Graduado en Ingeniería Informática

Sistemas de Información

Auditoría de Seguridad en OFIAUTO

Security Audit in OFIAUTO

Realizado por

José Sánchez-Rosso Almoguera

Tutorizado por

David Santo Orcero

Co-tutorizado por

Departamento

Departamento de Lenguajes y Ciencias de la Computación

Málaga, November 19, 2025

UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADUADO EN INGENIERÍA INFORMÁTICA

Sistemas de Información

Auditoría de Seguridad en OFIAUTO

Security Audit in OFIAUTO

Realizado por

José Sánchez-Rosso Almoguera

Tutorizado por

David Santo Orcero

Co-tutorizado por

Departamento

Departamento de Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, NOVEMBER 19, 2025

Fecha defensa: diciembre de 2025

Resumen

En la era digital, la seguridad de la información es un factor determinante para la protección de los activos empresariales y la confianza de los usuarios. Las auditorías de seguridad posibilitan a las empresas analizar la efectividad de las medidas de protección, identificar vulnerabilidades y mitigar riesgos antes de que sean explotados por actores malintencionados.

Este Trabajo de Fin de Grado se centra en la auditoría de seguridad aplicada a la infraestructura informática de una empresa, analizando los principales estándares y metodologías utilizadas en el sector, como ISO 27001 y NIST . Se estudiaran las herramientas de auditoría más utilizadas en el sector, como Lynis, OWASP ZAP y Legion entre ellos. Estas se aplicarán a la evaluación de redes, sistemas operativos y aplicaciones web de la empresa.

Para la parte práctica, se lleva a cabo una auditoría en el entorno de la empresa, siguiendo una metodología se adoptará la metodología estructurada y secuencial propuesta por David Santo Orcero en su libro "Pentesting con Kali" . A partir de los resultados obtenidos, se identifican los principales riesgos en los sistemas evaluados y se presentan soluciones para fortalecer la seguridad de la organización.

Los hallazgos demostraran la importancia de realizar auditorías periódicas para garantizar la protección de la información y el cumplimiento de las normativas vigentes. Como conclusión, se proponen estrategias para mejorar la seguridad en la empresa mediante la formación al personal de controles efectivos y buenas prácticas en la gestión de riesgos. **Palabras clave: Auditoría de seguridad, Infraestructura informática, Evaluación de riesgos, Ciberseguridad, Metodologías de pentesting**

Abstract

In the digital age, information security is a determining factor for the protection of business assets and user trust. Security audits enable companies to analyze the effectiveness of protection measures, identify vulnerabilities, and mitigate risks before they are exploited by malicious actors.

This Bachelor's Thesis focuses on security auditing applied to a company's IT infrastructure, analyzing the main standards and methodologies used in the sector, such as ISO 27001 and NIST. The most widely used audit tools in the sector will be studied, including OSINT, Wireshark, OWASP, and Legion, among others. These will be applied to the evaluation of the company's networks, operating systems, and web applications.

For the practical part, an audit will be conducted within the company environment, following the structured and sequential methodology proposed by David Santo Orcero in his book "Pentesting with Kali." Based on the results obtained, the main risks in the evaluated systems will be identified, and solutions will be presented to strengthen the organization's security.

The findings will demonstrate the importance of conducting periodic audits to ensure the protection of information and compliance with current regulations. As a conclusion, strategies will be proposed to improve security within the company through staff training on effective controls and best practices in risk management. **Keywords:Security audit, IT infrastructure, Risk assessment, Cybersecurity, Pentesting methodologies**

Agradecimientos

Bueno esto ha llegado a su fin, quería dar las gracias a toda mi familia por el cariño que me han dado durante esta etapa. No ha sido fácil, pero vuestros ánimos, comprensión y atención ha sido vital. A mis amigos que me siempre me han dado los mejores consejos, ayudándome y siendo un refugio donde poder despejarme y sentirme querido. Por último a Angelo contigo entendí, que es ser ingeniero aprender de ello, descubrir un mundo que me apasionaba y me sigue entusiasmando, en lo que me convertiría hoy en día no hubiera sido posible sino me lo hubieras enseñado. Muchas gracias de corazón. Por último cuanto me queda por aprender, esto es solo el comienzo. Quien diría hace 4 años cuando empecé que tendría este transcurso...

Índice

1 Introducción

1.0.1 Introducción

En el contexto digital actual, donde la información ha representado prácticamente uno de los activos más valiosos para empresas y organizaciones, la seguridad informática se ha convertido en un componente esencial y crítico. La constante evolución de las amenazas, junto con los ataques informáticos, ha contribuido a que las empresas hayan tenido que desarrollar estrategias y mecanismos de defensa para la prevención de estos ataques. Como consecuencia de lo anterior, la auditoría de seguridad se ha afianzado como una práctica indispensable para identificar, evaluar y mitigar dichas vulnerabilidades antes de que puedan ser explotadas por *hackers* de forma maliciosa.

Una auditoría de seguridad ha consistido en analizar el sistema de redes, aplicaciones y sistemas con el objetivo de detectar fallos de configuración, brechas de seguridad o prácticas deficientes que hayan puesto en riesgo la confidencialidad, integridad o disponibilidad de los activos tecnológicos de la empresa. Por ello, se han desarrollado metodologías de forma estructurada y herramientas específicas que nos han permitido anticiparnos a estas potenciales amenazas, determinar el grado de exposición al que se ha expuesto una organización y cómo poder reducirlo mediante mejoras tangibles en el ámbito defensivo de dicha organización o empresa.

1.0.2 Motivación

Este Trabajo de Fin de Grado ha nacido de la necesidad de comprender y aplicar, en un entorno realista, las fases de las que se ha compuesto una auditoría de seguridad, que ha comprendido desde la recogida de información hasta la presentación de resultados al personal técnico y la formación. Nuestra motivación se ha centrado en evidenciar la importancia que han conllevado estas auditorías en las organizaciones para garantizar la invulnerabilidad en cualquier infraestructura tecnológica.

1.0.3 Objetivos

Con este Trabajo de Fin de Grado (TFG) hemos perseguido fortalecer la seguridad en entornos empresariales mediante un análisis exhaustivo de su infraestructura tecnológica. Hemos llevado a cabo una auditoría de seguridad en una empresa real, examinando sus redes, sistemas y políticas de seguridad para identificar deficiencias y sugerir mejoras basadas en estándares

reconocidos y buenas prácticas del sector. Los objetivos han sido los siguientes:

- Analizar la infraestructura tecnológica de la empresa.
- Detectar las vulnerabilidades de la empresa.
- Evaluar los riesgos asociados y proponer medidas de mitigación.
- Documentar las evidencias siguiendo estándares profesionales.

1.0.4 Metodología

La metodología que hemos empleado en este Trabajo de Fin de Grado (TFG) se ha fundamentado en un enfoque estructurado y secuencial propuesto por David Santo Orcero en su libro "Pentesting con Kali" para llevar a cabo una auditoría de seguridad informática, específicamente mediante pruebas de penetración (pentesting). Este enfoque ha combinado estándares reconocidos internacionalmente con prácticas del hacking ético, adaptadas al entorno específico de la empresa auditada. [116] Se ha compuesto de las siguientes fases:

- **1.** Acuerdo de los objetivos
- **2.** Alcance y condiciones del test de intrusión
- **3.** Recolección inicial de permisos y autorizaciones
- **4.** Validación legal y de permisos del alcance y condiciones del test de intrusión
- **5.** Recolección de información
- **6.** Análisis de las vulnerabilidades
- **7.** Redacción del informe de auditoría y presentación de resultados a la propiedad
- **8.** Presentación de resultados al personal técnico y formación

1.0.5 Estructura del documento

Hemos organizado el documento de la siguiente manera:

- **Capítulo 1:** Introducción – hemos contextualizado la problemática, los objetivos y la estructura del trabajo.
- **Capítulo 2:** Estado del arte – hemos expuesto los conceptos clave sobre auditoría, el contexto histórico de la ciberseguridad y las normativas aplicables.
- **Capítulo 3:** Proceso de consentimiento y desarrollo del formulario de autorización – documento donde hemos establecido el marco legal, las herramientas que hemos utilizado y su impacto en la empresa.
- **Capítulo 4:** Herramientas utilizadas – material con el que hemos auditado.
- **Capítulos 5,6,7 y 8:** Resultados de la auditoría – hallazgos, vulnerabilidades detectadas y recomendaciones.
- **Capítulo 9:** Conclusiones – resumen de los hallazgos y recomendaciones finales.

2

Estado del Arte

2.0.1 Definición

El test de intrusión es el proceso para identificar vulnerabilidades de seguridad en una aplicación evaluando el sistema o la red con diversas técnicas maliciosas. Los puntos débiles de un sistema se explotan en este proceso a través de un ataque simulado autorizado.Esto Una prueba de penetración indica si las medidas defensivas existentes empleadas en el sistema son lo suficientemente fuertes como para evitar cualquier violación de seguridad. Los informes de las pruebas de penetración también sugieren las contramedidas que se pueden tomar para reducir el riesgo de piratería del sistema.[96, 116]

El propósito de esta prueba es asegurar datos importantes de personas externas como hackers que pueden tener acceso no autorizado al sistema. Una vez que se identifica la vulnerabilidad, se utiliza para explotar el sistema para obtener acceso a información confidencial.

Una prueba de penetración también se conoce como prueba de la pluma y un probador de penetración también se conoce como un hacker ético.

2.0.2 Evolución Historica

2.0.2.1 Años 1960 a 1970

El hacking ético como se conoce hoy en día deriva del hacking. En la década de los años 60, los primeros denominados *hackers* eran principalmente estudiantes de la institucion MIT(Massachusetts Institute of Technology) que se interesaban por el mundo de la tecnología, con ello buscaban obtener un mejor rendimiento y conocimiento de las máquinas de la época. Este concepto produjo un alto impacto por lo que en 1969 se crearía ARPANET, predecesor del internet esto conllevó a que los universitarios accedieran a sistemas remotos de los que no tenían permiso simplemente por curiosidad más que buscando hacer daño a los sistemas.Por lo que produjo una preocupación viendo el alto riesgo que tenían estas actividades.[110, 113, 150]

2.0.2.2 Años 1970 a 1980

Una decada más tarde en los años 70, se comienzan a obtener los primeros ejemplos de hacking, conocido como "phone phreaking" donde se buscaba realizar llamadas gratuitas mediante la explotación de debilidades del sistema telefónico, ya que en esa época eran de alto coste las llamadas de larga distancia. De la mano de John Draper, precursor de está técnica se dió

cuenta que un silbato emitía el mismo tono que las llamadas así surgió el "phone phreaking".[110, 150]

2.0.2.3 Años 1980 a 1990

A principios de la década de los 80, comienza la Edad de Oro del hacking debido a que la venta de ordenadores personales eran mas accesible, estos ordenadores se comunicaban mediante la red telefónica.A consecuencia de ello, se da una nueva influencia de grupos de hackers que se dedicaban a poner en práctica sus conocimientos para delinquir, pirateando software, creando virus y hackeando con la finalidad de robar información que con ella podrían defraudar a empresas o vender dicha información.Se estrenó en 1983 la conocida película 'War Games', que trata un adolescente que se adentra en el ordenador militar y casi provoca la Tercera Guerra Mundial.A causa de ello un grupo de adolescentes de Milwakee denominado 414, comienza a burlar por diversión los sistemas de red de diversas instituciones de alto perfil como el Centro de Cáncer Memorial Slon Kettering y el Laboratorio Nacional de Los Alamos, un centro de investigación de bombas nucleares que produjeron bombas atómicas utilizadas en la Segunda Guerra Mundial. Debido al alto revuelo de esta noticia Estados Unidos crea la primera ley relacionada con la piratería, la Ley de Abuso y Fraude Informático, en 1986, aunque no frenó la expansión de la cultura del hacking. [110, 150]

2.0.2.4 Años 1990 a 2000

Los años 90 se centra, en los conocidos crackers(hackers maliciosos), con el acceso de intenet extendido a todo el mundo, se incrementan de forma desproporcionada los delitos informáticos por lo que se produjeron delitos a gran escala como robar software a grandes empresas, fraude a una estación de radio con la finalidad de ganar un coche a sorteo y el primer robo digital a una institución financiera.Cinco años mas tarde en 1995 se desarrolla de la mano de Dan Farmer y Wietse Venema SATAN (Security Administrator Tool for Analyzing Networks), un escáner automático de vulnerabilidades, que se hace un nombre como herramienta de hacking. Ese mismo año, se denomina por primera vez el término "hacking ético" fue acuñado por el vicepresidente de IBM.A finales de la década de los 90, en 1999 la seguridad del software toma más importancia aún con el lanzamiento de Windows 98 de Microsoft por lo que se convierte en un año excepcional para la seguridad y la piratería.[150, 56]

2.0.2.5 Años 2000 al 2010

Los años 2000 destacan por varios aspectos, se formaliza el tecnicismo "pentesting" donde se pasa de ver desde un punto de vista de una práctica informal a ser un servicio profesional estandarizado.Por otro lado se popularizan herramientas como NESSUS que se trataba de un escáner de vulnerabilidades que es el relevo de SATAN.Finalmente se publicó en esta década , una norma internacional que proporcionó un código de buenas prácticas para la gestión de la seguridad de la información, incluyendo directrices sobre la realización de pruebas de seguridad y auditorías técnica por lo que influyó significativamente al avance de los test de intrusión.[56, 17]

2.0.2.6 Años 2010 a la actualidad

A partir de 2010, se centran en la automatización de pruebas mediante la necesidad de realizar evaluaciones más frecuentes que llevó al desarrollo de herramientas automatizadas que permitían pruebas más rápidas y eficientes. Además de regulaciones como el GDPR y la ISO 27001 requerían de pruebas de seguridad periódicas, consolidando el pentesting como una

práctica estándar para muchas organizaciones.Sin embargo estas pruebas producen que se cree el servicio (Pentesting como servicio) donde se ofrecían pruebas de intrusión a empresas.[144]

De la última década a la actualidad, realmente no ha habido mucho cambio se han centrado más en obtener mejores prácticas, cogiendo relevancia la integración de IA causa que las herramientas de pentesting comenzaron a incorporar inteligencia artificial para simular ataques más sofisticados y adaptativos.Ultimamente se adopta un enfoque de pruebas de intrusión continuas, permitiendo una evaluación constante de la seguridad y una respuesta más rápida a nuevas vulnerabilidades además las organizaciones comenzaron a adoptar estrategias más proactivas, utilizando las pruebas de intrusión no solo para identificar fallos, sino también para anticiparse a posibles amenazas.

2.0.3 Tipos de test de intrusión

2.0.3.1 Test de intrusión externo

El **test de intrusión externo** es una simulación controlada de un ataque informático realizada desde fuera de la red corporativa (por ejemplo, desde Internet) [57].Su objetivo es identificar vulnerabilidades expuestas al público, como pueden ser:

- Sitios web.
- Servidores expuestos.
- Dispositivos perimetrales (firewalls, routers).
- Aplicaciones en la nube.

Enfoque: Simular el rol de un atacante externo que no tiene acceso previo a la red interna.

Objetivos principales:

- Evaluar el nivel de exposición pública.
- Identificar brechas en el perímetro.
- Detectar vulnerabilidades en servicios expuestos.

2.0.3.2 Test de intrusión interno

El **test de intrusión interno** es una simulación controlada de un ataque realizada desde dentro de la red interna de la organización. Se realiza bajo el supuesto de que un atacante ha conseguido acceso a la red interna (por ejemplo, mediante phishing o un dispositivo comprometido).[68]

Enfoque: Simular el comportamiento de un empleado malintencionado o de un atacante que logró acceso a la red interna.

Objetivos principales:

- Evaluar la segmentación de red.
- Identificar privilegios mal gestionados.
- Detectar vulnerabilidades internas y rutas de escalado.

2.0.3.3 Enfoques de test de intrusión

En el ámbito de los test de intrusión, existen tres enfoques clásicos conocidos como pruebas de caja negra, caja blanca y caja gris[135]. Estos enfoques se diferencian según el nivel de conocimiento que el evaluador tiene sobre el sistema antes de comenzar las pruebas a las que se somete, se definen por las siguientes características:

- **Caja Negra (Black-box):**
  - El pentester no tiene conocimiento previo del sistema.
  - Simula el comportamiento de un atacante externo.
  - Es el tipo más cercano a un ataque real.
  - Requiere mayor tiempo de reconocimiento.
  - Se evalua la seguridad externa y resistencia de la infraestructura frente a atacantes desconocidos.
- **Caja Blanca (White-box):**
  - El evaluador tiene acceso completo a la información del sistema: código fuente, configuraciones y arquitectura.
  - Permite un análisis exhaustivo y profundo, con ello identificar problemas de seguridad o hardware
  - Es eficaz para comprobar la calidad del código detectando así fallos lógicos
  - Detecta vulnerabilidades internas, midiendo la efectividad de las medidas de seguridad implementada de forma interna
- **Caja Gris (Grey-box):**
  - Combinación de elementos de la caja blanca y negra.
  - Se dispone de información parcial, como credenciales limitadas o diagramas de red.
  - Simula un atacante con acceso restringido,con algún tipo de acceso o información privilegiada, como un empleado con permisos básicos,
  - Ofrece un equilibrio entre realismo y profundidad técnica, es decir, se combina la perspectiva interna y externa de manera equilibrada habilitando una evaluación más completa de las vulnerabilidades.

