
# INFORME TÉCNICO

# PENTEST DEL SISTEMA Z

Grupo 8

Integrantes:


| Nombre | Padrón | Mail |
|---|---|---|
| Clara Ruano Frugoli | 106835 | cruano@fi.uba.ar |
| Gabriel Katta | 105935 | gkatta@fi.uba.ar |
| Maria Paula Brück | 107533 | pbruck.ext@fi.uba.ar |
| Paolo Belforte | 109432 | pbelforte@fi.uba.ar |
| Rubén Bohórquez | 109442 | rbohorquez@fi.uba.ar |
| Ramiro Gestoso | 105950 | rgestoso@fi.uba.ar |
| Juan Ignacio Medone Sabatini | 103878 | jmedone@fi.uba.ar |

# Índice

# Introducción

La compañía XY experimentó recientemente la filtración pública de un expediente confidencial, evento que evidenció debilidades críticas en su esquema de protección de datos y expuso riesgos legales significativos. Ante este incidente, la dirección decidió emprender acciones inmediatas para reforzar la seguridad de su plataforma de gestión de expedientes en línea, conocida como Sistema Z.

Con el fin de conocer de manera objetiva el estado real de su ciberseguridad, XY nos solicitó la realización de un Penetration Test de tipo *Greybox* sobre la aplicación y su infraestructura de soporte. Esta modalidad, basada en un conocimiento parcial del entorno (acceso limitado a credenciales y documentación), replica el escenario de un atacante con información restringida pero verosímil, permitiendo descubrir vectores de ataque que habitualmente pasan inadvertidos en pruebas *Blackbox* o revisiones exclusivamente de código.

Los objetivos específicos del ejercicio fueron:

1. Identificar y validar vulnerabilidades técnicas que puedan comprometer la confidencialidad, integridad y disponibilidad de la información contenida en el Sistema Z y en los servidores subyacentes.
2. Evaluar la eficacia de los controles existentes frente a tácticas comunes de intrusión, escalación de privilegios y exfiltración de datos.
3. Medir el riesgo operativo que cada hallazgo representa para el negocio y su cumplimiento normativo.
4. Proporcionar recomendaciones priorizadas que permitan mitigar las debilidades detectadas y establecer un plan de mejoras a corto, mediano y largo plazo.

Nuestro objetivo general es dotar a la compañía XY de una visión integral y accionable que permita no solo remediar las debilidades actuales, sino también fortalecer su resiliencia ante futuros intentos de ataque para asegurar el cumplimiento de los tres pilares de la seguridad de su información.

# Metodología de trabajo

Se llevó a cabo una evaluación de tipo *Greybox*, combinando técnicas manuales y automatizadas, en una red de laboratorio que replicaba el entorno productivo del Sistema Z. Las pruebas incluyeron reconocimiento de superficie de ataque, validación de controles de autenticación, evaluación de servicios de red y explotación de vulnerabilidades encontradas.

El trabajo se estructuró en dos fases principales:

### Fase 1 – Análisis y diagnóstico

Se realizó una revisión manual y automatizada de la superficie de ataque, identificando servicios expuestos, configuraciones inseguras, flujos de autenticación defectuosos y vulnerabilidades conocidas. Esta fase tuvo como objetivo comprender la arquitectura, registrar hallazgos preliminares y validar fallos sin alteración del sistema.

### Fase 2 – Pruebas de explotación controladas

Se ejecutaron ataques controlados sobre vulnerabilidades críticas previamente identificadas, con el fin de evaluar el impacto real y mostrar los recursos comprometidos. Esto incluyó la explotación de Ghostcat, Axis2 y la escalada de privilegios a root, obteniendo escenarios de compromiso total del servidor.

# Herramientas utilizadas

### Entorno de pruebas

- **VM Kali Linux** – Máquina atacante con herramientas preinstaladas.
- **VM CentOS** – Servidor objetivo con el Sistema Z.

### Navegación y análisis manual

- **Firefox Browser** – Acceso y navegación por el sistema.
- **Thunder Client / REST Client** – Simulación de peticiones HTTP (login, creación/búsqueda de expedientes).

### Escaneo y enumeración

- **Nmap** – Detección de puertos abiertos y servicios expuestos.
- **db_nmap** – Escaneo integrado con Metasploit.
- **Greenbone OpenVAS** – Escaneo automatizado de vulnerabilidades.

