargas   |
|  7 | password | Wilson Mora     | wmora     |
|  8 | phU24FrA | Catalina Cardozo | ccardozo  |
|  9 | 1234     | Andres Palacio  | apalacio  |
| 10 | F4ESw3Br | Juliana Vasquez | jvasquez  |
| 11 | 1234     | Felipe Garzon   | fgarzon   |
| 12 | Ra4ehubr | Paola Diaz      | pdiaz     |
| 13 | zAfre8ru | Karl Marx       | kmarx     |
+----+----------+-----------------+-----------+
13 rows in set (0.00 sec)

mysql>
```

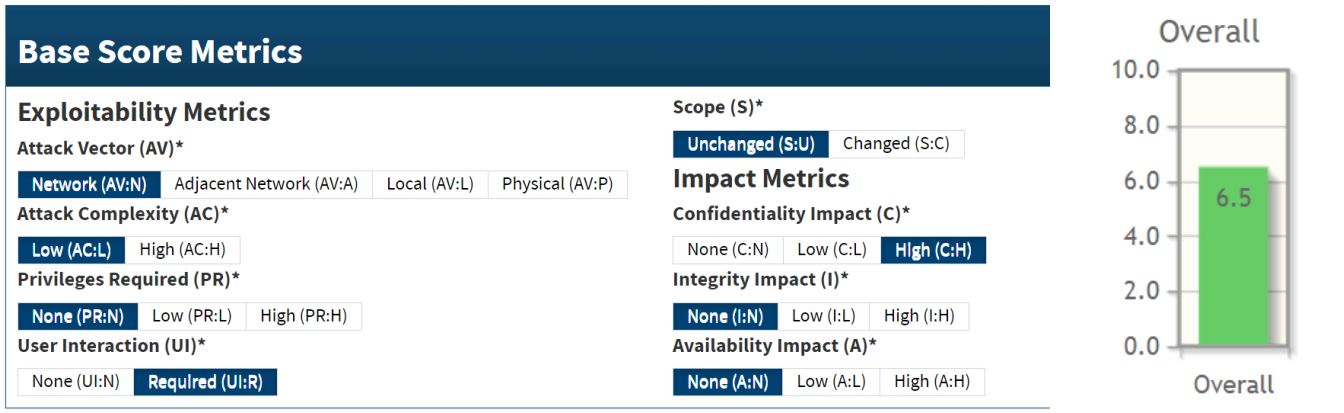
4.5. Vulnerabilidad 5: Expedientes privados expuestos a otros usuarios

- Descripción: Al loguearse al sistema Z como Pedro, se pueden acceder a expedientes de otros usuarios.
- Impacto: Los expedientes contienen información sensible y privada.
- Prueba de Concepto: Accediendo con el usuario user=pedro pass=pedro, podemos acceder a el expediente mediante el numero.
- Riesgo: **Critica**
- Solución: Limitar el acceso de usuarios a expedientes propios.

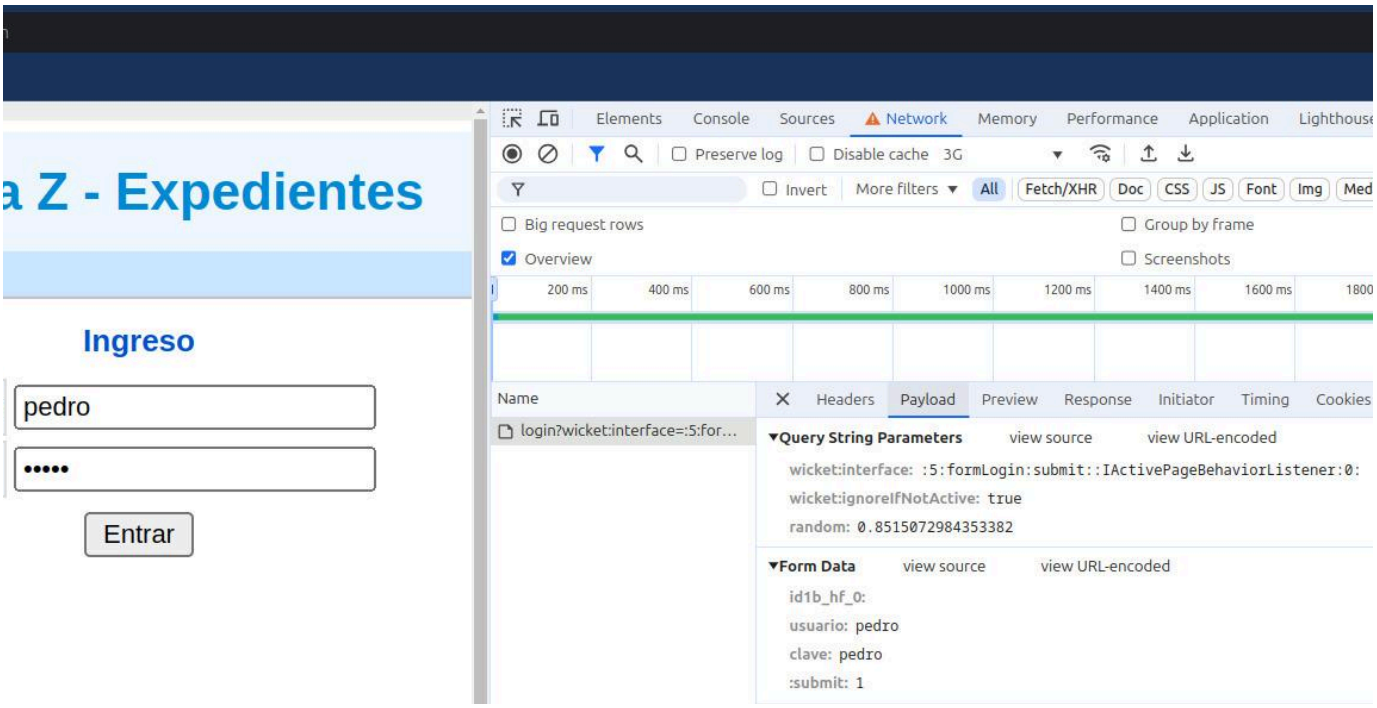


4.6. Vulnerabilidad 6: Datos sensibles expuestos al loguearse