2.0.4 Fases del pentesting

2.0.4.1 1. Acuerdo de los objetivos

En la primera fase se establecen los objetivos específicos de la auditoría de seguridad. Estos pueden ir desde detectar vulnerabilidades técnicas hasta evaluar la eficacia de las políticas de seguridad existentes o verificar el cumplimiento de normas legales como el RGPD. Los objetivos deben estar claramente definidos y acordados entre la organización y el equipo auditor, ya que serán la base para las decisiones metodológicas posteriores. Es esencial identificar qué se desea proteger , de quién y cómo se medirá el éxito de la auditoría. También se debe considerar si la auditoría se hará de forma interna o externa, si será con o

sin conocimiento previo del personal, y que profundidad abarcan dichas pruebas. Un acuerdo sólido en esta etapa garantiza una auditoría consolidada con las necesidades reales de la organización, facilita evitar malentendidos durante su ejecución.[116, 96]

2.0.4.2 2. Alcance y condiciones del test de intrusión

En esta etapa se define el perímetro de análisis de la auditoría. Se determinan los sistemas, aplicaciones, redes y dispositivos que posteriormente serán objeto del test de intrusión. Por otro lado se especifican las condiciones bajo las cuales se desarrollará el trabajo: si se hará en producción o entornos de pruebas, si se podrá interrumpir el servicio, o si habrá acceso físico o lógico. Se deben acordar tanto las técnicas permitidas (explotación de vulnerabilidades, ingeniería social, etc.) como aquellas que quedan fuera del alcance por motivos legales, técnicos o de disponibilidad. Un alcance bien definido comprende los riesgos que pueden suceder durante la auditoria por lo que ayuda a evitar interferencias innecesarias en la operativa diaria de la organización y enfoca los esfuerzos en las áreas más críticas. Esta fase también implica establecer con exactitud las métricas de éxito, el crono-grama de trabajo, los puntos de contacto y los procedimientos de emergencia en caso de encontrar fallos críticos o comprometer accidentalmente sistemas sensibles.[116]

2.0.4.3 3. Recolección inicial de permisos y autorizaciones

Antes de comenzar cualquier actividad técnica, es indispensable obtener todas los permisos necesarios para proceder a realizar la auditoría. Esto incluye permisos firmados por las partes responsables de los sistemas que serán evaluados, cláusulas de exención de responsabilidad y acuerdos de confidencialidad. Esta documentación protege tanto al equipo auditor como a la organización frente a implicaciones legales en caso de fallos, interrupciones o accesos no deseados. También deja constancia del consentimiento mutuo respecto a los procedimientos autorizados y los límites del análisis. En muchos casos, estas autorizaciones forman parte de un contrato de servicios más amplio, donde también se detalla la propiedad de los hallazgos, el tratamiento de datos personales y las posibles obligaciones de reportar vulnerabilidades a terceros. Este paso no solo es una práctica ética, sino también legal, y su omisión puede poner en riesgo todo el proceso y generar consecuencias civiles o penales para las partes involucradas.[116]

2.0.4.4 4. Validación legal y de permisos del alcance y condiciones del test de intrusión

Una vez recogidos los permisos iniciales, se debe tanto revisar como validar legalmente el alcance y condiciones del test de intrusión. Esta validación implica revisar que todo lo acordado no incumpla ninguna ley de forma que esté alineado con las normativas locales e internacionales sobre protección de datos, delitos informáticos, privacidad y uso responsable de sistemas de información. Es común que esta validación la realicen equipos jurídicos especializados que verifiquen que no se vulneren derechos de terceros, especialmente cuando el entorno a auditar incluye datos personales o servicios en la nube. Además, se asegura que los métodos propuestos (por ejemplo, fuerza bruta, escaneo de puertos, sniffing) no violen ninguna legislación vigente. Esta fase también contempla revisar acuerdos con terceros (como proveedores de hosting o software) para evitar incumplimientos contractuales. La validación legal garantiza que la auditoría se ejecute de forma ética, segura y jurídicamente

respaldada, lo cual es especialmente importante en auditorías externas o en sectores regulados como banca, salud o administración pública.[116]

2.0.4.5 5. Recolección de información

Esta fase es fundamental para comprender el entorno tecnológico de la organización. Se recopila información detallada sobre la infraestructura, incluyendo sistemas operativos, aplicaciones, configuraciones de red, políticas de seguridad y procedimientos operativos. Las técnicas empleadas pueden ser pasivas, como la revisión de documentación y entrevistas con el personal clave, o activas, como escaneos de red y pruebas de penetración controladas. El objetivo es identificar los activos críticos, entender cómo se comunican entre sí y detectar posibles puntos de entrada para un atacante. Una recolección de información exhaustiva proporciona una base sólida, facilitando así que no implique tanta complejidad para las fases posteriores de análisis y evaluación de vulnerabilidades. Además, ayuda a poner en contexto los hallazgos en función de la arquitectura y operaciones que detallan el método de trabajo de la organización.[116, 96]

2.0.4.6 6. Análisis de las vulnerabilidades

Con la información recopilada, se procede a identificar y evaluar las vulnerabilidades presentes en los sistemas y procesos. Se utilizan herramientas especializadas para detectar debilidades más frecuentes, configuraciones incorrectas y posibles brechas de seguridad. Este análisis no solo se enfoca en aspectos técnicos, sino también en procedimientos y políticas que puedan representar riesgos. Las vulnerabilidades se clasifican según su severidad y la probabilidad de explotación, permitiendo priorizar las acciones correctivas. Es esencial validar los hallazgos para evitar falsos positivos y garantizar que las recomendaciones sean relevantes y aplicables al entorno específico de la organización. Este análisis proporciona una visión clara de los riesgos actuales y potenciales, facilitando la toma de decisiones informadas para mejorar la seguridad.[116, 96]

2.0.4.7 7. Explotación de vulnerabilidades

En esta fase considerada crítica, ya que se utilizan las vulnerabilidades identificadas previamente para intentar acceder a los sistemas objetivo. Se emplean herramientas y técnicas específicas para explotar estas debilidades, con el fin de evaluar el impacto potencial de un ataque real . Es crucial realizar estas acciones de manera controlada y documentada, para evitar daños colaterales y garantizar la integridad de los sistemas durante la prueba, de manera que buscamos documentar en la gran mayoría cómo se ha hecho, es decir, demostrar que cabe la posibilidad de realizarlo. Existen excepciones como por ejemplo las redes wifi que se ven en la necesidad de explotarlas, ya que conlleva a permitir a partir de ellar descubrir nuevas máquinas y atacar a esas máquinas que aparentemente no están visibles, por otra parte también hay otro motivo de peso, estas explotaciones de vulnerabilidad que se dan sin generar daños en la empresa tanto en sistemas de información como paradas de servicios. La explotación efectiva proporciona una visión clara de las posibles rutas de ataque y ayuda a priorizar

las medidas de mitigación necesarias.[116]

2.0.4.8 8. Redacción de informe de auditoría

La fase final consiste en la elaboración de un informe detallado que documenta los hallazgos de la auditoría. Este informe debe ser claro, conciso y comprensible para diferentes audiencias, desde técnicos hasta directivos. Incluye una descripción del alcance, la metodología utilizada, los hallazgos identificados, su impacto potencial y recomendaciones específicas para mitigar los riesgos existentes. Además, puede contener un plan de acción propuesto, priorizando las medidas correctivas según su urgencia y efectividad. Un informe bien estructurado no solo comunica los problemas detectados, sino que también sirve como guía para mejorar la postura de seguridad de la organización de cara al futuro. Es recomendable que el informe sea revisado conjuntamente con las partes interesadas para asegurar la comprensión y el compromiso con las acciones propuestas.[116, 96]

2.0.4.9 9. Presentación de resultados al personal técnico y formación

Tras la elaboración del informe de auditoría, es fundamental presentar los hallazgos al personal técnico de la organización en un lenguaje que sea apto para todo el público, debido a que el personal técnico usa un idioma distinto. Esta presentación debe ser clara y detallada, destacando las vulnerabilidades encontradas, su impacto potencial y las recomendaciones para su mitigación de forma que sea más formativo que presentativo. Persiguiendo que el personal técnico de la empresa reciba una formación específica para que de cara al futuro pueda mitigar los problemas de la mano de unas buenas prácticas, fomentando así una cultura de ciberseguridad proactiva dentro de la organización. [116]

2.0.5 Herramientas

En el ámbito de la auditoría de seguridad informática, las herramientas especializadas desempeñan un papel crucial en la identificación y mitigación de riesgos. Estas herramientas permiten a los auditores realizar evaluaciones exhaustivas de los sistemas, detectar vulnerabilidades y asegurar el cumplimiento de las políticas de seguridad establecidas.
El estado del arte en auditoría de seguridad se caracteriza entre ellas por las herramientas avanzadas que facilitan un análisis detallado de los sistemas informáticos. Entre las herramientas más destacadas se encuentran:

- **Recogida de información básica**
  - **A través de OSINT:** Método de recopilación de información a partir de fuentes públicas como redes sociales, registros públicos y sitios web.[116]
  - **A través del DNS:** La información del DNS puede ser utilizada para mapear la infraestructura de una organización y detectar posibles puntos de entrada para ataques.[116]
  - **Lynis:** Herramienta de auditoría de seguridad para sistemas Unix/Linux. Realiza análisis exhaustivos del sistema para identificar configuraciones inseguras y vulnerabilidades.[116]

- **Nbtscan:** Utilidad que escanea redes para extraer información de SMB.[115]
- **Nikto:** Escáner de servidores web que detecta vulnerabilidades comunes, configuraciones erróneas y archivos peligrosos en aplicaciones web.[116]
- **Plecost:** Herramienta que analiza instalaciones de WordPress para identificar plugins vulnerables . [115]
- **Wpscan:** Escáner de seguridad para sitios WordPress que detecta vulnerabilidades en el núcleo, temas y plugins instalados.[116]
- **JoomScan:** Herramienta diseñada para detectar vulnerabilidades conocidas en sitios web que utilizan el plugin de Joomla. [115]

- **Análisis básico de vulnerabilidades**
  - **Yersinia:** Framework que permite realizar ataques a protocolos de red de capa 2, como STP, CDP y DHCP, simulando ataques para evaluar la seguridad de la red.
  - **Sparta/Legion:** Herramientas con interfaz gráfica que automatizan tareas de escaneo y enumeración de redes, facilitando la identificación de servicios y posibles vulnerabilidades.[116]

- **Auditorías a redes wifi**
  - **Cifrado WEP:** Protocolo de seguridad para redes inalámbricas que ha sido superado por su vulnerabilidad a ataques, permitiendo la recuperación de claves con relativa facilidad.[116]
  - **Pixewps:** Herramienta que explota vulnerabilidades en la implementación de WPS (Wi-Fi Protected Setup) para recuperar claves WPA/WPA2 mediante ataques de fuerza bruta offline.[116]
  - **Airodump-ng:** Utilidad que captura paquetes de redes inalámbricas, permitiendo la recopilación de información sobre puntos de acceso y clientes conectados.[116]
  - **Aircrack-ng:** suite de herramientas para auditar redes inalámbricas, que permite descifrar claves WEP y WPA/WPA2 capturando y analizando paquetes de datos.[116]

- **Ataque a contraseñas**
  - **Hydra:** Herramienta que realiza ataques de fuerza bruta y de diccionario contra diversos servicios de autenticación, como SSH, FTP y HTTP, para probar la fortaleza de las contraseñas.[116]
  - **John the Ripper:** Programa que permite recuperar contraseñas cifradas mediante ataques de diccionario y fuerza bruta, siendo útil para evaluar la seguridad de las contraseñas almacenadas.[116]
  - **Mimikatz:**Herramienta que extrae credenciales almacenadas en memoria en sistemas Windows, permitiendo la obtención de contraseñas en texto claro y hashes.[116]

- **Auditorías a aplicaciones web**

- **Inyección SQL:** Técnica de ataque que inserta código SQL malicioso en formularios o URLs para manipular bases de datos, permitiendo el acceso no autorizado a información sensible.[116]
- **XSS:** Vulnerabilidad que permite la inyección de scripts maliciosos en páginas web, afectando a los usuarios que las visitan.[116]
- **OWASP:** Proyecto que proporciona recursos y herramientas para mejorar la seguridad de las aplicaciones web, incluyendo listas de las vulnerabilidades más críticas.[116]
- **OWASP Zed Attack Proxy:** Herramienta de código abierto que ayuda a encontrar vulnerabilidades en aplicaciones web mediante escaneos automatizados y pruebas manuales.[116]
- **Nikto:** (Mencionado anteriormente) Escáner de servidores web que detecta vulnerabilidades comunes y configuraciones erróneas.[116]
- **Sqlmap:** Herramienta que automatiza la detección y explotación de vulnerabilidades de inyección SQL, permitiendo la extracción de datos de bases de datos comprometidas.[116]
- **Wapiti:** Escáner que identifica vulnerabilidades en aplicaciones web, como inyecciones SQL y XSS, mediante el análisis de las respuestas del servidor a diferentes entradas.[116]
- **Wpscan:** (Mencionado anteriormente) Escáner de seguridad para sitios WordPress que detecta vulnerabilidades en el núcleo, temas y plugins instalados.[116]

- **Metasploit**: Plataforma que proporciona herramientas para desarrollar, probar y ejecutar exploits contra sistemas remotos, facilitando la realización de pruebas de penetración y la evaluación de la seguridad de los sistemas.[116]

Estas herramientas, entre otras, conforman el conjunto de recursos técnicos que los auditores emplean para llevar a cabo evaluaciones de seguridad efectivas. Su correcta aplicación no solo mejora la capacidad de detección de vulnerabilidades, sino que también contribuye a la implementación de soluciones preventivas y correctivas adecuadas.

2.0.6 Pruebas de Penetración y Cumplimiento Normativo

En esta sección, se analizarán las principales normativas y estándares que, directa o indirectamente, implican la realización de pruebas de penetración como mecanismo para garantizar el cumplimiento normativo y la mejora continua de la seguridad en las organizaciones.

2.0.6.1 GDPR (Reglamento General de Protección de Datos)

El GDPR es una regulación de la Unión Europea que establece normas para la protección de datos personales de los ciudadanos de la UE y del Espacio Económico Europeo (EEE). Aunque no exige explícitamente la realización de pruebas de penetración, el artículo 32 del GDPR establece que las organizaciones deben implementar medidas técnicas y organizativas apropiadas para garantizar un nivel de seguridad adecuado al riesgo. Esto incluye la necesidad de probar y evaluar regularmente la eficacia de estas medidas, lo que hace que las pruebas de penetración sean una práctica recomendada para cumplir con el GDPR.[47, 50, 14]

2.0.6.2 Esquema Nacional de Seguridad (ENS)

El Esquema Nacional de Seguridad (ENS), regulado por el Real Decreto 311/2022, establece los principios y requisitos mínimos necesarios para garantizar la protección adecuada de la información en el ámbito de la Administración Pública española y entidades del sector privado que colaboran con ella. Aunque el ENS no exige explícitamente la realización de pruebas de penetración, sí requiere la implementación de medidas de seguridad proporcionales al nivel de riesgo, lo que puede incluir este tipo de pruebas.[47, 53]

Entre los aspectos relevantes del ENS relacionados con las pruebas de seguridad se encuentran:

- **Análisis y gestión de riesgos**: Identificación y evaluación de riesgos para determinar las medidas de seguridad adecuadas.
- **Auditorías de seguridad**: Evaluaciones periódicas para verificar el cumplimiento de las medidas de seguridad implementadas.
- **Mejora continua**: Revisión y actualización constante de las medidas de seguridad para adaptarse a nuevos riesgos y tecnologías.

La adopción de pruebas de penetración como parte de las estrategias de seguridad permite a las organizaciones alinearse con los principios del ENS, proporcionando una evaluación práctica de la eficacia de sus controles y contribuyendo a la mejora continua de su sistema de gestión de seguridad de la información.

2.0.6.3 NIST SP 800-115 (Guía Técnica para Pruebas de Seguridad de la Información)

La publicación especial NIST SP 800-115,proporciona directrices para la planificación y ejecución de pruebas técnicas de seguridad de la información. Aunque no impone requisitos obligatorios, esta guía es ampliamente reconocida y utilizada como referencia para realizar pruebas de penetración estructuradas y efectivas.[132]

El NIST SP 800-115 establece un enfoque metodológico que incluye las siguientes fases:

- **Planificación**: Definición de objetivos, alcance y reglas de compromiso para las pruebas.
- **Descubrimiento**: Recolección de información y análisis de los sistemas objetivo para identificar posibles vulnerabilidades.
- **Ataque**: Ejecución de pruebas para explotar las vulnerabilidades identificadas y evaluar el impacto potencial.
- **Informe**: Documentación de los hallazgos, análisis de riesgos y recomendaciones para mitigar las vulnerabilidades detectadas.

Este marco proporciona una base sólida para que las organizaciones evalúen la eficacia de sus controles de seguridad y mejoren continuamente su postura de ciberseguridad.