### Ataques específicos y bases de datos

- **Zaproxy** – Análisis del tráfico, testing de vulnerabilidades web, fuzzing de parámetros.
- **SQLMap** – Pruebas de inyección SQL.
- **MySQL client** – Interacción directa con bases de datos.

- **SSH client** – Conexión directa a servidores remotos.

### Explotación

- **Metasploit Framework (msfconsole)** – Ejecución de exploits y módulos auxiliares.
- **sshpass + rockyou.txt** – Fuerza bruta contra SSH usando diccionario.
- **AJP Shooter** – Explotación de la vulnerabilidad Ghostcat (AJP/Tomcat).

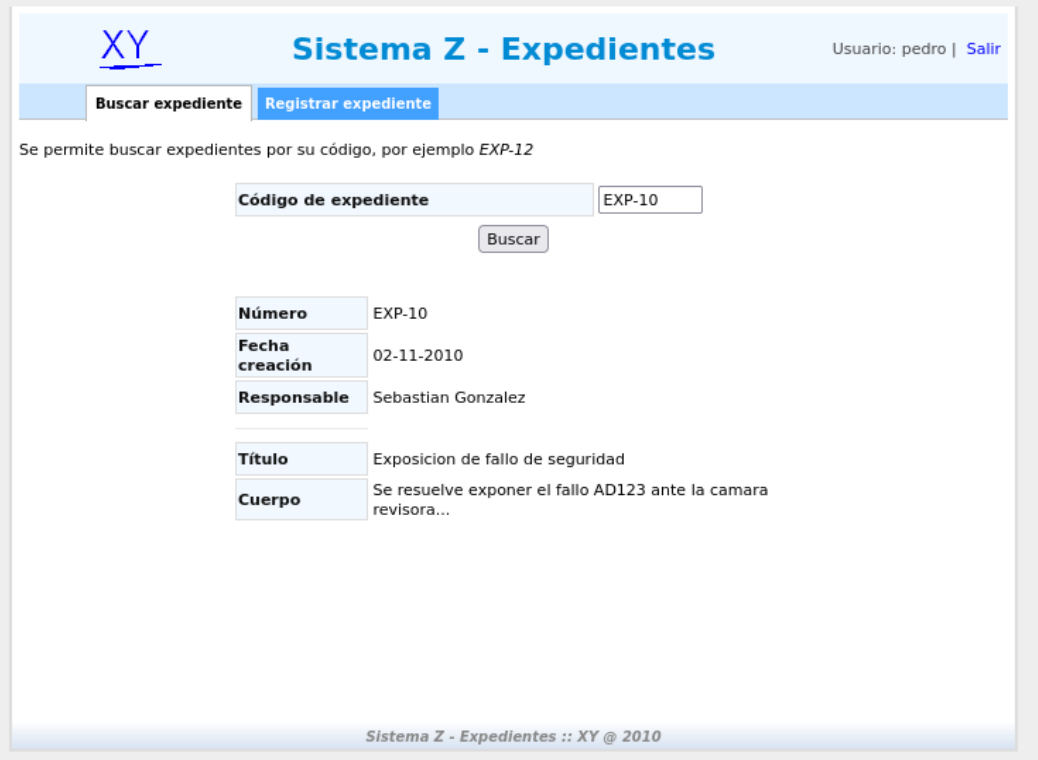
# Detalle de vulnerabilidades

### Sistema Z Login

- **No hay protección contra ataques de fuerza bruta:** no existe ningún mecanismo de bloqueo tras múltiples intentos fallidos (por ejemplo bloqueo por IP, rate limiting, captchas o delays). Esto permite probar combinaciones de usuario/clave de forma ilimitada, facilitando ataques de fuerza bruta manuales o automatizados.
- **No se invalida la sesión al ingresar a /login ni redirige a la home page:** cuando un usuario previamente autenticado navega a /login, el sistema muestra la pantalla de login sin invalidar la sesión activa ni redirigir a la página principal. Esto genera confusión y riesgo, pues puede parecer que el usuario cerró sesión cuando en realidad sigue autenticado, abriendo la puerta a accesos no autorizados.
- **Las credenciales de testeo (“pedro - pedro” y demás) no fueron eliminadas:** ante una eventual filtración de la documentación del sistema, un agente malicioso puede hacerse de estas credenciales de prueba e ingresar.
- **Cookies de sesión mal configuradas:**
    - Cookies de sesión con *httpOnly* en false.
    - Cookies de sesión con *secure* en false. Esto hace que el sistema sea susceptible a ataques XSS y man in the middle.
    - Cookies de sesión con *samesite* en none. Esto significa que la cookie puede provenir de peticiones desde otros sitios, lo que convierte al sistema susceptible a ataques CSRF.
    - El sistema permite la reutilización de tokens de sesión antiguos,.