- Descripción: Al loguearse al sistema Z, el POST que se realiza envía en el body de la request el usuario y la contraseña sin encriptar,
- Impacto: Un man in the middle puede interceptar esa request y utilizar el usuario y contraseña para comprometer las cuentas de los usuarios en el sistema violando la confidencialidad de los datos.
- Prueba de concepto: Al usar el inspector del browser se puede ver la request con su payload.
- Riesgo: **Media**
- Solución: Encriptar las contraseñas en el payload.



Evidencias

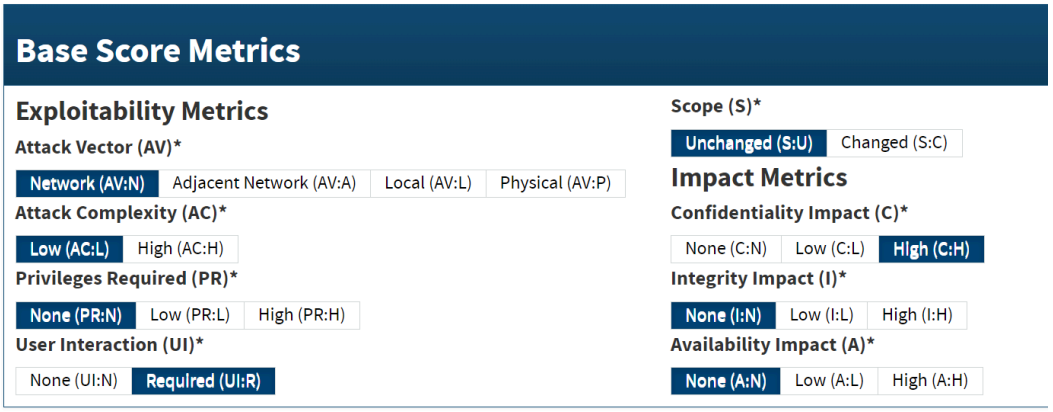


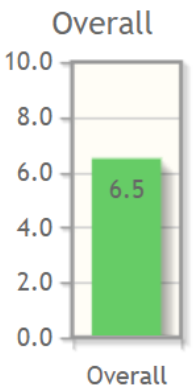
Análisis de Impacto

- Impacto en la confidencialidad: La exposición de contraseñas en texto claro y la posibilidad de acceso no autorizado a MySQL comprometen seriamente la confidencialidad de los datos.
- Impacto en la integridad: El acceso a una cuenta permite que los atacantes modifiquen información sensible del sistema y de la base de datos.
- Impacto en la disponibilidad: El atacante podría potencialmente interrumpir o detener servicios críticos.