2.0.6.4 ISO/IEC 27001

La norma ISO/IEC 27001 es un estándar internacional desarrollado por la ISO(Organización Internacional de Normalización) y la IEC(Comisión Electrotécnica Internacional), no obliga explícitamente a realizar pruebas de penetración.Sin embargo, enfatiza la necesidad de evaluar la eficacia de los controles de seguridad implementados, mediante un ciclo de mejora continuo (Planificar, Hacer, Verificar y Actuar). Ayudando así a las organizaciones a gestionar los riesgos de seguridad, mejorar la confianza de los clientes y socios, cumplir con regulaciones legales y mejorar la reputación de la empresa. En entornos complejos o con aplicaciones personalizadas, las pruebas de penetración son esenciales para garantizar la protección adecuada de la información.[14, 47, 50]

2.0.6.5 PCI DSS (Estándar de Seguridad de Datos para la Industria de Tarjetas de pago)

El estándar PCI DSS exige la realización de pruebas de penetración internas y externas al menos una vez al año, así como después de cualquier cambio significativo en la infraestructura de TI. Estas pruebas son fundamentales para identificar y corregir vulnerabilidades que podrían comprometer la seguridad de los datos de tarjetas de pago.[47, 50, 14]

2.0.6.6 SWIFT CSCF (Sociedad para las Telecomunicaciones Interbancarias y Financieras Mundiales)

El marco CSCF de SWIFT establece controles de seguridad obligatorios para las instituciones financieras que utilizan la red SWIFT. Entre estos controles, se incluye la realización de pruebas de penetración regulares para validar la configuración de seguridad operativa e identificar posibles brechas de seguridad.[14]

2.0.6.7 Guía de Pruebas de Seguridad de OWASP (WSTG)

La Guía de Pruebas de Seguridad de OWASP (WSTG) es un recurso fundamental para la evaluación de la seguridad en aplicaciones web. Desarrollada por la comunidad de OWASP, proporciona un marco estructurado para realizar pruebas de seguridad, abarcando desde la recopilación de información hasta la explotación de vulnerabilidades.[107]

Aunque la WSTG no es una norma certificable, su adopción es ampliamente reconocida en la industria como una práctica recomendada para garantizar la seguridad de las aplicaciones web. Implementar las metodologías descritas en la WSTG puede ayudar a las organizaciones a cumplir con requisitos de seguridad establecidos en normativas como el Reglamento General de Protección de Datos (GDPR) y la Directiva NIS2, al proporcionar evidencia de pruebas de seguridad realizadas de manera sistemática y exhaustiva.

2.0.6.8 ISO/IEC 27002:2022

La norma ISO/IEC 27002:2022 ofrece un conjunto de controles de seguridad de la información, ciberseguridad y protección de la privacidad. Esta revisión de 2022 reorganiza los controles en cuatro categorías: organizacionales, de personas, físicos y tecnológicos, sumando un total de 93 controles.

Aunque la ISO/IEC 27002:2022 no prescribe explícitamente la realización de pruebas de penetración, varios de sus controles implican la necesidad de evaluar y probar la eficacia de las medidas de seguridad implementadas. Por ejemplo, el control 8.8 se centra en la gestión de vulnerabilidades técnicas, lo que puede incluir la realización de pruebas de penetración para identificar y remediar debilidades en los sistemas.[139]

Tabla 2.1. *Resumen de Normativas y Estándares (Parte 1)*

Normativa/Estándar	Obligatoriedad	Ámbito de Aplicación	Relación con Pruebas de Penetración
GDPR (Reglamento General de Protección de Datos)	No obligatorio, pero recomendado	Unión Europea	El artículo 32 establece la necesidad de implementar medidas técnicas y organizativas apropiadas, incluyendo la evaluación regular de su eficacia.
Esquema Nacional de Seguridad (ENS)	No obligatorio, pero puede ser requerido según el nivel de seguridad	España	Requiere medidas de seguridad proporcionales al nivel de riesgo, lo que puede incluir pruebas de penetración.
NIST SP 800-115	No obligatorio, pero proporciona directrices detalladas	Estados Unidos	Ofrece un enfoque metodológico para la planificación y ejecución de pruebas técnicas de seguridad de la información.
ISO/IEC 27001	No obligatorio, pero implica la evaluación de controles	Internacional	Enfatiza la necesidad de evaluar la eficacia de los controles de seguridad implementados mediante un ciclo de mejora continuo.

2.0.6.9 SOC 2 (Controles de Organizaciones de Servicios 2)

SOC 2 es un estándar desarrollado por el AICPA()Instituto Americano de Contadores Públicos Certificados) que evalúa los controles internos de una organización relacionados con la seguridad, disponibilidad, integridad del procesamiento, confidencialidad y privacidad de los datos. Aunque SOC 2 no exige explícitamente la realización de pruebas de penetración, se recomienda su implementación para demostrar la eficacia de los controles de seguridad y cumplir con los criterios de confianza del servicio establecidos por el AICPA.[118]

La realización de pruebas de penetración en el contexto de SOC 2 puede ayudar a identificar posibles vulnerabilidades y áreas de mejora en los sistemas de una organización, fortaleciendo así su postura de seguridad y aumentando la confianza de los clientes y socios comerciales.

2.0.6.10 Comparativa Normativas y Estándares Relacionados con Pruebas de Penetración

2.0.7 Pentesting en aplicaciones móviles

El crecimiento exponencial del uso de dispositivos móviles ha transformado la manera en que las personas interactúan con la tecnología, ya que hoy en día, la información personal y sensible se almacena en los dispositivos móviles, que pueden albergar desde datos bancarios y credenciales de accesos hasta registros de salud y conversaciones y comunicaciones privadas.

Tabla 2.2. *Resumen de Normativas y Estándares (Parte 2)*

Normativa/Estándar	Obligatoriedad	Ámbito de Aplicación	Relación con Pruebas de Penetración
PCI DSS	Obligatorio	Internacional	Exige pruebas de penetración internas y externas al menos una vez al año y después de cambios significativos en la infraestructura.
SWIFT CSCF	Obligatorio	Internacional	Establece controles de seguridad obligatorios, incluyendo pruebas de penetración regulares para validar la configuración de seguridad operativa.
Guía de Pruebas de Seguridad de OWASP (WSTG)	No obligatorio, pero ampliamente adoptado	Internacional	Proporciona un marco estructurado para realizar pruebas de seguridad en aplicaciones web, reconocido como una práctica recomendada.
ISO/IEC 27002:2022	No obligatorio, pero incluye controles relacionados	Internacional	Ofrece un conjunto de controles de seguridad de la información, incluyendo la gestión de vulnerabilidades técnicas que pueden implicar pruebas de penetración.
SOC 2	No obligatorio, pero recomendado	Estados Unidos	Evalúa los controles internos relacionados con la seguridad, disponibilidad, integridad del procesamiento, confidencialidad y privacidad de los datos; se recomienda la realización de pruebas de penetración para demostrar la eficacia de los controles.

Sin embargo, este considerable aumento también ha ampliado la superficie de ataque para los ciberdelincuentes.Teniendo en cuenta que estos dispositivos, superan en valor y sensibilidad a la almacenada en los ordenadores tradicionales, por ello esta concentración de datos críticos en los dispositivos móviles, se ha convertido en uno de los principales objetivos de los ciberdelincuentes.A consecuencia de esto, se ha vuelto una práctica esencial recurrir a las pruebas de penetración en dispositivos móviles se ha convertido en una práctica esencial para identificar y mitigar vulnerabilidades específicas de estas plataformas, garantizando la seguridad de los datos y la integridad de las aplicaciones móviles.

2.0.7.1 Panorama actual de amenazas en aplicaciones móviles

El panorama de amenazas en aplicaciones móviles ha experimentado una evolución significativa en los últimos años, con un aumento notable en la sofisticación y frecuencia de los ataques. Según el *Cyber Security Report 2025* de Check Point Software Technologies, se ha observado un incremento del 45% en los ciberataques a nivel global, destacando la urgencia de fortalecer las defensas cibernéticas. [25]

- **Incremento de Malware Móvil**

  Uno de los aspectos más preocupantes es la proliferación de malware diseñado específicamente para dispositivos móviles. Check Point ha identificado un aumento en la actividad de troyanos como *Joker* y *Anubis*, que se infiltran en aplicaciones legítimas para robar información sensible y credenciales de los usuarios. [23]

- **Explotación de Vulnerabilidades en Aplicaciones Populares**

  Los atacantes han centrado sus esfuerzos en explotar vulnerabilidades presentes en aplicaciones populares. Por ejemplo, se ha detectado una campaña que utiliza páginas CAPTCHA falsas para distribuir el malware *Lumma Stealer*, afectando a múltiples países y aprovechando vectores de infección como descargas de juegos modificados y correos electrónicos de phishing dirigidos a usuarios de GitHub [22].

- **Amenazas Avanzadas Persistentes (APT) y Ataques Dirigidos**

  Las Amenazas Avanzadas Persistentes (APT) han adoptado técnicas más sofisticadas para comprometer dispositivos móviles. Grupos como *APT29* han llevado a cabo campañas de phishing dirigidas a diplomáticos europeos, utilizando herramientas basadas en la nube para evadir las medidas de seguridad tradicionales [21].

- **Uso Malicioso de la Inteligencia Artificial**

  La inteligencia artificial (IA) se ha convertido en una herramienta tanto para defensores como para atacantes. Los ciberdelincuentes están utilizando IA generativa para crear campañas de phishing más convincentes y desarrollar malware con capacidades avanzadas de evasión, lo que dificulta su detección por las soluciones de seguridad convencionales [24].

- **Tendencias Emergentes en Ciberseguridad Móvil**

  Además de las amenazas mencionadas, se observa una tendencia creciente en ataques dirigidos a infraestructuras críticas mediante botnets como *AndroxghOst*, que integran capacidades de *Mozi* para comprometer dispositivos IoT y servidores web, ampliando así su alcance y potencial destructivo [23].

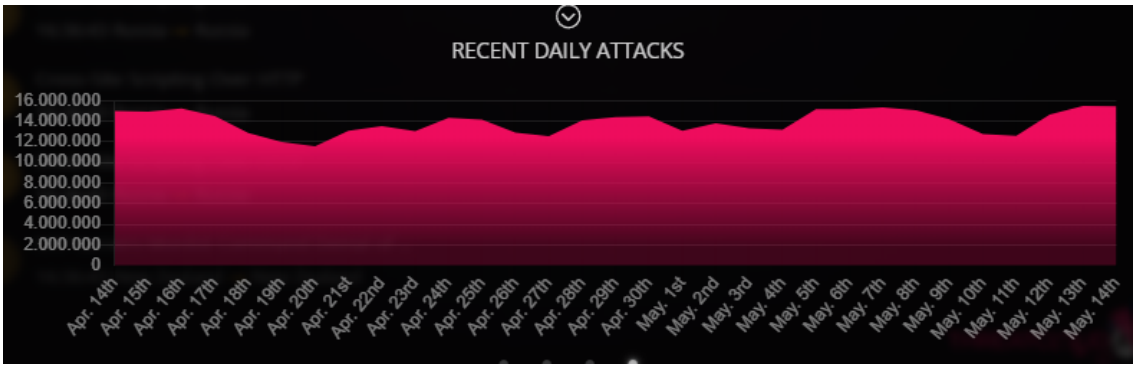


Figura 2.1. *Evolución de ataques cibernéticos diarios a nivel mundial (abril-mayo 2025). Fuente: Check Point ThreatCloud.*

- **Ataques Recientes a Nivel Mundial**

  La Figura 2.1 muestra la evolución de los ataques cibernéticos diarios a nivel mundial, según los datos de Check Point ThreatCloud hasta el 14 de mayo de 2025. Se observa que el volumen de ataques diarios se mantiene en cifras extremadamente elevadas, superando consistentemente los 10 millones de incidentes por día durante el último mes, donde el principal objetivo son los móviles mediante el malware. [26]

  Este patrón revela una persistencia elevada en la actividad maliciosa global, alcanzando picos cercanos a los 15 millones de ataques diarios en fechas clave como el 30 de abril y el 6 de mayo. Estas cifras evidencian la creciente presión a la que están sometidas las infraestructuras digitales, incluidas las aplicaciones móviles, que forman parte importante de estos vectores de ataque.[26]

  Este escenario refuerza la necesidad de implementar prácticas rigurosas de pentesting en aplicaciones móviles como medida preventiva y de protección frente a un entorno de amenazas cada vez más activo y sofisticado.

2.0.7.2 Vulnerabilidades frecuentes en aplicaciones móviles

El auge de las aplicaciones móviles ha traído consigo numerosos desafíos en términos de ciberseguridad. Estas aplicaciones, al manejar datos sensibles y operar en dispositivos personales, se convierten en objetivos atractivos para los atacantes. OWASP (Proyecto abierto de seguridad de aplicaciones web) publica periódicamente el OWASP Mobile Top 10, un listado de las vulnerabilidades más críticas en aplicaciones móviles. A continuación, se describen las 10 vulnerabilidades más frecuentes, detallando su naturaleza, impacto y ejemplos de buenas prácticas para mitigarlas.[108]

1. **Uso Inadecuado de Credenciales**

   Esta vulnerabilidad ocurre cuando las aplicaciones manejan de forma insegura las credenciales de los usuarios, como almacenar contraseñas en texto plano, incluir claves API en el código fuente o utilizar credenciales predeterminadas. Estas prácticas facilitan que atacantes obtengan acceso no autorizado a sistemas y datos sensibles. Es fundamental implementar mecanismos seguros de almacenamiento y transmisión de credenciales, como el uso de almacenes seguros, cifrado adecuado y protocolos de autenticación robustos.[108]

2. **Seguridad Inadecuada en la Cadena de Suministro**

La integración de componentes de terceros, como bibliotecas y SDKs, sin una evaluación adecuada puede introducir vulnerabilidades en la aplicación. Un atacante podría comprometer estos componentes para insertar código malicioso o explotar fallos existentes. Es esencial realizar auditorías de seguridad a los componentes externos, validar sus fuentes y mantenerlos actualizados para mitigar estos riesgos.[108]

3. **Autenticación y Autorización Inseguras**

   Implementaciones deficientes de mecanismos de autenticación y autorización pueden permitir que usuarios no autorizados accedan a funcionalidades o datos restringidos. Esto incluye el uso de contraseñas débiles, falta de verificación de tokens o sesiones, y controles de acceso mal configurados. Es crucial aplicar principios de seguridad como la autenticación multifactor, la gestión adecuada de sesiones y el principio de mínimo privilegio para proteger la aplicación.[108]

4. **Validación Insuficiente de Entradas/Salidas**

   La falta de validación adecuada de los datos que la aplicación recibe o envía puede permitir ataques como inyecciones de código, corrupción de datos o exposición de información sensible. Por ejemplo, un atacante podría introducir código malicioso en campos de entrada que no son correctamente sanitizados. Implementar validaciones estrictas y sanitización de datos es esencial para prevenir estos ataques, utilizando siempre listas blancas y validaciones tanto del lado cliente como del servidor.[108]

5. **Comunicación Insegura**

   Transmisión de datos sensibles sin cifrado adecuado, como el uso de HTTP en lugar de HTTPS, expone la información a posibles interceptaciones por parte de atacantes. Esto puede incluir credenciales, datos personales o información financiera. Es vital utilizar protocolos de comunicación seguros y certificados válidos para proteger la integridad y confidencialidad de los datos en tránsito, aplicando medidas como HSTS (HTTP Strict Transport Security).[108]

6. **Controles de Privacidad Inadecuados**

   La recopilación, almacenamiento o uso de datos personales sin el consentimiento adecuado o sin medidas de protección puede violar la privacidad del usuario y las regulaciones de protección de datos. Esto incluye el acceso no autorizado a información como ubicación, contactos o mensajes. Implementar políticas de privacidad claras, obtener el consentimiento explícito y controles de acceso adecuados es fundamental para proteger la información del usuario.[108]

7. **Protecciones Binarias Insuficientes**

   La ausencia de medidas como la ofuscación del código o la detección de manipulación permite que atacantes realicen ingeniería inversa, modifiquen la aplicación o inserten código malicioso. Esto puede comprometer la integridad de la aplicación y la seguridad de los datos. Aplicar técnicas de protección del código, ofuscación, detección de depuración y verificación de la integridad de la aplicación en tiempo de ejecución ayuda a mitigar estos riesgos.[108]

8. **Configuración de Seguridad Incorrecta**

   Configuraciones predeterminadas o erróneas, como permisos excesivos, servicios innecesarios habilitados o exposiciones de información de depuración, pueden abrir puertas a ataques. Estas configuraciones pueden ser explotadas por ata-

cantes para obtener acceso no autorizado o información sensible. Es importante revisar y ajustar las configuraciones de seguridad antes de desplegar la aplicación, eliminando permisos innecesarios y desactivando funciones de depuración en producción.[108]

9. **Almacenamiento Inseguro de Datos**

   Guardar información sensible en ubicaciones no seguras, como almacenamiento externo sin cifrado, puede permitir el acceso no autorizado a dichos datos. Esto incluye contraseñas, tokens de sesión o datos personales. Utilizar mecanismos de almacenamiento seguro, aplicar cifrado fuerte y almacenar la menor cantidad de información sensible posible son prácticas clave para proteger la información almacenada en el dispositivo.[108]

10. **Criptografía Insuficiente**

    El uso de algoritmos de cifrado obsoletos o mal implementados puede comprometer la confidencialidad e integridad de la información. Esto incluye el uso de claves débiles, falta de salting en hashes o implementación incorrecta de protocolos criptográficos. Adoptar prácticas criptográficas modernas, utilizar bibliotecas auditadas y seguir estándares reconocidos es crucial para asegurar los datos y mantener la aplicación protegida frente a ataques criptográficos.[108]

2.0.7.3 Metodologías y Estándares

El *pentesting* en dispositivos móviles se estructura en fases que permiten identificar y mitigar vulnerabilidades en aplicaciones y sistemas operativos móviles. La metodología recomendada por OWASP, en particular la *Mobile Application Security Testing Guide (MASTG)*, proporciona un marco detallado para llevar a cabo estas evaluaciones de seguridad, en las que se garantiza una cobertura completa de vulnerabilidades potenciales mediante metodologías estructuradas que combinan técnicas manuales y automatizadas [106].

1. **Recolección de Información (Reconocimiento)**

   En esta fase inicial, se recopila información relevante sobre el objetivo, incluyendo detalles del sistema operativo, configuraciones de red, servicios expuestos y posibles vectores de ataque. El objetivo es obtener una comprensión profunda del entorno para identificar áreas potenciales de vulnerabilidad [120].

2. . **Análisis Estático**

   El análisis estático implica examinar el código fuente o los binarios del sistema sin ejecutarlos. Se busca identificar vulnerabilidades en la lógica de programación, configuraciones inseguras y posibles puntos débiles que puedan ser explotados [120].