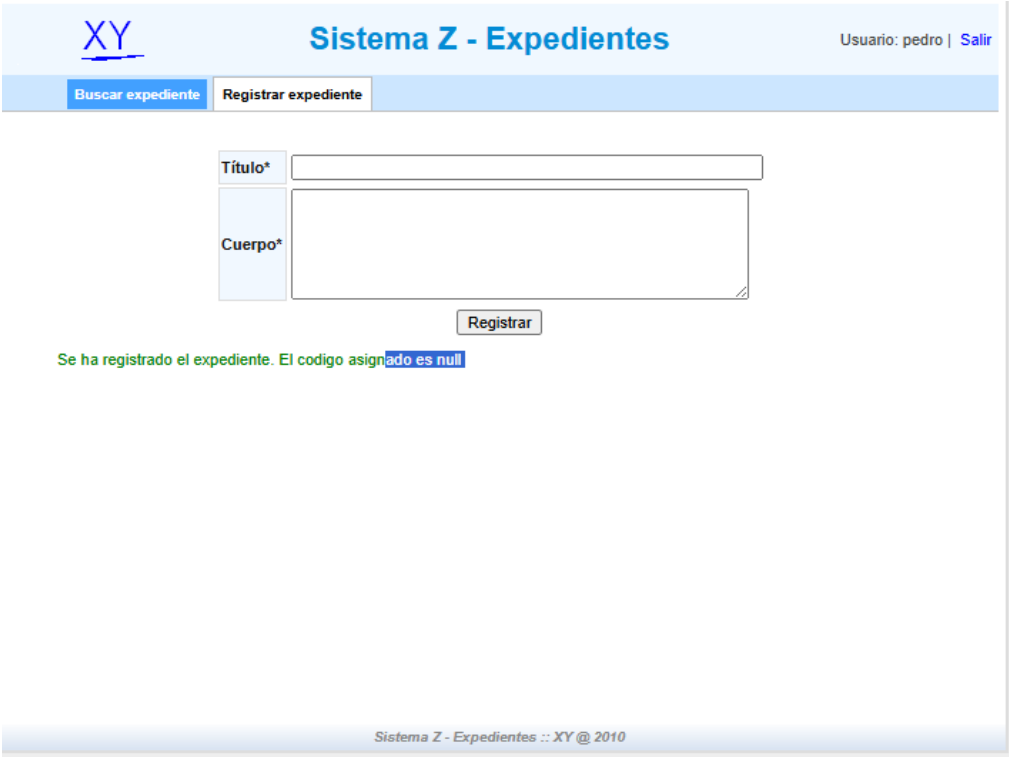
## Sistema Z Expedientes

- **Cualquier usuario puede acceder a los expedientes de cualquier otro:** cualquier usuario puede consultar expedientes que no le pertenecen, rompiendo la confidencialidad de los mismos.




*Evidencia de un usuario accediendo a un expediente de otro usuario.*

- **Crear expedientes con información inválida mediante un cliente REST puede generar expedientes sin código (*null*) y corromper los siguientes intentos:** mediante peticiones POST sin cookie de sesión, es posible crear expedientes sin cuerpo ni título, asignándoles código *null*. Esto corrompe la base de datos y afecta la creación correcta de futuros expedientes.




*Evidencia de creación de expedientes con información inválida.*

- **Búsqueda y creación de expedientes sin cookie de sesión:** grave falla del control de acceso.