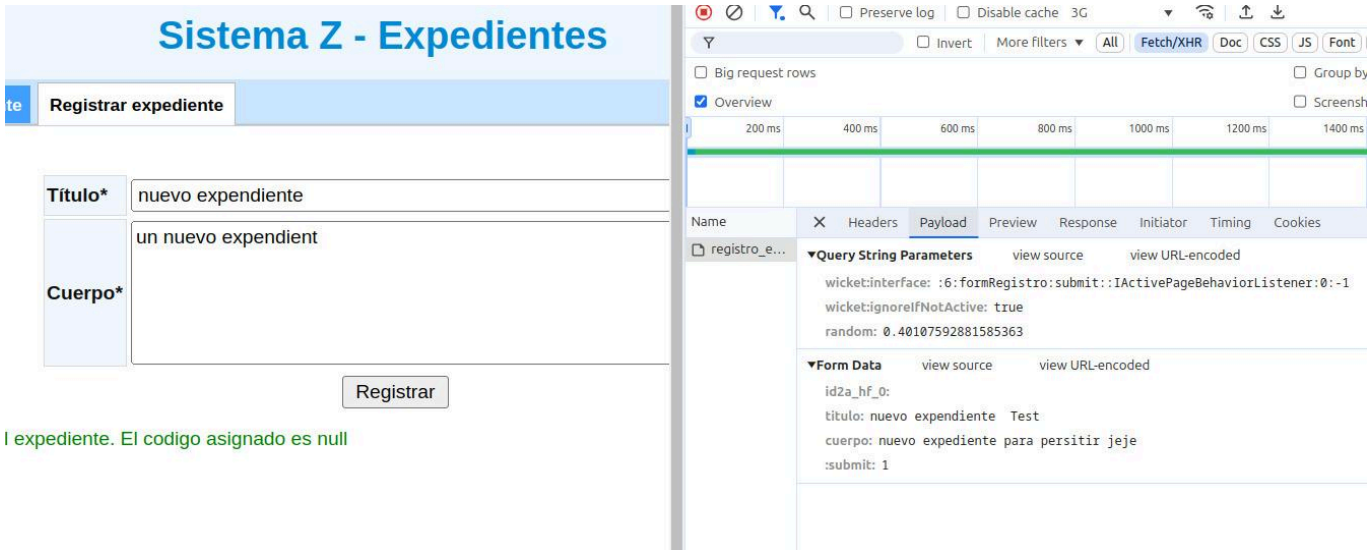
4.7. Vulnerabilidad 7: Datos sensibles expuestos al crear expediente

- Descripción: Al registrar un nuevo expediente en el sistema Z, el POST que se realiza envía en el body del título y la descripción del expediente sin ninguna encriptación,
- Impacto: Un man in the middle puede interceptar esa request y guardar/exponer esos datos sensibles
- Prueba de concepto: Al usar el inspector del browser se puede ver la request con su payload.
- Riesgo: **Media**
- Solución: Encriptar el contenido del expediente al enviarse.





Evidencias



Análisis de Impacto

- Impacto en la confidencialidad: La exposición de datos sensibles como el contenido del título y su cuerpo pone seriamente en riesgo la confidencialidad de los datos.

Recomendaciones

A corto plazo

- Actualizar el software a la última versión disponible.
- Eliminar las credenciales sensibles y configurar el sistema.
- Forzar un cambio de contraseña para todos los usuarios cuyas contraseñas sean débiles.

A mediano plazo

- Implementar políticas de contraseñas fuertes y utilizar algoritmos seguros para el almacenamiento de contraseñas.
- Realizar un análisis de auditoría y control de acceso en la base de datos y los sistemas críticos.

A largo plazo

- Realizar pruebas periódicas de seguridad (pentests anuales o semestrales).
- Implementar un sistema de gestión de identidad y acceso (IAM) que controle de manera centralizada las credenciales y los permisos de los usuarios.

Conclusión

El pentest reveló vulnerabilidades críticas en la infraestructura de SistemaZ, las cuales podrían ser explotadas por atacantes para comprometer la seguridad de los sistemas y la confidencialidad de la información. Se recomienda tomar medidas correctivas de manera inmediata, enfocándose en la actualización de software, la mejora en la gestión de contraseñas, y la implementación de mejores prácticas de seguridad para proteger los datos y la infraestructura a largo plazo.