3. **Análisis Dinámico**

   En esta etapa, se ejecuta el sistema en un entorno controlado para observar su comportamiento en tiempo real. Se monitorean las interacciones con el sistema operativo, la red y otros componentes para detectar anomalías o comportamientos inseguros [120].

4. **Explotación**

Una vez identificadas las vulnerabilidades, se intenta explotarlas de manera controlada para evaluar su impacto real. Esta fase ayuda a comprender la gravedad de las vulnerabilidades y la efectividad de las medidas de seguridad existentes [120].

5. **Informe de Resultados**

   La fase final consiste en documentar todos los hallazgos, incluyendo las vulnerabilidades descubiertas, las pruebas realizadas y las recomendaciones para mitigar los riesgos identificados. Este informe es esencial para que los responsables de seguridad puedan tomar decisiones informadas y mejorar la postura de seguridad del sistema [120].

   Es importante destacar que, aunque esta metodología está adaptada para dispositivos móviles, su estructura es muy similar a la de una prueba de penetración general, siguiendo principios y fases comunes en el ámbito de la seguridad informática .

2.0.7.4 Herramientas Actuales para Pentesting en Aplicaciones Móviles

El pentesting en aplicaciones móviles requiere el uso de diversas herramientas especializadas que permiten evaluar la seguridad en cada fase del proceso. A continuación, se presentan las herramientas más relevantes, clasificadas según las etapas del pentesting, siguiendo las directrices de OWASP y considerando las soluciones proporcionadas por Kaspersky [106].

1. **Recolección de Información (Reconocimiento)**

   - **Nmap**: Herramienta para escaneo de redes y detección de servicios [146].
   - **Shodan**: Motor de búsqueda para identificar dispositivos conectados a Internet [90].
   - **Recon-ng**: Framework para realizar tareas de reconocimiento en aplicaciones web [109].
   - **theHarvester**: Herramienta para recolectar correos electrónicos y nombres de dominio [109].

2. **Análisis Estático**

   - **MobSF (Mobile Security Framework)**: Framework para análisis de seguridad de aplicaciones móviles [69].
   - **Jadx**: Herramienta para descompilar archivos APK y obtener el código fuente en Java [109].
   - **Ghidra**: Herramienta de ingeniería inversa desarrollada por la NSA [109].
   - **APKTool**: Herramienta para descompilar y recompilar archivos APK [109].

3. **Análisis Dinámico**

   - **Frida**: Herramienta para instrumentación dinámica de aplicaciones [141].
   - **Objection**: Herramienta para pruebas de seguridad en aplicaciones móviles sin necesidad de root o jailbreak [141].
   - **Burp Suite**: Plataforma para pruebas de seguridad en aplicaciones web [20].
   - **ZAP (Zed Attack Proxy)**: Herramienta de pruebas de penetración en aplicaciones web [147].

4. **Explotación**

- **Metasploit Framework**: Plataforma para desarrollar y ejecutar exploits contra sistemas vulnerables [145].
- **Drozer**: Herramienta para el análisis de seguridad de aplicaciones Android [141].
- **Cobalt Strike**: Herramienta para pruebas de penetración y simulación de ataques [109].
- **Armitage**: Interfaz gráfica para Metasploit que facilita la gestión de ataques [109].

5. **Informe de Resultados**

- **Dradis**: Plataforma para la gestión y presentación de informes de seguridad [109].
- **Faraday**: Entorno colaborativo para la gestión de pruebas de penetración y generación de informes [109].
- **Serpico**: Herramienta para la generación de informes de seguridad personalizados [109].
- **Kali Linux Reporting Tools**: Conjunto de herramientas incluidas en Kali Linux para la generación de informes [109].

2.0.7.5 Conclusión

A lo largo de este capítulo se ha evidenciado cómo el crecimiento imparable del ecosistema móvil ha venido acompañado de un incremento proporcional de riesgos y amenazas cibernéticas. Las aplicaciones móviles, debido a la sensibilidad de los datos que gestionan y a su integración en todos los aspectos de la vida diaria, constituyen hoy un vector de ataque prioritario para los ciberdelincuentes.

El panorama actual de amenazas presenta una evolución constante hacia técnicas más sofisticadas, donde destacan desde el malware específico para móviles hasta las amenazas avanzadas persistentes (APT) y el uso de inteligencia artificial para desarrollar campañas de ataques más convincentes. Este contexto refuerza la importancia de realizar pruebas de seguridad proactivas y continuas sobre estas plataformas.

Asimismo, se han expuesto las principales vulnerabilidades que afectan a las aplicaciones móviles según el OWASP Mobile Top 10, destacando debilidades comunes en almacenamiento, comunicación, autenticación y malas prácticas de programación, que pueden ser fácilmente explotadas si no se abordan de manera adecuada.

Para abordar estas amenazas y vulnerabilidades, se ha detallado la metodología de pentesting en dispositivos móviles, estructurada en fases que permiten una evaluación exhaustiva y sistemática. Esta metodología, ampliamente respaldada por OWASP y la industria, guarda una gran similitud con los procesos de pruebas de penetración tradicionales, adaptándose a las particularidades del entorno móvil.

Finalmente, se ha presentado un repertorio de herramientas especializadas, clasificadas según las fases del pentesting, destacando tanto soluciones abiertas reconocidas por OWASP como aportaciones de compañías líderes en ciberseguridad como Kaspersky. Estas herramientas constituyen un recurso esencial para los profesionales de la seguridad en su labor de evaluación y mitigación de riesgos.

En conclusión, el pentesting en aplicaciones móviles se posiciona hoy como un componente indispensable en cualquier estrategia de seguridad integral, no solo por la creciente sofisticación de las amenazas, sino por la criticidad de la información que estas aplicaciones manejan y el impacto potencial que su explotación puede generar tanto en usuarios individuales como en organizaciones.

2.0.8 Aplicaciones actuales

Integración de la Seguridad en el Ciclo de Vida del Desarrollo de Software (DevSecOps)

La integración de la seguridad en cada fase del ciclo de vida del desarrollo de software, conocida como DevSecOps, se ha convertido en una práctica esencial. Las pruebas de intrusión se incorporan desde las etapas iniciales del desarrollo, permitiendo la identificación y mitigación temprana de vulnerabilidades. Esta integración promueve una cultura de seguridad continua y proactiva, facilitando la colaboración entre equipos de desarrollo, operaciones y seguridad.[10]

Automatización de Pruebas de Penetración

El uso de herramientas automatizadas, como AppCheck, permite realizar escaneos de seguridad eficientes y repetitivos. Estas soluciones automatizan la detección de vulnerabilidades en aplicaciones, sitios web y redes, facilitando la identificación temprana de riesgos y reduciendo la carga de trabajo manual. La automatización también permite realizar pruebas de intrusión de manera más frecuente, mejorando la postura de seguridad general.[13]

Evaluación de la Seguridad en Entornos de Computación en la Nube

Con la creciente adopción de servicios en la nube, las pruebas de intrusión se adaptan para evaluar la seguridad de infraestructuras, plataformas y aplicaciones basadas en la nube. Esto incluye la identificación de configuraciones erróneas y vulnerabilidades específicas de entornos cloud. La evaluación de la seguridad en la nube es crucial para garantizar la protección de datos y servicios críticos.[67]

Pruebas de Seguridad en Dispositivos de Internet de las Cosas (IoT)

La proliferación de dispositivos IoT amplía la superficie de ataque. Las pruebas de intrusión se enfocan en evaluar la seguridad de estos dispositivos, identificando vulnerabilidades en su firmware, protocolos de comunicación y configuraciones predeterminadas, para prevenir accesos no autorizados y ataques a gran escala. La seguridad de los dispositivos IoT es esencial para proteger redes y sistemas interconectados.[32]

Evaluación de la Seguridad de Aplicaciones Móviles y APIs

Las aplicaciones móviles y las interfaces de programación de aplicaciones (APIs) son objetivos frecuentes de ataques. Las pruebas de intrusión se centran en identificar vulnerabilidades como inyecciones SQL, XSS y problemas de autenticación, asegurando la integridad y confidencialidad de los datos manejados por estas aplicaciones. La evaluación de la seguridad de aplicaciones móviles y APIs es fundamental para proteger la información sensible de los usuarios. [97]

2.0.9 Aplicaciones futuras

Aplicación de Inteligencia Artificial y Aprendizaje Automático

La incorporación de inteligencia artificial (IA) y aprendizaje automático (ML) en las pruebas de intrusión permite automatizar la detección y explotación de vulnerabilidades. Herramientas como RapidPen utilizan modelos de lenguaje para realizar pruebas de penetración de manera autónoma, mejorando la eficiencia y precisión de las evaluaciones de seguridad. La IA y el ML

también facilitan la identificación de patrones de ataque y la predicción de amenazas emergentes.[88]

Implementación del Modelo de Confianza Cero (Zero Trust)

El enfoque de "confianza cero" implica verificar continuamente la identidad y el acceso de usuarios y dispositivos, sin asumir confianza por defecto. Las pruebas de intrusión se adaptan para evaluar la eficacia de este modelo, asegurando que las políticas de acceso y autenticación sean robustas y efectivas. La implementación del modelo Zero Trust es esencial para proteger entornos de trabajo híbridos y remotos.[5]

Automatización de la Respuesta a Incidentes de Seguridad

La automatización en la respuesta a incidentes permite una reacción rápida y coordinada ante amenazas de seguridad. Las pruebas de intrusión se integran con sistemas automatizados que detectan y responden a ataques en tiempo real, minimizando el impacto y facilitando la recuperación. La automatización también mejora la eficiencia operativa y reduce la carga de trabajo de los equipos de seguridad.[12]

Desarrollo de Herramientas de Pruebas de Penetración Basadas en Aprendizaje por Refuerzo

El uso de técnicas de aprendizaje por refuerzo en las pruebas de intrusión permite a los sistemas aprender y adaptarse a diferentes entornos de red. Herramientas como RAT aplican esta técnica para descubrir vulnerabilidades en firewalls de aplicaciones web, mejorando la eficacia de las pruebas de seguridad. El aprendizaje por refuerzo también facilita la simulación de ataques avanzados y la evaluación de la resiliencia de los sistemas.[11]

3

Proceso de Consentimiento y Desarrollo del Formulario de Autorización

3.0.1 Introducción

Para llevar a cabo esta auditoría, emplearemos diversas herramientas especializadas en la evaluación de sistemas, redes y aplicaciones web, tales como Lynis, Sparta, Legion, Airodump-ng, Mimikatz, OWASP ZAP, Nikto, SQLMap, Wapiti y WPScan. Estas herramientas nos permitirán realizar un análisis exhaustivo y detallado de los diferentes componentes tecnológicos de la empresa. Para realizar estos análisis, es fundamental destacar que todas las actividades de auditoría se llevarán a cabo con el consentimiento expreso de OFIAUTO, garantizando en todo momento la confidencialidad, integridad y disponibilidad de la información manejada. Asimismo, nos aseguraremos de que las pruebas no interfieran con las operaciones normales de la empresa y de que se cumplan todas las normativas legales y éticas aplicables.

Este capítulo documenta el consentimiento otorgado por la empresa para la realización de la auditoría de seguridad, detallando los objetivos, alcance, metodología, herramientas utilizadas y las condiciones bajo las cuales se llevará a cabo. Consideramos que la colaboración y transparencia entre las partes involucradas son esenciales para el éxito de este proyecto y para fortalecer la seguridad de la información en OFIAUTO.

3.0.2 Formulario de Autorización de Pentesting

Antes de llevar a cabo cualquier auditoría de seguridad, es imprescindible establecer un marco legal y técnico que garantice la autorización explícita del cliente. Este proceso formaliza los permisos necesarios para realizar pruebas que, por su naturaleza, podrían tener implicaciones sobre la disponibilidad o integridad de los sistemas evaluados. A continuación, detallamos el proceso de consentimiento y el contenido típico del formulario de autorización que utilizamos en una auditoría de tipo Pentesting.[116]

El formulario de autorización constituye el documento legal mediante el cual el cliente nos otorga su consentimiento expreso para que llevemos a cabo pruebas de seguridad sobre uno o varios activos específicos. En el caso analizado, el cliente "OFIAUTO " autorizó a nuestra empresa "José Sánchez-Rosso Almoguera " para realizar un test de penetración con las siguientes características:

- **Ámbito:** Incluye pruebas sobre la infraestructura accesible desde la URL institucional (*https://urldelaempresa.es*), abarcando tanto entornos internos como externos.
- **Tipo de prueba:** Las pruebas que vamos a realizar, diferenciando entre interna, externa o ambas, además de especificar si el tipo de pruebas será de caja blanca, negra o gris.
- **Periodo:** Duración de las pruebas, especificando en qué horario se establecerán.
- **Puntos de contacto:** Especificamos los responsables técnicos tanto por parte del cliente como de nuestro equipo auditor, con sus respectivos teléfonos y correos electrónicos, para la notificación de incidencias y en caso de emergencia.

3.0.3 Tipos de Pruebas Autorizadas

En el marco del consentimiento, establecemos claramente el tipo de pruebas que vamos a ejecutar:

- **Pruebas:** Especificamos qué tipo de pruebas vamos a realizar durante la auditoría y explicamos en qué consiste cada una de ellas.
- **Metodología de prueba:** Detallamos los tipos de enfoques que emplearemos.
- **Herramientas:** Describimos las herramientas que utilizaremos durante la auditoría de seguridad, organizadas por fases, junto con su propósito, alcance y posibles impactos.

3.0.4 Restricciones y Conformidades del Cliente

El formulario también impone una serie de requisitos y declaraciones por parte del cliente, además de las conformidades y derechos que este posee:

En caso de conformidad con la concesión de esta autorización, el Cliente declara lo siguiente:

- **Vigencia de la Autorización:** La autorización para llevar a cabo la auditoría será válida desde las fechas acordadas, ambos inclusive.
- **Propiedad de los Sistemas:** El cliente declara ser el propietario legítimo de los sistemas objeto de la auditoría y afirma tener la autoridad necesaria para permitirnos realizar las pruebas de seguridad correspondientes.[116]
- **Copias de Seguridad:** El cliente se compromete a haber realizado copias de seguridad completas y verificadas de todos los sistemas incluidos en el alcance de la auditoría, asegurando la posibilidad de restaurarlos a su estado original en caso de ser necesario.[116]
- **Naturaleza de las Pruebas:** El cliente reconoce que la auditoría implica el uso de herramientas y técnicas diseñadas para detectar vulnerabilidades de seguridad, y acepta que es imposible eliminar completamente todos los riesgos asociados al uso de estas herramientas.[116]
- **Modificaciones del Sistema:** El cliente se compromete a no realizar modificaciones en los sistemas o aplicaciones durante la ejecución del pentesting. En caso de que se realicen cambios, consideraremos que el alcance de la auditoría ha sido alterado, y podremos dar por finalizada la auditoría, entregando un informe con los resultados obtenidos hasta ese momento. Cualquier nueva auditoría sobre sistemas modificados deberá ser negociada como un nuevo encargo.[116]
- **Aceptación de Riesgos:** El cliente reconoce los posibles efectos colaterales de las pruebas de seguridad y acepta los riesgos asociados a la realización de la auditoría.[116]
- **Modificaciones al Acuerdo:** Cualquier cambio en las condiciones y limitaciones descritas deberá realizarse por escrito y ser aceptado por ambas partes.[116]

3.0.5 Acuerdo de Confidencialidad

Como parte del consentimiento, incluimos un acuerdo de confidencialidad que regula el tratamiento de la información sensible:

3.0.5.1 Definición y Alcance

El Acuerdo de Confidencialidad, también conocido como NDA (Non-Disclosure Agreement), es un documento legalmente vinculante mediante el cual las partes involucradas en una auditoría de seguridad nos comprometemos a no divulgar información sensible o confidencial obtenida durante el proceso. Este acuerdo es esencial para proteger los datos técnicos, operativos y estratégicos de la organización auditada.[116]

3.0.5.2 Obligaciones del Auditor

Nos comprometemos a:

- Utilizar la información confidencial exclusivamente para los fines establecidos en la auditoría.
- No divulgar, copiar ni reproducir dicha información sin el consentimiento expreso y por escrito del cliente.
- Implementar medidas de seguridad adecuadas para proteger la información confidencial contra accesos no autorizados.
- Destruir o devolver toda la información confidencial al finalizar la auditoría, según lo acordado con el cliente.

3.0.5.3 Exclusiones

Las obligaciones de confidencialidad no se aplicarán a la información que:

- Sea de dominio público al momento de su divulgación o que llegue a serlo sin incumplimiento del acuerdo.
- Haya sido obtenida legalmente de terceros sin restricciones de confidencialidad.[116]
- Deba ser divulgada por mandato legal o judicial, en cuyo caso notificaremos previamente al cliente, salvo prohibición legal.[116]
- Cuya comunicación o uso sin restricciones haya sido aprobada por la empresa cliente.[116]

3.0.5.4 Derechos de Propiedad Intelectual

Este acuerdo no implica la concesión de derechos de propiedad intelectual sobre la información confidencial. Todos los derechos, títulos e intereses sobre dicha información permanecerán con el emisor. No adquiriremos ningún derecho de propiedad intelectual por el acceso a la información confidencial.[116]

3.0.5.5 Consecuencias del Incumplimiento

El incumplimiento de las obligaciones establecidas en este Acuerdo de Confidencialidad podrá dar lugar a las siguientes consecuencias legales:

- **Responsabilidad Civil:** En caso de incumplimiento, deberemos indemnizar al titular de la información por los daños y perjuicios ocasionados, incluyendo el lucro cesante y el daño emergente, conforme a lo establecido en el Código Civil español.
- **Acciones Judiciales:** El titular de la información podrá emprender acciones legales para obtener la cesación de la conducta infractora y la reparación de los daños causados, incluyendo la posibilidad de solicitar medidas cautelares.
- **Sanciones Contractuales:** Si el acuerdo contempla cláusulas penales, deberemos abonar las penalizaciones estipuladas por incumplimiento, sin perjuicio de las indemnizaciones por daños y perjuicios adicionales.
- **Responsabilidad Penal:** En casos graves, la divulgación no autorizada de información confidencial podrá constituir un delito penal, conforme a lo dispuesto en el Código Penal español, con las consecuencias legales correspondientes.