## Sistema Z general

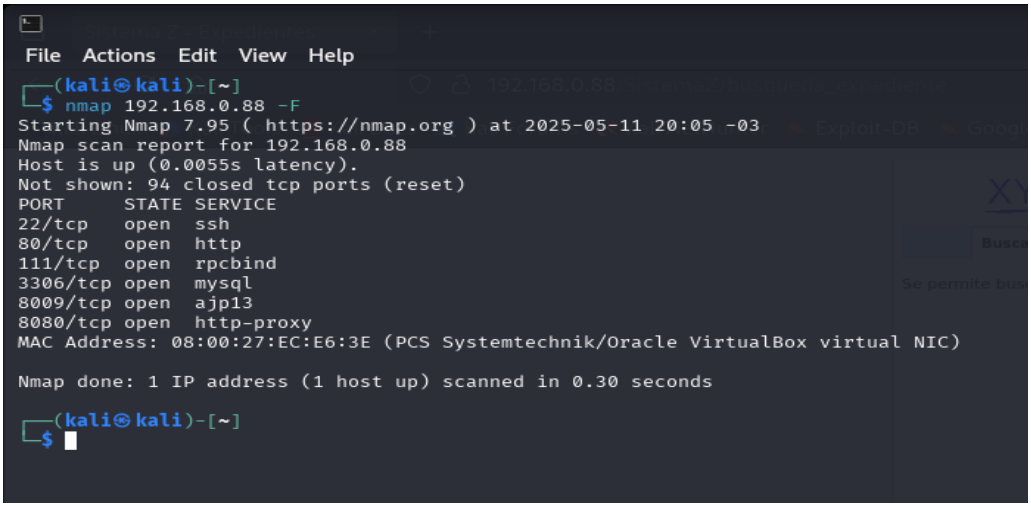
- **No hay código de error claro ante timeouts:** falta respuesta adecuada para errores de timeout, causando confusión en la aplicación cliente.
- **Susceptible a ataques CSRF:** no existen tokens anti-CSRF o mecanismos equivalentes, aumentando el riesgo de acciones no autorizadas por usuarios autenticados.
- **Uso de HTTP en lugar de HTTPS:** toda la comunicación no está cifrada, exponiendo credenciales y datos sensibles a interceptaciones.
- **Vulnerabilidad de Apache Axis2:** con msfconsole se puede realizar el exploit “auxiliary/scanner/http/axis_login”.

## Servidor (red y servicios)

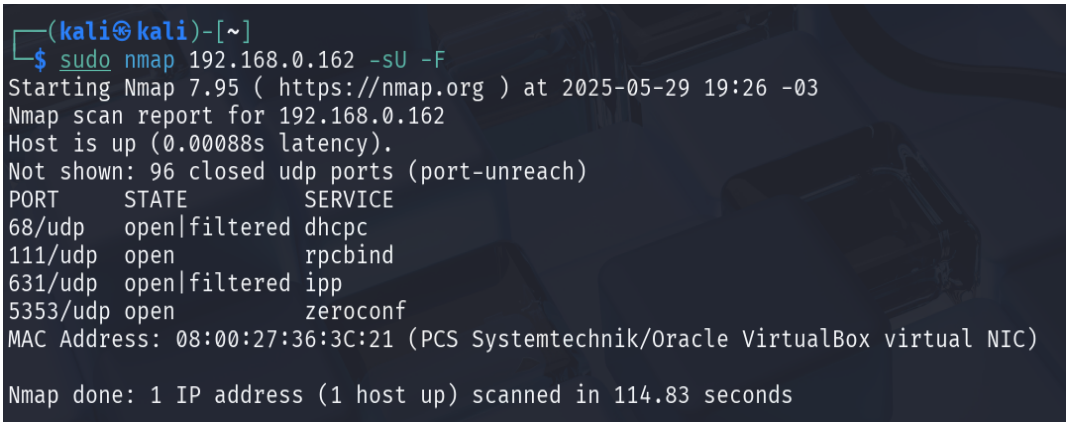
### Puertos abiertos a la red

- 22 (SSH).
- 3306 (MySQL).
- 111 (RPCBind) y 8009 (AJP13), ambos vulnerables.

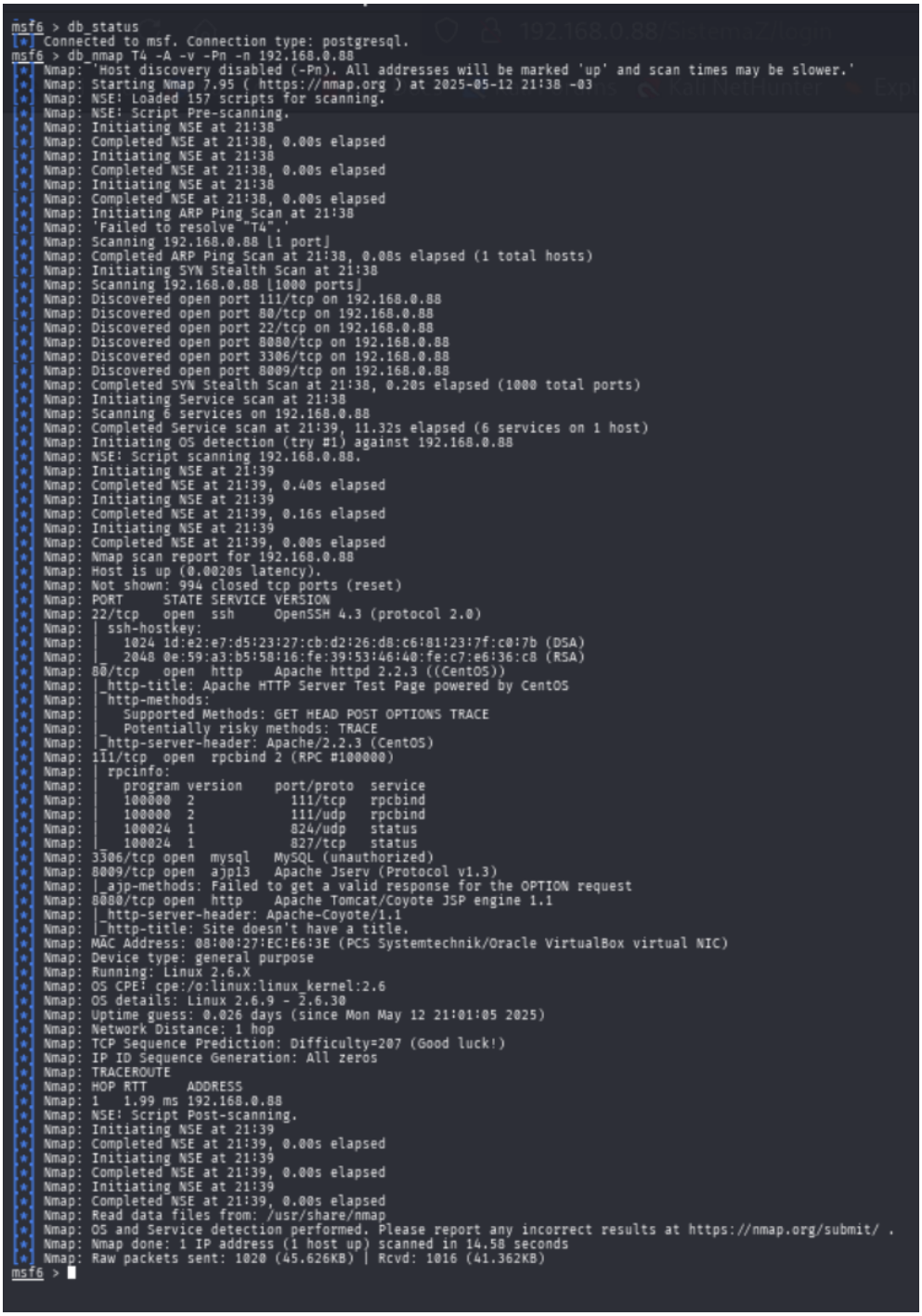
- UDP: 68 (DHCPC), 111 (RPCBind), 631 (IPP), 5353 (Zeroconf).