3.0.5.6 Derechos de Propiedad Intelectual

Este acuerdo no implica la concesión de derechos de propiedad intelectual sobre la información confidencial. Todos los derechos, títulos e intereses sobre dicha información permanecerán con el emisor. No adquiriremos ningún derecho de propiedad intelectual por el acceso a la información confidencial.[116]

3.0.5.7 Duración del Acuerdo

La obligación de confidencialidad permanecerá en vigor durante un período de **tres (3) años** a partir de la finalización de la auditoría, salvo que las partes acuerden un plazo diferente por escrito.

3.0.5.8 Legislación Aplicable

Este acuerdo se regirá e interpretará de acuerdo con las leyes del país en el que realicemos la auditoría, siendo competentes los tribunales de dicha jurisdicción para resolver cualquier disputa derivada del mismo.[116]

3.0.5.9 Firma y Aceptación

Con la firma de este documento, ambas partes manifestamos nuestra conformidad y aceptación de todas las cláusulas y condiciones estipuladas en el presente acuerdo. Asimismo, declaramos haber leído y comprendido el contenido íntegro del mismo, comprometiéndonos a cumplir con las obligaciones y responsabilidades aquí descritas.[116]

3.0.6 Conclusión

El formulario de autorización constituye una pieza fundamental para garantizar un entorno legalmente seguro, éticamente responsable y técnicamente preparado para llevar a cabo pruebas de intrusión. Este documento no solo nos protege a nosotros como auditores y al cliente, sino que también estructura la auditoría en base a límites definidos, conservando la integridad de los sistemas y la confidencialidad de la información evaluada. Véase el **Apéndice** A, para más detalles.

4

Herramientas Utilizadas

4.0.1 Introducción

En el ámbito de la seguridad informática, la selección y aplicación de herramientas especializadas es esencial para llevar a cabo auditorías y pruebas de penetración efectivas. Estas herramientas nos permiten identificar vulnerabilidades, evaluar riesgos y fortalecer la postura de seguridad de los sistemas evaluados.

En este capítulo presentamos las principales herramientas que empleamos en las distintas fases del proyecto, desde la recopilación de información hasta la auditoría de aplicaciones web. Cada herramienta la hemos seleccionado por su eficacia, relevancia en el ámbito profesional y su capacidad para abordar aspectos específicos de la seguridad informática.

A continuación, detallamos las herramientas utilizadas, organizadas según las fases del proceso de auditoría, destacando sus funcionalidades y el papel que desempeñan en la identificación y mitigación de riesgos de seguridad.

4.0.2 Recogida de Información

4.0.2.1 Lynis

Lynis es una herramienta de auditoría de seguridad de código abierto diseñada para sistemas basados en Unix, incluyendo Linux y macOS. Su propósito principal es realizar análisis exhaustivos del sistema para identificar vulnerabilidades, configuraciones inseguras y áreas susceptibles de mejora en la seguridad.[8] Una característica destacada de Lynis es su capacidad para ejecutarse sin necesidad de privilegios de administrador, lo que la hace especialmente útil en entornos donde disponemos de acceso local pero no de permisos elevados.[30]

En el contexto de nuestro proyecto, donde buscamos evaluar la seguridad del sistema desde una cuenta sin privilegios de administrador, Lynis se presenta como una herramienta adecuada. Su ejecución en modo no privilegiado nos permite obtener

una visión general del estado de seguridad del sistema, identificando posibles áreas de mejora sin comprometer la integridad del mismo. Además, Lynis genera informes detallados que podemos utilizar para planificar acciones de fortalecimiento de la seguridad.[64, 116]

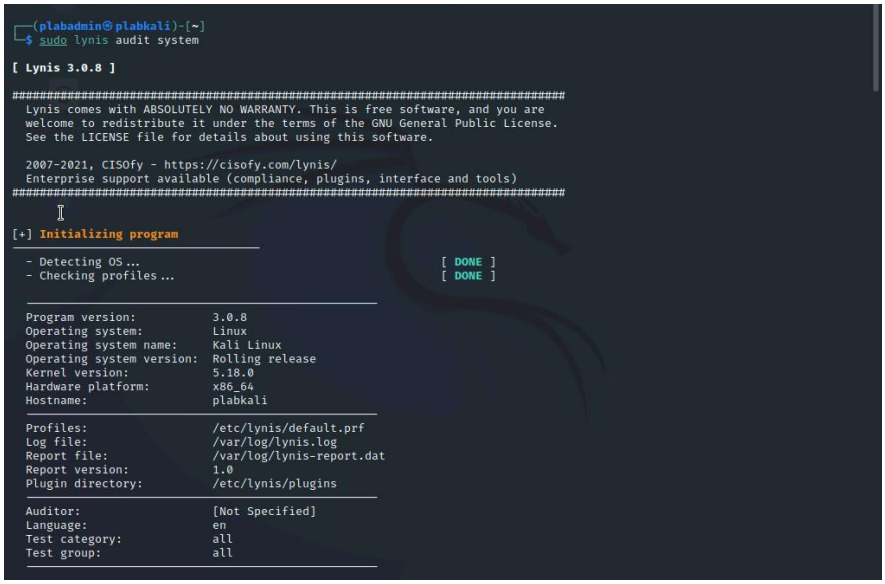


Figura 4.1. *Interfaz Lynis*

4.0.3 Análisis de Vulnerabilidades

4.0.3.1 Legion

Legion es una herramienta de código abierto para pruebas de penetración en redes, caracterizada por su interfaz gráfica intuitiva y su capacidad para realizar tareas de reconocimiento, escaneo y explotación de sistemas de información de manera semi-automatizada. Basada en *Sparta*, Legion integra múltiples herramientas como *Nmap, Nikto, Hydra, SMBenum, DirBuster, SSLyze, Vulners* y *WhatWeb*, permitiéndonos ejecutar más de 100 scripts programados automáticamente para facilitar la detección de vectores de ataque en hosts. [52]

En el contexto de nuestro proyecto, utilizamos Legion por su capacidad para realizar escaneos de puertos y servicios similares a *Nmap*, proporcionando una visión detallada de los servicios activos en las máquinas objetivo. Además, nos permite ejecutar acciones específicas sobre los puertos abiertos y los servicios que ofrecen las máquinas, facilitando la identificación de posibles vulnerabilidades y de la estructura web. La integración con *Nikto* nos permite realizar análisis de seguridad en servidores web, identificando configuraciones incorrectas y vulnerabilidades conocidas. Asimismo, Legion detecta automáticamente *CVEs* (Vulnerabilidades Comunes y Expuestas), vinculando las vulnerabilidades encontradas con exploits disponibles en bases de datos como Exploit-DB, lo que resulta esencial para realizar nuestras auditorías de seguridad de forma efectiva. [116]

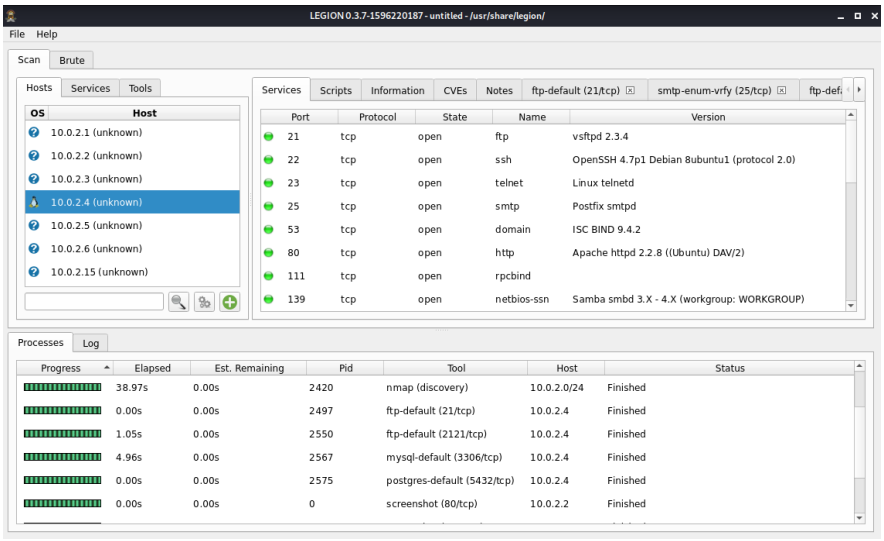


Figura 4.2. *Interfaz Legion*

4.0.4 Auditoría de redes inalámbricas

4.0.4.1 Airodump-ng

Airodump-ng es una herramienta de la suite Aircrack-ng utilizada para la captura de paquetes en redes inalámbricas 802.11. Permite escanear redes Wi-Fi disponibles, identificando puntos de acceso, clientes conectados, canales utilizados, niveles de señal y tipos de cifrado empleados. Su funcionalidad es esencial para la auditoría de redes con cifrado WEP, ya que facilita la recolección de vectores de inicialización (IVs) necesarios para el posterior descifrado de la clave mediante ataques estadísticos. [2, 3]

En el contexto de este proyecto, empleamos Airodump-ng para escanear las redes disponibles, identificar aquellas que utilizan cifrado WEP y capturar los datos necesarios para evaluar su seguridad. Esta herramienta es fundamental para realizar un análisis exhaustivo de la seguridad en redes inalámbricas, permitiéndonos detectar vulnerabilidades y proponer medidas correctivas adecuadas.[116]

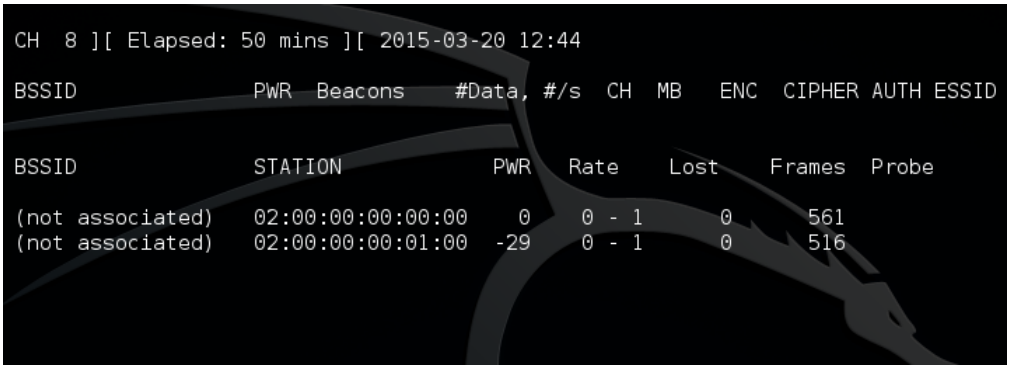


Figura 4.3. *Interfaz Airodump-ng*

4.0.4.2 Aireplay-ng

Aireplay-ng es una herramienta de la suite Aircrack-ng diseñada para la inyección de paquetes en redes inalámbricas 802.11, permitiendo la ejecución de diversos ataques con el fin de evaluar la seguridad de las mismas. Su funcionalidad principal es generar tráfico que puede ser utilizado posteriormente para descifrar claves WEP y WPA-PSK. Entre los ataques que se pueden realizar con Aireplay-ng se encuentran la desautenticación de clientes para capturar handshakes WPA, la autenticación falsa, el reenvío interactivo de paquetes y la reinyección automática de peticiones ARP [4].

En el contexto de este proyecto, empleamos Aireplay-ng tras la identificación de una red objetivo mediante Airodump-ng, lo que nos permite ejecutar ataques específicos que facilitan la captura de información crítica para la auditoría de seguridad. Esta herramienta es esencial para simular escenarios de ataque y evaluar la robustez de las medidas de seguridad implementadas en redes inalámbricas.[116]

4.0.4.3 Aircrack-ng

Aircrack-ng es una suite de herramientas de código abierto diseñada para auditar la seguridad de redes inalámbricas, especializada en la recuperación de claves WEP y WPA/WPA2-PSK mediante técnicas de análisis estadístico y ataques de diccionario. [3]

En el contexto de este proyecto, utilizamos Aircrack-ng como herramienta principal para descifrar claves WEP y WPA, empleando los paquetes capturados previamente con *Airodump-ng* y generados mediante *Aireplay-ng*. Esta herramienta es esencial para evaluar la robustez de las redes inalámbricas auditadas, permitiéndonos identificar vulnerabilidades en los protocolos de cifrado y proponer medidas correctivas adecuadas.[116]

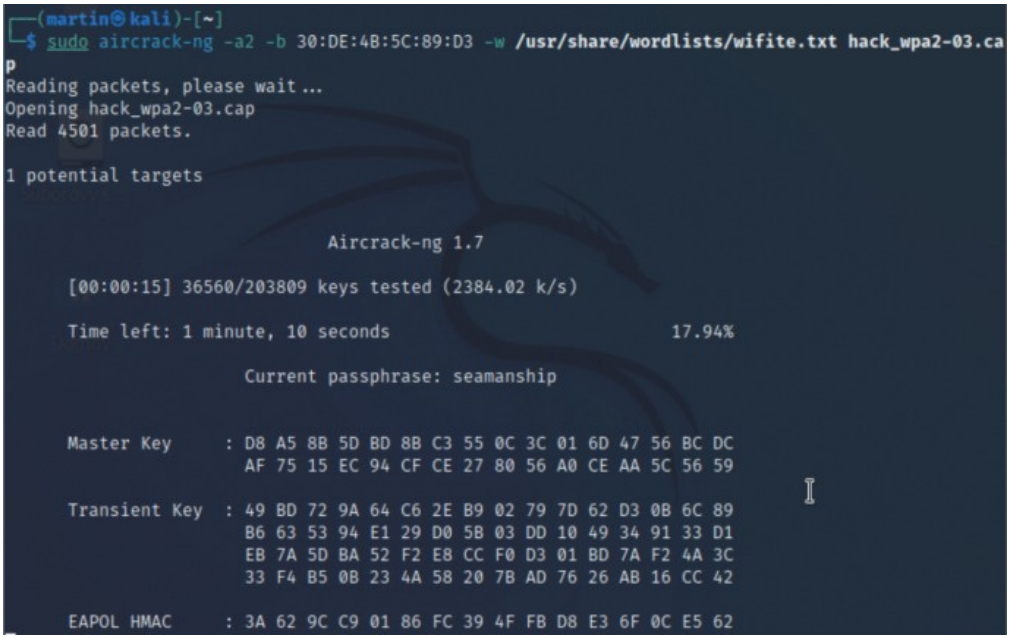


Figura 4.4. *Interfaz Aircrack-ng*

4.0.5 Ataques a contraseñas

4.0.5.1 Mimikatz

Mimikatz es una herramienta de código abierto desarrollada para la extracción de credenciales y análisis de autenticación en sistemas Windows. Nos permite recuperar contraseñas en texto claro, hashes de contraseñas, tickets Kerberos y otros datos sensibles directamente desde la memoria del sistema. Es compatible con versiones de Windows XP, Vista, 7, 8, 8.1 y 10, tanto en arquitecturas i386 como AMD64, pero no funciona en Windows 2000 debido a diferencias en la gestión de memoria y estructuras internas del sistema operativo [51].

En el contexto de este proyecto, empleamos Mimikatz para extraer credenciales de usuarios y tickets de Kerberos (TGT) desde sistemas Windows, facilitando la realización de ataques como "pass-the-ccache" y la evaluación de la seguridad en entornos de Active Directory. Aunque Mimikatz no está diseñado para el escalado de privilegios, asume que se ejecuta con privilegios elevados, lo que nos permite acceder a información crítica para la auditoría de seguridad. Además, su integración como payload en Metasploit lo convierte en una herramienta versátil y ampliamente utilizada en pruebas de penetración y evaluaciones de seguridad [9].

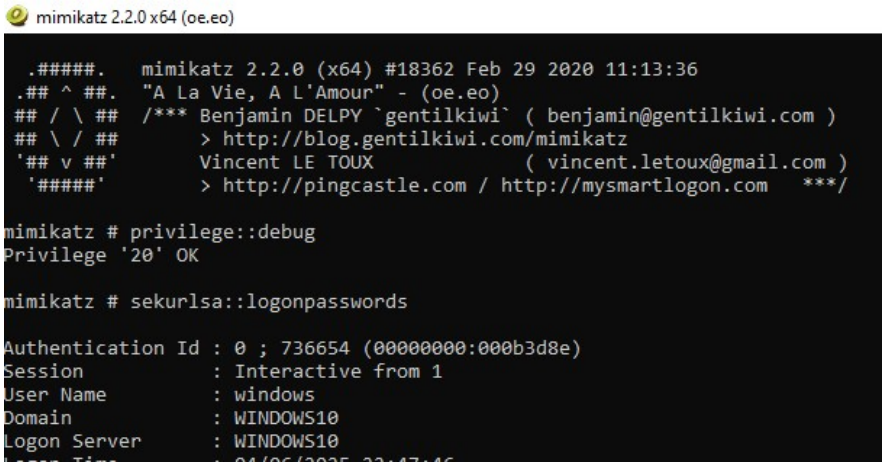


Figura 4.5. *Interfaz Mimikatz*

4.0.6 Auditorías a aplicaciones web

4.0.6.1 OWASP ZAP

OWASP ZAP (Zed Attack Proxy) es una herramienta de código abierto desarrollada por el proyecto OWASP, diseñada para realizar pruebas de seguridad en aplicaciones web. Ofrece capacidades de escaneo tanto pasivo como activo, permitiendo identificar vulnerabilidades comunes como inyecciones SQL, XSS y CSRF. ZAP funciona como un proxy inverso, interceptando y modificando el tráfico HTTP/HTTPS entre el cliente y el servidor, lo que facilita el análisis y la manipulación de las solicitudes y respuestas [142]. Además, incorpora técnicas de fuzzing, enviando datos malformados a la aplicación para detectar posibles fallos de seguridad [142]. La herramienta también permite una exploración manual mediante la activación del HUD (Heads-Up

Display), que proporciona una interfaz interactiva para realizar pruebas más dirigidas [127]. Finalmente, ZAP genera informes de auditoría detallados que recogen las vulnerabilidades detectadas junto con recomendaciones de mitigación, siendo esencial para evaluar la seguridad de las aplicaciones web auditadas.