*Resultado del escaneo de puertos TCP con nmap.*

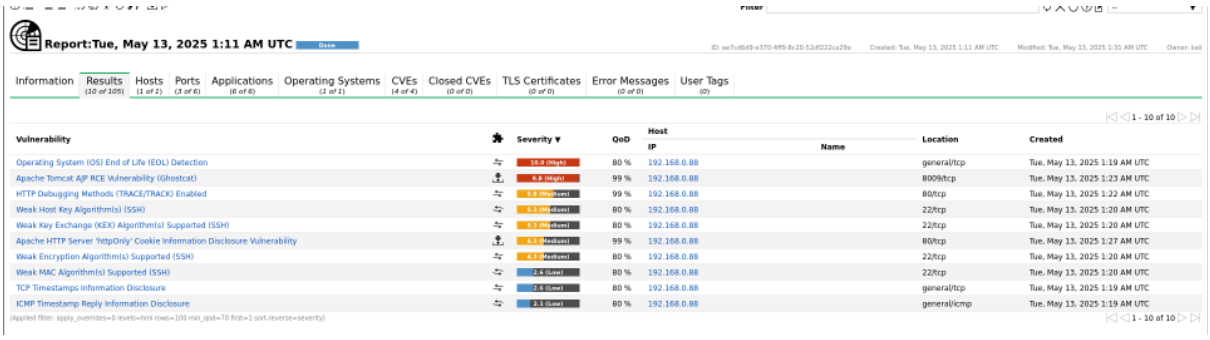



*Resultado del escaneo de puertos UDP con nmap.*




*Resultado del escaneo de puertos con db_nmap dentro de* ***metasploit****.*

## Vulnerabilidades encontradas con Greenbone/OpenVAS




*Resultado del escaneo realizado con Greenbone/OpenVAS.*

- **OS End of Life: CentOS 5 (sin soporte ni parches) (Severidad 10.0 - Crítica):**
    - El sistema operativo identificado es CentOS 5, el cual alcanzó su fin de soporte el 31 de marzo de 2017. Esto implica que no recibe actualizaciones de seguridad, lo que lo expone a múltiples vulnerabilidades conocidas y explotables. Su uso incumple normativas como PCI-DSS o ISO 27001, y representa un riesgo grave para la integridad y confidencialidad de la información. Se recomienda migrar a una versión actualizada de inmediato.
- **Ghostcat: vulnerabilidad crítica en Apache Tomcat AJP (CVE-2020-1938) (Severidad 9.8 - Crítica):**
    - Se detectó una vulnerabilidad crítica en el conector AJP del servidor Apache Tomcat (puerto 8009/tcp), conocida como Ghostcat. Permite a un atacante leer o incluir archivos internos del servidor, afectando versiones anteriores a 7.0.100, 8.5.51 y 9.0.31. Esta vulnerabilidad podría ser usada para robar credenciales o ejecutar código malicioso.
- **Métodos TRACE/TRACK habilitados (Severidad 5.8 - Media):**
    - El servidor web en el puerto 80 permite métodos HTTP TRACE y TRACK, comúnmente usados para depuración. Esta configuración puede facilitar ataques de Cross-Site Tracing (XST), permitiendo el robo de cookies o tokens de autenticación si se combina con otras vulnerabilidades como XSS. Es recomendable deshabilitarlos.
- **Exposición de cookies *HttpOnly* en respuestas HTTP 400 sin ErrorDocument personalizado (Severidad 4.3 - Media):**
    - Se detectó que el servidor responde a errores HTTP 400 sin un ErrorDocument personalizado, lo cual puede provocar la exposición de cookies marcadas como *HttpOnly*. Esto representa un riesgo de

secuestro de sesión si un atacante induce al usuario a acceder a URLs malformadas.

- **Algoritmos de clave SSH débiles - DSA (Severidad 5.3 - Media):**
    - El servidor SSH permite el uso del algoritmo ssh-dss (DSA), considerado obsoleto y débil. Esta configuración puede facilitar ataques de fuerza bruta o colisiones criptográficas, comprometiendo la autenticación SSH.
- **KEX inseguros en SSH (Severidad 5.3 - Media):**
    - Se detectó que el servidor SSH utiliza algoritmos de intercambio de claves débiles como diffie-hellman-group-exchange-sha1, los cuales emplean SHA-1 y grupos primos pequeños. Esto los hace vulnerables a ataques de tipo Logjam, especialmente por actores con capacidades avanzadas.
- **Cifrados débiles en SSH (Severidad 4.3 - Media):**
    - El servidor SSH admite cifrados inseguros como CBC (aes128-cbc, 3des-cbc), RC4 (arcfour), y algoritmos obsoletos como blowfish-cbc. Estas opciones facilitan ataques de padding oracle y ruptura criptográfica, comprometiendo la confidencialidad de las comunicaciones.
- **MACs débiles en SSH (Severidad 2.6 - Baja):**
    - También se observó que el servidor permite algoritmos MAC como hmac-md5 y hmac-sha1-96, considerados débiles. Esto representa un riesgo bajo, pero real, para la integridad de las comunicaciones si un atacante intercepta tráfico cifrado.
- **Timestamps TCP y respuesta ICMP Timestamp exponiendo info sensible (Severidad 2.6 - Baja):**
    - Se detectó que el servidor responde con timestamps TCP, lo que permite a un atacante inferir el tiempo de actividad del sistema. Esta información puede ser usada para afinar ataques como secuestro de sesiones o predicción de secuencias TCP.
    - El host responde a solicitudes ICMP tipo timestamp, revelando la hora interna del sistema. Aunque de bajo impacto, esta información puede combinarse con otros vectores para comprometer mecanismos que dependen del tiempo, como generadores de números aleatorios.