En el contexto de este proyecto, utilizamos OWASP ZAP por su capacidad para realizar escaneos pasivos y activos automatizados, facilitando la identificación de vulnerabilidades sin necesidad de intervención manual constante. Su función de proxy inverso nos permite interceptar y modificar el tráfico entre el cliente y el servidor, lo que es crucial para analizar cómo la aplicación maneja diferentes tipos de datos. La técnica de fuzzing integrada en ZAP nos permite alimentar la aplicación con datos malformados o inesperados, ayudando a descubrir fallos de seguridad que podrían no ser evidentes con pruebas convencionales. Además, la posibilidad de realizar una exploración manual activando el HUD proporciona una interfaz interactiva que facilita pruebas más detalladas y específicas. La generación de informes de auditoría detallados nos permite documentar las vulnerabilidades encontradas y las recomendaciones para su mitigación, lo que es fundamental para mejorar la seguridad de la aplicación evaluada.[116]

Figura 4.6. *Interfaz OWASP ZAP*

4.0.6.2 Nikto

Nikto es una herramienta de código abierto utilizada para realizar auditorías de seguridad en aplicaciones web mediante el análisis exhaustivo de servidores web. Capaz de detectar más de 6.700 vulnerabilidades conocidas, Nikto examina configuraciones inseguras, archivos y scripts peligrosos, versiones obsoletas de software y otros problemas de seguridad en servidores como Apache, Nginx o Lighttpd [55]. Su capacidad para generar informes detallados facilita la documentación de hallazgos y la planificación de medidas correctivas [1].

En el contexto de este proyecto, empleamos Nikto para analizar el sistema del servidor, identificando posibles debilidades

que podrían ser explotadas por atacantes.[116]

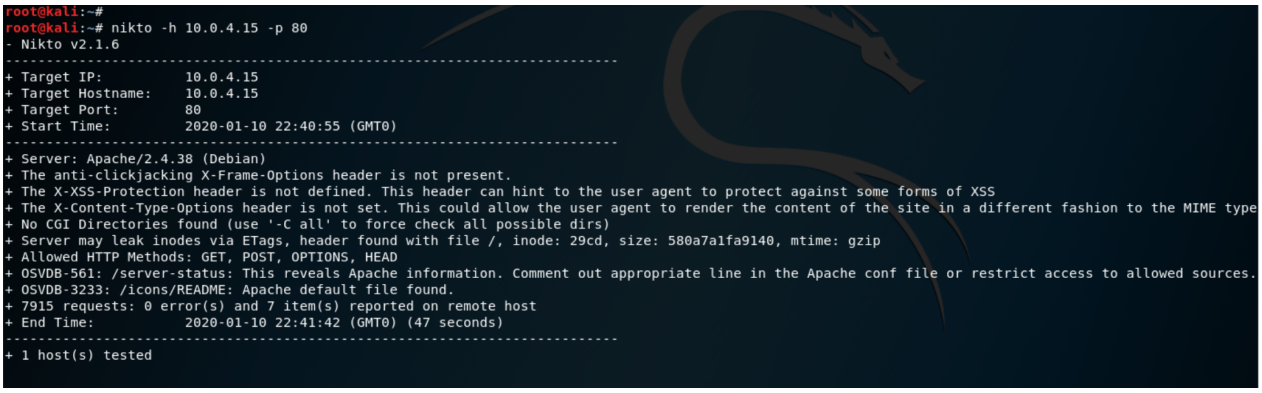


Figura 4.7. *Interfaz Nikto*

4.0.6.3 Wapiti

Wapiti es una herramienta de código abierto diseñada para realizar auditorías de seguridad en aplicaciones web mediante pruebas de caja negra, sin necesidad de acceder al código fuente. Su funcionalidad principal radica en la detección de diversas vulnerabilidades comunes en aplicaciones web, incluyendo Cross-Site Scripting (XSS), inyecciones SQL, inyecciones LDAP, errores en la gestión de ficheros, detección de ejecución de comandos e inyecciones CRLF [46].

En el contexto de este proyecto, empleamos Wapiti para analizar aplicaciones web en busca de estas vulnerabilidades, proporcionando informes detallados que facilitan la identificación y corrección de posibles fallos de seguridad. Esta herramienta es esencial para evaluar la robustez de las aplicaciones web auditadas y proponer medidas correctivas adecuadas.[116]

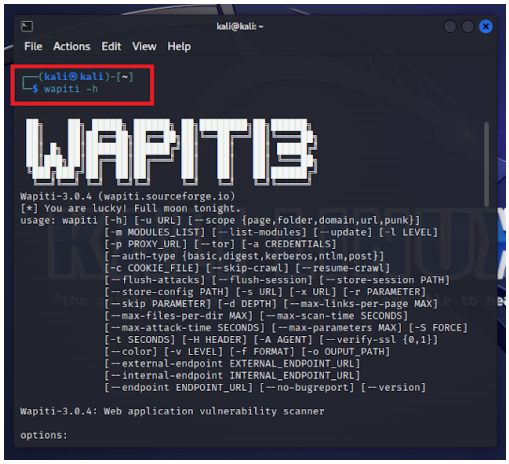


Figura 4.8. *Interfaz Wapiti*

4.0.6.4 WPScan

WPScan es una herramienta de código abierto especializada en la auditoría de seguridad de sitios web basados en WordPress. Permite identificar vulnerabilidades en el núcleo de WordPress, así como en plugins y temas instalados, mediante escaneos automatizados que detectan configuraciones inseguras, versiones obsoletas y otros problemas de seguridad [149]. WPScan también ofrece funcionalidades como la enumeración de usuarios, la detección de archivos de configuración expuestos y la verificación de contraseñas débiles [117].

En el contexto de este proyecto, empleamos WPScan para analizar la seguridad de sitios WordPress, proporcionando informes detallados que facilitan la identificación y corrección de posibles fallos de seguridad.[116]

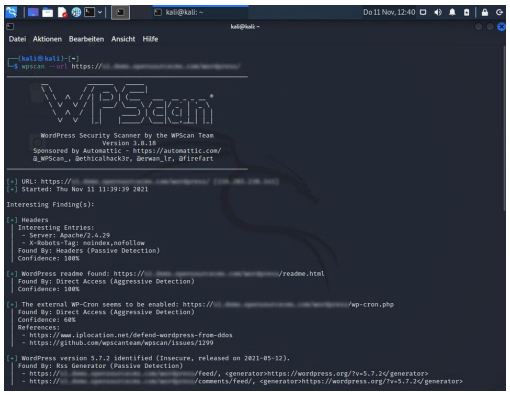


Figura 4.9. *Interfaz WPScan*

4.0.6.5 Sqlmap

sqlmap es una herramienta de código abierto especializada en la detección y explotación automatizada de vulnerabilidades de *SQL Injection*. Automatiza el proceso de identificación de puntos inyectables y proporciona un motor de detección potente que permite desde el fingerprinting de la base de datos hasta la extracción de datos o la ejecución de comandos en el sistema operativo cuando es posible.

Figura 4.10. *Interfaz sqlmap*

Entre sus funcionalidades más relevantes se encuentran la detección automática de técnicas de inyección (error-based, boolean-based blind, time-based, UNION, stacked queries, etc.), la enumeración de bases de datos/tablas/columnas, el volcado de filas/-credenciales (con soporte para cracking de hashes) y el uso de `tamper` scripts para intentar evadir filtros o WAFs. Además,

sqlmap ofrece multitud de parámetros para ajustar el nivel/riesgo de las pruebas, gestionar sesiones, y especificar payloads y técnicas. [116]

En el contexto de este proyecto, sólo se obtuvo autorización para realizar pruebas sobre la base de datos de la página web objeto de auditoría. Se empleó sqlmap únicamente sobre ese objetivo autorizado y, tras las pruebas automatizadas y verificación manual de los resultados obtenidos, no se identificaron vulnerabilidades explotables por *SQL Injection* en la aplicación auditada.[116]

5

Fase de Análisis Básico de Vulnerabilidades

5.1 Legion

En esta sección, realizamos primeramente un escaneo general de toda la red para detectar a los hosts. Para ello consta de seis fases de escaneo **nmap** donde perseguimos escanear los puertos activos; una vez detectados procedimos a desglosar en diferentes tipos de dispositivos para descubrir las vulnerabilidades concretas de cada tipo de dispositivo, teniendo en cuenta que los dispositivos de un mismo ámbito no implican que tengan los mismos puertos abiertos, por lo que conlleva realizar distintos escaneos de vulnerabilidades de distinta forma. En este escaneo se detalla que la ip usada para auditar es **192.168.1.215**, por lo que no la tenemos en cuenta.

5.1.0.1 Fase 1: Detección de hosts y escaneo de puertos 80,81,443,4443,8080,8081,8082

Realizamos en la primera fase el siguiente escaneo, buscando encontrar qué hosts están activos y qué puertos tienen abiertos para posteriormente encontrar vulnerabilidades en estos, ya que son los puertos más usados por los dispositivos.

- **Puerto 80 (TCP)** – HTTP: tráfico web sin cifrar; muy utilizado y generalmente redirige a HTTPS.
- **Puerto 81 (TCP)** – HTTP alternativo: usado ocasionalmente para servidores o proxies web secundarios.
- **Puerto 443 (TCP)** – HTTPS: tráfico web cifrado mediante TLS/SSL, estándar seguro para servidores web.
- **Puerto 4443 (TCP)** – HTTPS alternativo: registrado como "pharos", a menudo usado para interfaces seguras o portales VPN.
- **Puerto 8080 (TCP)** – HTTP alternativo: frecuentemente servidores web, proxies o Tomcat en entornos no root.
- **Puerto 8081 (TCP)** – HTTP de administración: común en paneles de control internos (Tomcat, appliances).

- **Puerto 8082 (TCP)** – HTTP/HTTPS interno: similar al 8081, usado por herramientas web administrativas o proxies.

El código ejecutado fue el siguiente:

/usr/bin/nmap -T4 -sV -p T:80,81,443,4443,8080,8081,8082 -oA /root/.local/share/legion/tmp/legion-m8v80fcx-running/nmap/202506101642
nmapstage1 192.168.1.0/24

Dando como resultado el primer informe que utilizaremos para recopilar información acerca de qué tipo de dispositivo es y qué puertos de los mencionados están activos. El informe obtenido aparece en el apéndice.

5.1.0.2 Fase 2: Escaneo de puertos tcp y udp

En esta fase, a diferencia de la anterior, nos centramos en buscar puertos tanto tcp como udp y fueron los siguientes:

- **Puerto 25 (TCP)** – SMTP: transferencia de correo entre servidores; a menudo bloqueado por ISP para prevenir spam :contentReference[oaicite:1]index=1.
- **Puerto 135 (TCP/UDP)** – Microsoft DCE/RPC Endpoint Mapper: mapea servicios RPC en Windows (también llamado RPC Locator) :contentReference[oaicite:2]index=2.
- **Puerto 137 (UDP)** – NetBIOS Name Service: resolución de nombres en redes Windows :contentReference[oaicite:3]index=3.
- **Puerto 139 (TCP/UDP)** – NetBIOS Session Service: compartición de archivos e impresoras legacy en Windows :contentReference[oaicite:4]index=4.
- **Puerto 445 (TCP/UDP)** – Microsoft-DS SMB: compartición moderna de archivos y servicios de directorio en Windows :contentReference[oaicite:5]index=5.
- **Puerto 1433 (TCP)** – Microsoft SQL Server: acceso al motor de bases de datos SQL Server.
- **Puerto 3306 (TCP)** – MySQL: acceso al servidor de bases de datos MySQL/MariaDB.
- **Puerto 5432 (TCP)** – PostgreSQL: acceso a la base de datos PostgreSQL.
- **Puerto 161 (UDP)** – SNMP (agent): protocolo para monitorizar y gestionar dispositivos de red.
- **Puerto 162 (UDP)** – SNMP TRAP: recepción de alertas SNMP desde agentes.
- **Puerto 1434 (UDP)** – Microsoft SQL Server Browser Service: respuesta a consultas de cliente para instancias SQL Server (puerto dinámico) en versiones anteriores.

Nos informamos de cuáles son accesibles y en qué modo se hallan dichos puertos; esos puertos seleccionados son los más utilizados en tcp y udp. El código ejecutado fue el siguiente:

/usr/bin/nmap -T4 -sV -p T:25,135,137,139,445,1433,3306,5432,U:137,161,162,1434 -oA /root/.local/share/legion/tmp/legion-m8v80fcx-running/nmap/20250610164525500468-nmapstage2 192.168.1.0/24

Nuevamente obtuvimos el siguiente informe, donde aparece desglosada la información del estado de los puertos de cada host. El informe obtenido aparece en el apéndice.

5.1.0.3 Fase 3: Escaneo de vulnerabilidades general

Una vez recopilada la información de los puertos principales procedimos a realizar un escaneo de vulnerabilidades con un script denominado "vulners" que trata de comprobar las vulnerabilidades más comunes en los puertos mencionados anteriormente.

El código usado fue el siguiente:

/usr/bin/nmap -sV –script=vulners -vvvv -oA /root/.local/share/legion/tmp/legion-m8v80fcx-running/nmap/20250610164540166709-nmapstage3 192.168.1.0/24

Vemos cómo van apareciendo en el informe las primeras vulnerabilidades, que posteriormente, derivado de este informe, seccionaremos en función del dispositivo y puertos. El informe obtenido aparece en el apéndice.

5.1.0.4 Fase 4: Escaneo de tcp y udp

En esta fase, a diferencia de la fase 2, nuestro objetivo es detectar estos servicios:

- Puertos de transferencia de archivos y acceso remoto: 21 (FTP), 22 (SSH), 23 (Telnet).
- Correo electrónico: 110 (POP3).
- RPC/NFS para sistemas Unix/Linux: 111 y 2049.
- Acceso remoto gráfico en Windows: 3389 (RDP).
- Servidores web no root: 8080.
- Seguridad/VPN: 500 (IPsec IKE).
- Comunicación VoIP: 5060 (SIP).

El código usado fue el siguiente:

/usr/bin/nmap -T4 -sV -p T:23,21,22,110,111,2049,3389,8080,U:500,5060 -oA /root/.local/share/legion/tmp/legion-m8v80fcx-running/nmap/20250610165214124133-nmapstage4 192.168.1.0/24

De la misma forma vemos cómo se van detectando, a medida que avanzan las fases, más puertos con sus servicios correspondientes. El informe obtenido aparece en el apéndice.

5.1.0.5 Fase 5: Escaneo de tcp y udp

En esta fase tratamos de investigar acerca de los siguientes servicios, además de acotar casi todo el espectro de puertos comúnmente cargados.

- **Puerto 25 (TCP) – SMTP:** Protocolo utilizado por servidores de correo para envío/salida. Usado en spam y ataques de relay abiertos; normalmente se bloquea desde redes residenciales.
- **Puerto 135 (TCP/UDP) – Microsoft DCE/RPC Endpoint Mapper:** Punto inicial de los servicios RPC en Windows. Un objetivo común para elevación de privilegios y ataques de servicios como Exchange o DCOM.
- **Puerto 137 (UDP) – NetBIOS Name Service:** Resolución de nombres en redes Windows legacy. Permite descubrimiento de nombres que puede filtrar información sobre hosts :contentReference[oaicite:3]index=3.
- **Puerto 138 (UDP) – NetBIOS Datagram Service:** Distribución de información NetBIOS sin conexión, usado para alertas y navegación por red.
- **Puerto 139 (TCP/UDP) – NetBIOS Session Service:** Servicio legacy de SMB/CIFS, puede permitir acceso a recursos compartidos en versiones antiguas de Windows.
- **Puerto 445 (TCP/UDP) – Microsoft-DS (SMB moderno):** Comparable a 139, pero para SMB sobre TCP. Muy crítico (EternalBlue, WannaCry, etc.).
- **Puerto 1433 (TCP) – Microsoft SQL Server:** Puerto por defecto para conexión a SQL Server. Target frecuente para ataques de fuerza bruta e inyección SQL.
- **Puerto 3306 (TCP) – MySQL/MariaDB:** Puerto estándar para clientes de MySQL; expuesto en bases de datos mal configuradas o accesibles desde Internet.
- **Puerto 5432 (TCP) – PostgreSQL:** Puerto por defecto de Postgres; similar al anterior en cuanto a riesgos de configuración errónea.
- **Puerto 161 (UDP) – SNMP agent:** Protocolo de gestión de red, permite lectura (y a veces escritura) de configuraciones/router tables mediante community strings.
- **Puerto 162 (UDP) – SNMP Trap:** Recibe alertas automáticas de dispositivos SNMP; lógico apuntar aquí en auditorías de red.
- **Puerto 1434 (UDP) – SQL Server Browser Service:** Responde con lista de instancias SQL y puertos asociados; si está accesible, facilita descubrimiento de instancias vulnerables.

El código usado fue el siguiente:

/usr/bin/nmap -T4 -sV -p T:0-20,24,26-79,81-109,112-134,136,138,140-442,444,446-1432,1434-2048,2050-3305,3307-3388,3390-5431,5433-8079,8081-29999 -oA /root/.local/share/legion/tmp/legion-m8v80fcx-running/nmap/20250610165240388228-nmapstage5 192.168.1.0/24

En esta fase el proceso es más largo de ejecutar al ser más puertos los que precisan de escaneo. El informe obtenido aparece en el apéndice.

5.1.0.6 Fase 6: Escaneo de Servicios personalizados y backdoors

En esta fase tratamos de investigar acerca de los siguientes servicios: - Servicios y servidores personalizados: aplicaciones internas o microservicios que se configuran en estos puertos. - Aplicaciones distribuidas que escuchan en high-ports (ej. servicios basados en HTTP, APIs, herramientas internas). - Conexiones invertidas o backdoors que suelen usar puertos no estándar para evitar detección.

- **TCP 30000–49151**: puertos registrados, usados por aplicaciones específicas.
- **TCP 49152–65535**: puertos dinámicos/efímeros asignados automáticamente.
- **Objetivo del escaneo**: detectar servicios personalizados, backdoors o software interno.

El código usado fue el siguiente:

/usr/bin/nmap -T4 -sV -p T:30000-65535 -oA /root/.local/share/legion/tmp/legion-m8v80fcx running/nmap/20250610174019651139-nmapstage6 192.168.1.0/24

En esta fase el proceso es más largo de ejecutar, al ser más puertos los que precisan de escaneo y sin saber realmente qué servicio se está ejecutando. El informe obtenido aparece en el apéndice.

5.1.1 Conclusiones de la fase de análisis básico

En estas fases hemos realizado un análisis básico de vulnerabilidades, donde hemos detectado los hosts activos, los puertos abiertos y los servicios que se ejecutan en ellos. A partir de esta información, hemos generado informes detallados que servirán como base para las siguientes fases del pentesting. Para ello realizaremos escaneos *nmap* concretos diferenciando en tipo de sistema operativo, puertos abiertos y servicios que se ejecutan en ellos. Estos informes los utilizaremos para identificar vulnerabilidades específicas en cada dispositivo y servicio, permitiendo así una auditoría más precisa y efectiva.

5.1.2 Escaneo de Puertos Activos de las Impresoras y Escaners

En esta parte, realizamos primeramente un análisis de las impresoras detectando qué servicios tenían activos, y cómo detectar sus vulnerabilidades de forma más concreta no general como en la fase 3. Para ello diferenciamos en función del modelo de la impresora, ya que cada modelo tiene unos puertos activos diferentes y unos servicios diferentes.

5.1.2.1 Impresora HL-L5100DN

En esta impresora *Brother* detectamos que este host tenía activo los puertos 80,443,631,515,9100 y que tenían los servicios http, printer y jetdirect.

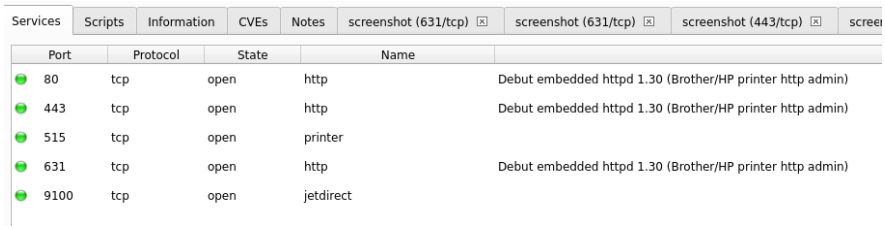


***Figura 5.1**. Puertos Activos de la impresora HL-L5100DN*

5.1.2.2 Impresora HL-L5100DN Modelo 2, HL-L5210-DN, HL5200DW

En estas impresoras *Brother* detectamos que este host tenía activo los puertos 80,443,631,515,9100 y concretamente los servicios http, https, printer, ipp y jetdirect.

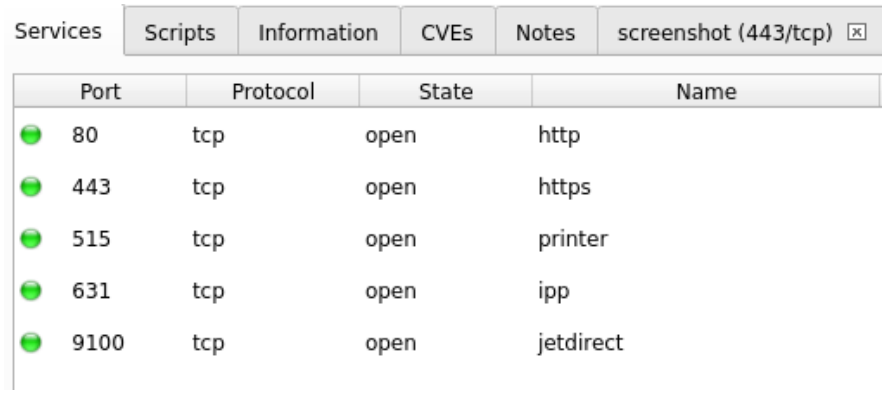


***Figura 5.2**. Puertos Activos de la impresora HL-L5100DN Modelo 2*

5.1.2.3 Impresora HP Laserjet M210DW

En esta impresora *HP* detectamos que tenía activo los puertos 80,443,631,515,9100 y concretamente los servicios ftp, http, telnet, tcpwrapped, soap y jetdirect.

***Figura 5.3**. Puertos Activos de la impresora HP LASERJET M210DW*

5.1.2.4 Impresora RICOH MP C401SR

En esta impresora *RICOH* detectamos que tenía activo los puertos 21, 23, 80, 139, 514, 515, 631, 3702, 7444, 9100, 10021, 18315 y que concretamente ofrecía los servicios ftp, telnet, ipp, tcpwrapped, shell, printer, ws-discovery, unknown y jetdirect.

Figura 5.4. *Puertos Activos de la impresora RICOH MP C401SR*

5.1.2.5 Impresora RICOH MP C401SR Modelo 2

En esta impresora *RICOH* detectamos que tenía activo los puertos 21, 23, 80, 139, 514, 515, 631, 3702, 7444,8080, 9100, 10021, 18315, 65000 y que concretamente ofrecía los servicios ftp, telnet, ipp, tcpwrapped, shell, printer, ws-discovery, unknown, http y jetdirect.

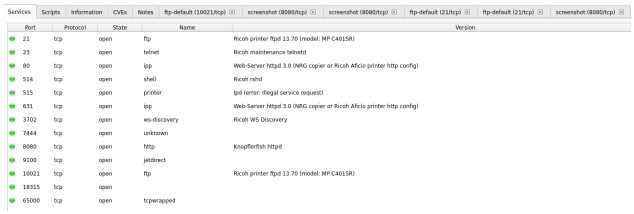


Figura 5.5. *Puertos Activos de la impresora RICOH MP C401SR Modelo 2*

5.1.2.6 Escaner PFU

En este escaner *PFU* detectamos que tenía activo los puertos 80, 53218,53219 y que concretamente ofrecía los servicios lighttpd y unknown.

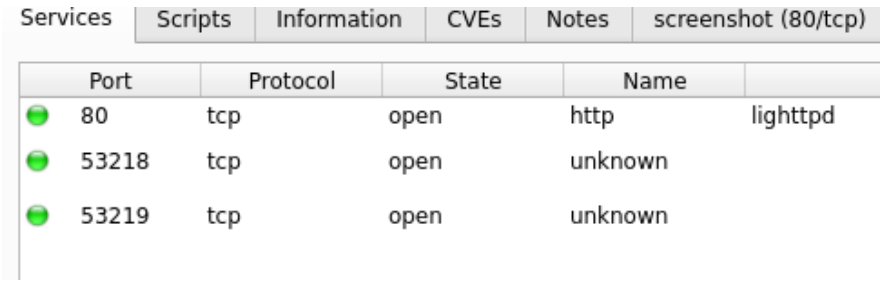


Figura 5.6. *Puertos Activos del escaner PFU*

5.1.2.7 Escaner PFU Modelo 2

En este escaner *PFU* detectamos que tenía activo los puertos 80, 53218,53219 y que concretamente ofrecía los servicios http y unknown.

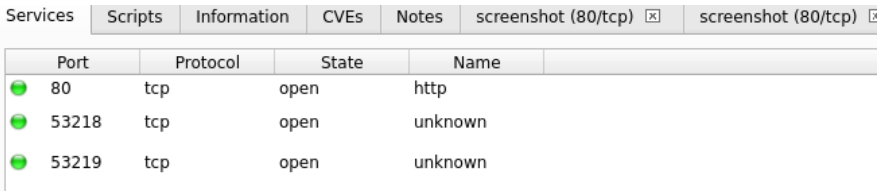


***Figura 5.7**. Puertos Activos del escaner PFU*

5.1.3 Scripts usados para las impresoras

5.1.3.1 Impresora HL-L5100DN, Impresora HL-L5100DN Modelo 2, HL-L5210-DN, HL5200DW, Escaner PFU

Puerto	Servicio	Script(s)	Objetivo
80, 443	HTTP/HTTPS	vuln, vulners, http-vuln-cve*, http-robots.txt.nse, http-enum.nse, http-methods.nse, http-auth.nse	Detectar CVE, rutas ocultas, directorios, métodos inseguros, autenticación
515, 631	CUPS	cups-info.nse, cups-queue-info.nse	Información de impresora, colas de impresión
9100	JetDirect	pjl-ready-message.nse	Interrogar mensajes PJL, evaluar controles de acceso
53218,53219	unknown	–	Puertos desconocidos, no se ejecutan scripts NSE.

***Tabla 5.1**. Scripts usados por servicio y objetivo*

5.1.3.2 Impresora HP Laserjet M210DW

Puerto	Servicio	Script(s)	Objetivo
21	TCP/FTP	ftp-anon.nse	Detecta acceso anónimo o listado de directorios públicos en servidores FTP.
80, 443, 8080, 631, 3910	HTTP, IPP, SOAP (gSOAP)	vuln, vulners, http-vuln-cve*, http-robots.txt.nse, http-enum.nse, http-methods.nse, http-auth.nse, http-apache-negotiation.nse	Detectan CVEs, rutas ocultas, recursos y métodos HTTP. Evaluan autenticación y contenido. Útiles ante interfaces web SOAP expuestas.
515, 631	CUPS/IPP	cups-info.nse, cups-queue-info.nse	Obtienen configuración y colas de impresión, detectando servicios expuestos o inseguros.
9100	JetDirect	pjl-ready-message.nse	Evalúa exposición de impresoras JetDirect mediante comandos PJL.

Tabla 5.2. *Scripts usados por servicio y objetivo*

5.1.3.3 Impresora RICOH MP C401SR

Puerto	Servicio	Script(s)	Objetivo
21, 10021	FTP	ftp-anon.nse	Detecta acceso anónimo vía FTP con credenciales por defecto (CVE-2019-14309).
80, 631, 3702, 7444	HTTP / IPP	vuln, vulners, http-vuln-cve*, http-robots.txt.nse, http-enum.nse, http-methods.nse, http-auth.nse, http-apache-negotiation.nse	Detecta CVEs (XSS, overflows), rutas ocultas, métodos inseguros, autenticación débil y configuración HTTP expuesta.
515	CUPS / LPD	cups-info.nse, cups-queue-info.nse	Extrae datos de configuración y colas. Detecta posibles vulnerabilidades (ej. buffer overflow en LPD).
9100	JetDirect	pjl-ready-message.nse	Lee o modifica comandos PJL; útil para evaluar acceso no autorizado en impresoras.

Tabla 5.3. *Scripts usados por servicio y objetivo*

5.1.3.4 Impresora RICOH MP C401SR Modelo 2 Parte 1

5.1.3.5 Impresora RICOH MP C401SR Modelo 2 Parte 2

5.1.3.6 Escaner PFU Modelo 2

Puerto	Servicio	Script(s)	Objetivo
21	tcp/ftp	ftp-anon.nse	Detectar acceso FTP anónimo, listar directorios y posibles permisos de escritura.
23	tcp/telnet (maintenance)	–	Puerto de mantenimiento. No se aplica script específico, se evalúa exposición del servicio.
80, 631, 7444, 8080, 3702	HTTP / IPP / SOAP	vuln, vulners, http-vuln-cve*, http-robots.txt.nse, http-enum.nse, http-methods.nse, http-auth.nse, http-apache-negotiation.nse	Detectar CVEs (ej. CVE-2024-47939), rutas ocultas, métodos HTTP inseguros, autenticación débil y posibles interfaces SOAP expuestas.

Tabla 5.4. *Scripts usados por servicio y objetivo*

Puerto	Servicio	Script(s)	Objetivo
514, 515, 631	CUPS / IPP / LPD	cups-info.nse, cups-queue-info.nse	Obtener configuración de impresoras y colas, útil para detectar fallos en red de impresión.
9100	JetDirect	pjl-ready-message.nse	Leer/modificar comandos PJL en impresoras HP/Ricoh. Riesgo de exposición de datos.
10021	tcp/ftp	ftp-anon.nse	Igual que el puerto 21: acceso anónimo como FTP secundario o backup.
18315, 65000	tcpwrapped	–	Puertos protegidos (tcpwrapped), no se ejecutan scripts NSE.

Tabla 5.5. *Scripts usados por servicio y objetivo*

Puerto	Servicio	Script(s)	Objetivo
80	HTTP	vuln, vulners, http-vuln-cve*, http-robots.txt, http-enum, http-methods, http-put, http-auth, http-default-accounts, http-auth-finder, http-form-fuzzer	Detectar vulnerabilidades conocidas (CVEs), rutas ocultas, métodos HTTP inseguros, formularios web mal configurados y credenciales por defecto.
53218	Desconocido	–	Puerto abierto sin servicio identificado; no se aplicaron scripts NSE.
53219	Desconocido	–	Igual que el anterior; puerto abierto pero sin servicio conocido ni scripts aplicables.

Tabla 5.6. *Scripts usados por servicio y objetivo*

5.1.4 Escaneo de Puertos Activos de las Cámaras de Videovigilancia

En esta parte, realizamos primeramente un análisis de las cámaras detectando qué servicios tenían activos, y cómo detectar sus vulnerabilidades de forma más concreta no general como en la fase 3. Para ello diferenciamos en función del modelo de la cámara; solo detectamos un único modelo por lo que procedimos a realizar en una sola cámara la prueba de vulnerabilidades.

5.1.4.1 Cámara de videovigilancia

En la cámara *Grandstream Network* detectamos que este host tenía activos los puertos 22 y 80 que ofrecían los servicios dropbear y minihttpd.

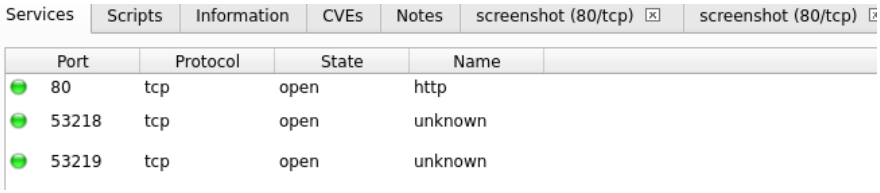


Figura 5.8. *Puertos Activos de la Camara de Videovigilancia*

5.1.5 Scripts usados para la Cámara de Videovigilancia

Puerto	Servicio	Script(s)	Objetivo principal
22	SSH	sshv1, ssh2-enum-algos, ssh-hostkey, ssh-auth-methods	Identificar versión del protocolo SSH, algoritmos de cifrado, clave pública del host y métodos de autenticación habilitados.
80	HTTP	vuln, vulners, http-vuln-cve*, http-robots.txt, http-enum, http-methods, http-auth-finder, http-passwd, http-frontpage-login, http-title, http-server-header, http-config-backup, http-virustotal	Detectar CVEs en servicios web, analizar encabezados HTTP, enumerar directorios y métodos, detectar archivos sensibles (passwd, backup), login en FrontPage y posibles exposiciones indexadas en VirusTotal.

Tabla 5.7. *Scripts usados por servicio y objetivo*

5.1.6 Escaneo de Puertos Activos de dispositivos de Red

En esta parte, realizamos primeramente un análisis de los dispositivos de red detectando qué servicios tenían activos, y cómo detectar sus vulnerabilidades de forma más concreta no general como en la fase 3.

5.1.6.1 Hardware de Red Millenial

El host *Millenial Network* se trata de un hardware de red de tipo router o switch; detectamos que este host tenía activos los puertos 80, 554 y 23000 que ofrecían los servicios http, rtsp e inovaport1.

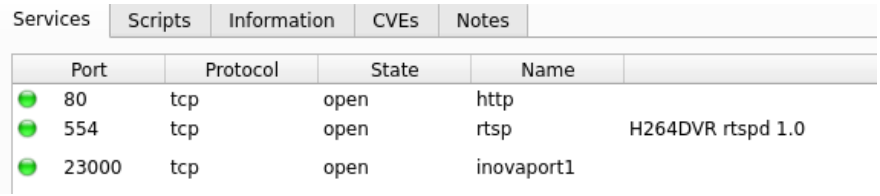


Figura 5.9. *Puertos Activos del hardware de red Millenial*

5.1.6.2 Hardware de Red TpLink

El host *TpLink* se trata de un hardware de red de tipo router o switch; detectamos que este host tenía activos los puertos 80, 135, 139, 445 y 5040, que ofrecían los servicios http, msrpc, netbios-ssn, microsoft-ds e unknown.