## Otras vulnerabilidades

- **Acceso root sin restricciones físicas para cambiar contraseña.**
- **Denial of Service (DoS) por carga alta en login y búsqueda expedientes:**
    - El sistema se cuelga si se introducen contraseñas repetidamente o si se realizan múltiples búsquedas (incluso del mismo expediente). Esta

condición permite ataques de denegación de servicio que afectan la disponibilidad del sistema. Se recomienda implementar limitadores de tasa o caché en backend.

### Explotación controlada de vulnerabilidades

Durante la fase de explotación se validaron de manera práctica dos vulnerabilidades críticas previamente identificadas:

- **Ghostcat (CVE-2020-1938):** se utilizó la herramienta *AJPShooter* para explotar el conector AJP en el puerto 8009. A través de esta falla, fue posible acceder a archivos internos del servidor Tomcat, incluyendo clases Java de la aplicación. Esto permitió identificar estructuras internas del sistema y localizar un backend expuesto (/WS-Expedientes) basado en Axis2.

- **Ejecución remota de código en Axis2:** tras identificar que el backend utilizaba Axis2, se extrajeron configuraciones sensibles y credenciales del archivo `axis2.xml`. Se aprovechó luego un exploit público (`axis2_deployer`) para obtener acceso remoto al servidor. A partir de esa sesión, se escaló en privilegios a root utilizando una vulnerabilidad local (`glibc_origin_expansion_priv_esc`) presente en el sistema operativo desactualizado (CentOS 5).

Estas pruebas demuestran que un atacante podría tomar control total del servidor combinando errores de configuración, software obsoleto y falta de segmentación entre servicios expuestos.

El proceso detallado de la explotación de vulnerabilidades se encuentra en el anexo.

# Conclusión

El pentest al Sistema Z evidenció fallas que comprometen seriamente su seguridad y demandan acciones inmediatas.
A partir del análisis de login, módulos de expedientes, servicios expuestos y configuraciones de red se identificaron tres áreas clave de riesgo:

## Vulnerabilidades críticas

- **Acceso remoto no autenticado:** Ghostcat (AJP 8009), Axis2 y puerto MySQL 3306 abierto.
- **Explotación de credenciales:** usuarios de prueba activos y contraseñas enviadas en texto plano.
- **Debilidades de infraestructura:** sistema operativo EOL, cifrados/algoritmos SSH débiles y ausencia de TLS.

## Configuraciones incorrectas

- **Ausencia de rate-limiting, bloqueo por IP y timeouts en login y SSH:** permite ataques de fuerza bruta sin restricciones.
- **Gestión de sesión insegura:** cookies sin *Secure*/*HttpOnly*, reutilización de tokens, lo cual permiten ataques de tipo CSRF.
- **API de expedientes sin controles:** lectura y creación sin cookie de sesión, posibilidad de registros corruptos y DoS lógico.

## Conciencia de seguridad insuficiente

- **Persistencia de credenciales por defecto y mensajes de error genéricos:** ocultan potenciales intrusiones o fallos graves del sistema.
- **Carencia de guías de hardening y formación:** se ve reflejada en configuraciones por omisión y puertos innecesariamente expuestos.

Corregir estas deficiencias críticas, mejorar las configuraciones y reforzar la capacitación del personal reducirá significativamente el riesgo operativo y mejorará la postura de seguridad global del Sistema Z.

# Anexo

En este documento se presenta a detalle la explotación de las vulnerabilidades OS-EOL y Ghostcat con capturas de pantalla del pentest realizado, del cual se habló previamente en la sección Explotación controlada de vulnerabilidades.