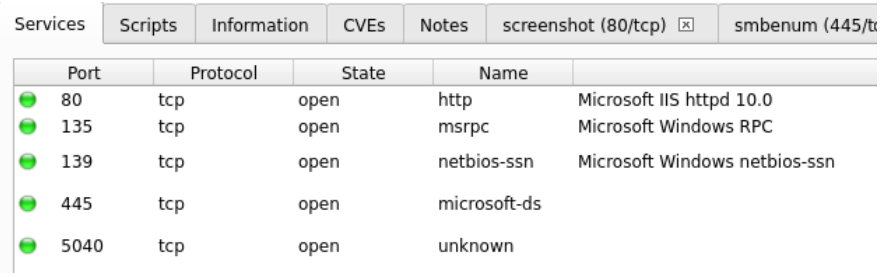


Figura 5.10. *Puertos Activos del hardware de red TpLink*

5.1.7 Scripts usados para el hardware de red

5.1.7.1 Millenial Network

Puerto	Servicio	Script(s)	Objetivo principal
22	SSH	sshv1, ssh2-enum-algos, ssh-hostkey, ssh-auth-methods	Identificamos versión del protocolo SSH, algoritmos de cifrado, clave pública del host y métodos de autenticación habilitados.
80	HTTP	vuln, vulners, http-vuln-cve*, http-robots.txt, http-enum, http-methods, http-auth-finder, http-passwd, http-frontpage-login, http-title, http-server-header, http-config-backup, http-virustotal	Detectamos CVEs en servicios web, analizamos encabezados HTTP, enumeramos directorios y métodos, y buscamos archivos sensibles (passwd, backup), login en FrontPage y posibles exposiciones indexadas en VirusTotal.

Tabla 5.8. *Scripts usados por servicio y objetivo*

5.1.7.2 TpLink

Puerto	Servicio	Script(s)	Objetivo principal
80	HTTP	vuln, vulners, http-vuln-cve*, http-robots.txt, http-enum, http-methods, http-tplink-dir-traversal.nse, http-virustotal	Detectamos vulnerabilidades web, rutas ocultas, métodos inseguros y directory traversal en routers TP-Link. Verificamos presencia en VirusTotal.
135	MSRPC	vuln, vulners	Identificamos CVEs relacionados con servicios RPC (como ejecución remota en DCOM o MSRPC).
139	NetBIOS-SSN	vuln, vulners	Escaneamos en busca de vulnerabilidades en servicios NetBIOS, como acceso no autenticado a recursos compartidos.
445	Microsoft-DS	vuln, vulners	Detectamos CVEs en servicios SMB, incluidas vulnerabilidades como EternalBlue o fuga de credenciales.
5040	unknown	vuln, vulners	Aplicamos escaneo de vulnerabilidades generales sobre puerto desconocido para evaluar posibles servicios expuestos.

Tabla 5.9. *Scripts NSE aplicados a puertos detectados en host TP-Link*

5.1.8 Escaneo de Puertos Activos de los Ordenadores

En esta parte, realizamos primeramente un análisis de los ordenadores detectando qué servicios tenían activos, y cómo detectar sus vulnerabilidades de forma más concreta, no general como en la fase 3. Para ello diferenciamos en función del modelo del ordenador; se detectaron varios modelos, por lo que procedimos a realizar en cada uno la prueba de vulnerabilidades.

5.1.8.1 Hewlett Packard – Integrated Lights-Out (iLO)

En el ordenador *Hewlett Packard – Integrated Lights-Out (iLO)* detectamos los siguientes puertos activos: 22, 80, 443, 17988 y 17990, que ofrecían los servicios ssh, http, https y unknown.

Port	Protocol	State	Name	
22	tcp	open	ssh	HP Integrated Lights-Out mpSSH 0.2.1 (protocol 2.0)
80	tcp	open	http	HP Integrated Lights-Out web interface
443	tcp	open	http	HP Integrated Lights-Out web interface
17988	tcp	open	unknown	
17990	tcp	open	unknown	

Figura 5.11. *Puertos Activos del Ordenador Hewlett Packard*

5.1.8.2 Hewlett Packard

En el ordenador *Hewlett Packard* detectamos que este host tenía activos los puertos 135, 139 y 445, que ofrecían los servicios microsoft-ds, netbios-ssn y msrpc.

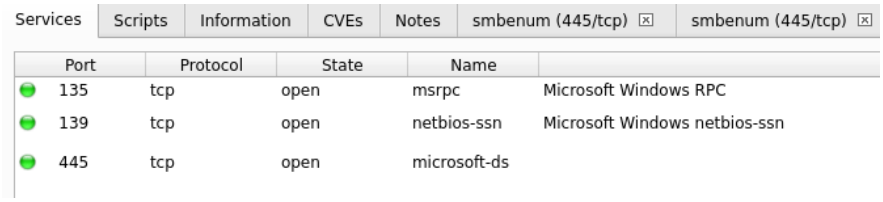


Figura 5.12. *Puertos Activos del Ordenador Hewlett Packard*

5.1.8.3 Hon Hai Precision

En el ordenador *Hon Hai Precision* detectamos que este host tenía activos los puertos 135, 139, 445, 5040 y 7680, que ofrecían los servicios microsoft-ds, netbios-ssn, msrpc y tcpwrapped.

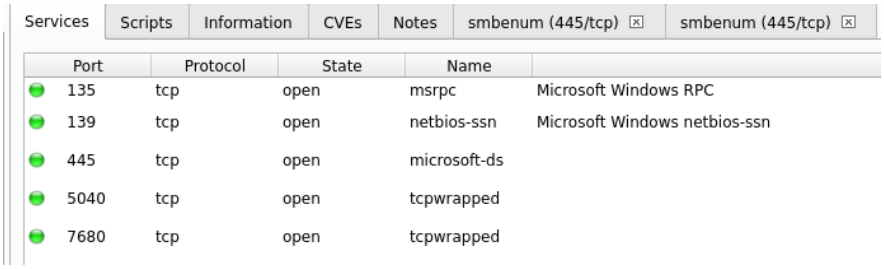


Figura 5.13. *Puertos Activos del Ordenador Hon Hai Precision*

5.1.9 Scripts usados para los Ordenadores

5.1.9.1 Hewlett Packard – Integrated Lights-Out (iLO)

5.1.9.2 Hewlett Packard

5.1.9.3 Hon Hai Precision

5.1.10 Escaneo de Puertos Activos de las IIS

Analizamos cada equipo con IIS activo, evaluando vulnerabilidades de forma concreta (no genérica como en la Fase 3). Al detectarse pocos perfiles de servidor, realizamos pruebas individualizadas (escaneos, revisión de configuraciones y versiones), documentando las vulnerabilidades encontradas en cada caso.

5.1.10.1 Hewlett Packard

En el servidor de sistema operativo *Hewlett Packard* detectamos activos los puertos 80, 135, 139, 445, 2103, 2107 y 5040, correspondientes a servicios como Microsoft IIS httpd 10.0, msrpc, netbios-ssn y microsoft-ds.

Puerto	Servicio	Script(s)	Objetivo
22	SSH (mpSSH 0.2.1)	ssh-hostkey, ssh2-enum-algos, ssh-brute, vuln, vulners	Enumeramos claves y algoritmos, y detectamos autenticación débil o vulnerabilidades conocidas.
80	HTTP (web interface iLO)	http-enum, http-title, http-robots.txt, http-headers, http-form-brute, http-vuln-cve*	Mapeamos la interfaz web (rutas y formularios), comprobamos cabeceras y buscamos CVEs web.
443	HTTPS (web interface iLO)	ssl-cert, ssl-enum-ciphers, http-enum, http-title, http-headers, http-vuln-cve*	Analizamos el certificado y los cifrados; mapeamos contenido seguro y buscamos CVEs aplicables.
17988, 17990	iLO virtual media / management (unknown)	banner, http-enum, ssl-cert (si aplica), vuln, vulners	Identificamos protocolo o servicio mediante banner o HTTP y buscamos vulnerabilidades específicas de iLO.

Tabla 5.10. *Scripts NSE aplicados a puertos y servicios detectados en la imagen*

Puerto	Servicio	Script(s)	Objetivo
135	MSRPC	vuln, vulners, http-vuln-cve*, http-vuln-*	Enumeramos y detectamos vulnerabilidades asociadas a DCOM y servicios RPC, incluyendo posibles CVEs críticos.
139	NetBIOS-SSN	smb-enum-shares, smb-enum-users, smb-vuln-ms08-067, smb-vuln-conficker, smb-vuln-cve2009-3103	Enumeramos usuarios y recursos compartidos; detectamos vulnerabilidades críticas como MS08-067, Conficker o acceso no autorizado.
445	Microsoft-DS	smb-vuln-ms17-010, smb-vuln-cve-2017-7494, smb-enum-shares, smb-enum-users, smb-vuln-conficker	Detectamos vulnerabilidades SMB explotables (EternalBlue, Samba RCE) y obtenemos información sensible compartida o cuentas de red.

Tabla 5.11. *Scripts NSE aplicados a puertos y servicios SMB/RPC*

Puerto	Servicio	Script(s)	Objetivo
135	MSRPC	vuln, vulners, http-vuln-cve*, http-vuln-*	Detectamos vulnerabilidades relacionadas con servicios RPC y exposición de interfaces DCOM.
139	NetBIOS-SSN	smb-enum-shares, smb-enum-users, smb-vuln-ms08-067, smb-vuln-conficker, smb-vuln-cve2009-3103	Enumeramos recursos compartidos y usuarios; detectamos vulnerabilidades críticas de SMB como Conficker y MS08-067.
445	Microsoft-DS	smb-vuln-ms17-010, smb-vuln-cve-2017-7494, smb-enum-shares, smb-enum-users	Exploramos posibles vulnerabilidades SMB modernas (EternalBlue, Samba RCE) y accesos a recursos de red.
5040, 7680	tcpwrapped	firewall-bypass	Identificamos puertos protegidos por cortafuegos mediante técnicas de evasión o análisis de filtrado.

Tabla 5.12. *Scripts NSE aplicados a servicios SMB/RPC y detección de filtrado*

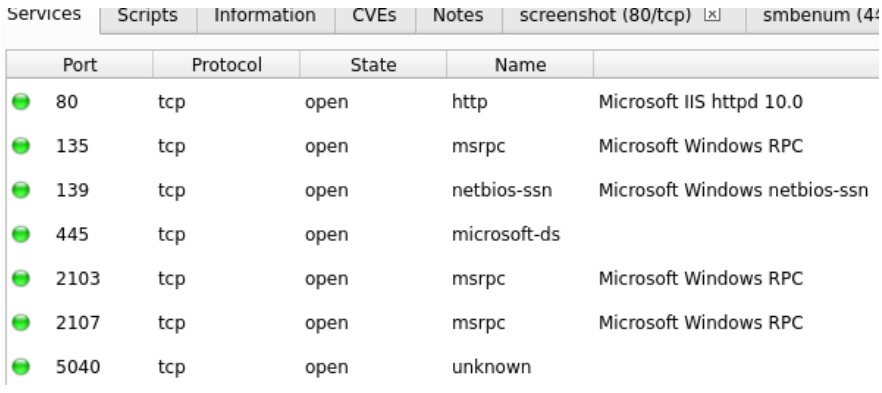


Figura 5.14. *Puertos Activos del IIS Hewlett Packard*

5.1.10.2 Asustek Compute y Dell

En el servidor de sistema operativo *Asustek Computer y Dell* detectamos activos los puertos 80, 5040 y 7680, asociados a los servicios Microsoft IIS httpd 10.0, unknown y pando-pub.

Services | Scripts | Information | CVEs | Notes | screenshot (80/tcp)

Port	Protocol	State	Name	
80	tcp	open	http	Microsoft IIS httpd 10.0
5040	tcp	open	unknown	
7680	tcp	open	pando-pub	

Figura 5.15. *Puertos Activos del IIS Asustek Computer*

5.1.10.3 Dell Modelo 2

En el servidor de sistema operativo *Dell Modelo 2* detectamos activos los puertos 80, 135, 139, 445, 2103, 2107, 5040 y 7680, asociados a los servicios Microsoft IIS httpd 10.0, msrpc, netbios-ssn, microsoft-ds y varios identificados como tcpwrapped.

Services | Scripts | Information | CVEs | Notes | screenshot (80/tcp) | smbenum (445/tcp)

Port	Protocol	State	Name	
80	tcp	open	http	Microsoft IIS httpd 10.0
135	tcp	open	msrpc	Microsoft Windows RPC
139	tcp	open	netbios-ssn	Microsoft Windows netbios-ssn
445	tcp	open	microsoft-ds	
2103	tcp	open	tcpwrapped	
2107	tcp	open	tcpwrapped	
5040	tcp	open	tcpwrapped	
7680	tcp	open	tcpwrapped	

Figura 5.16. *Puertos Activos del IIS Dell Modelo 2*

5.1.10.4 HewlettPackard Modelo 2

En el servidor de sistema operativo *HewlettPackard Modelo 2* detectamos activos los puertos 80, 5040 y 9444, asociados a los servicios Microsoft IIS httpd 10.0, unknown y wso2esb-console

Services | Scripts | Information | CVEs | Notes | screenshot (80/tcp)

Port	Protocol	State	Name	
80	tcp	open	http	Microsoft IIS httpd 10.0
5040	tcp	open	unknown	
9444	tcp	open	wso2esb-console	

Figura 5.17. *Puertos Activos del IIS HewlettPackard Modelo 2*

5.1.10.5 HewlettPackard Modelo 3

En el servidor de sistema operativo *HewlettPackard Modelo 3* detectamos activos los puertos 80, 135, 139, 445, 5040, 5357, 7070, 7680 y 10002, asociados a los servicios Microsoft IIS httpd 10.0, msrpc, netbios-ssn, microsoft-ds y varios tcpwrapped.

Services | Scripts | Information | CVEs | Notes | screenshot (5357/tcp) | screenshot (80/tcp)

Port	Protocol	State	Name	
80	tcp	open	http	Microsoft IIS httpd 10.0
135	tcp	open	msrpc	Microsoft Windows RPC
139	tcp	open	netbios-ssn	Microsoft Windows netbios-ssn
445	tcp	open	microsoft-ds	
5040	tcp	open	tcpwrapped	
5357	tcp	open	tcpwrapped	
7070	tcp	open	tcpwrapped	
7680	tcp	open	tcpwrapped	
10002	tcp	open	tcpwrapped	

***Figura 5.18**. Puertos Activos del IIS HewlettPackard Modelo 3*

5.1.10.6 HewlettPackard Modelo 4

En el servidor de sistema operativo *HewlettPackard Modelo 4* detectamos activos los puertos 623 y 16992, asociados a los servicios Microsoft IIS httpd 10.0 e Intel AMT httpd 7.1.3.

Port	Protocol	State	Name	
623	tcp	open	http	Intel Active Management Technology User Notification Service http admin 5.2.1
16992	tcp	open	http	Intel Active Management Technology User Notification Service http admin 5.2.1

***Figura 5.19**. Puertos Activos del IIS HewlettPackard Modelo 4*

5.1.10.7 HewlettPackard Modelo 5

En el servidor de sistema operativo *HewlettPackard Modelo 5* detectamos activos los puertos 80, 623, 7070, 7680, 10002 y 16992 asociados a los servicios Microsoft IIS httpd 10.0, Intel AMT httpd 7.1.3, RealServer y pando-pub.

Port	Protocol	State	Name	
80	tcp	open	http	Microsoft IIS httpd 10.0
623	tcp	open	http	Intel Active Management Technology User Notification Service httpd 7.1.3
7070	tcp	open	realserver	
7680	tcp	open	pando-pub	
10002	tcp	open	documentum	
16992	tcp	open	http	Intel Active Management Technology User Notification Service httpd 7.1.3

***Figura 5.20**. Puertos Activos del IIS HewlettPackard Modelo 5*

5.1.11 Scripts usados para el IIS

5.1.11.1 Scripts usados para el IIS Hewlett Packard

Puerto(s)	Servicio	Script(s)	Objetivo
80	HTTP (IIS 10.0)	http-enum, http-title, http-headers, http-vuln-*, http-vuln-cve*, http-robots.txt.nse, http-iis-webdav-vuln	Enumeramos rutas, analizamos cabeceras HTTP y detectamos vulnerabilidades en IIS (como WebDAV) y CVEs conocidas en el servidor web.
135, 2103, 2107	MSRPC	vuln, vulners, msrpc-enum	Detectamos vulnerabilidades asociadas a servicios RPC y enumeramos interfaces expuestas; analizamos servicios administrativos o de backend.
139, 445	SMB (NetBIOS-SSN / Microsoft-DS)	smb-enum-shares, smb-enum-users, smb-os-discovery, smb-brute, smb-vuln-ms17-010	Enumeramos recursos y usuarios; detectamos el sistema operativo y vulnerabilidades críticas como EternalBlue.
5040	Desconocido	vuln, vulners	Intentamos identificar el servicio oculto y detectar posibles vulnerabilidades asociadas.

Tabla 5.13. *Análisis agrupado de servicios y scripts NSE aplicados*

5.1.11.2 Scripts usados para el IIS Asustek Compute y Dell

Puerto(s)	Servicio	Script(s)	Objetivo
80	HTTP (IIS 10.0)	http-enum, http-title, http-headers, http-vuln-*, http-vuln-cve*, http-robots.txt.nse, http-iis-webdav-vuln	Detectamos rutas ocultas, cabeceras inseguras, vulnerabilidades específicas de IIS y módulos como WebDAV.
5040	Desconocido	vuln, vulners	Intentamos identificar el servicio activo en el puerto e identificar vulnerabilidades conocidas asociadas.
7680	pando-pub	vuln, vulners	Analizamos el servicio "pando-pub", poco común, buscando CVEs o vulnerabilidades documentadas.

Tabla 5.14. *Scripts NSE aplicados a puertos abiertos del host con IIS*

5.1.11.3 Scripts usados para el IIS Dell Modelo 2

Puerto(s)	Servicio	Script(s)	Objetivo
80	HTTP (IIS 10.0)	http-title, http-enum, http-headers, http-vuln-*, http-vuln-cve*, http-iis-webdav-vuln	Enumeramos páginas, extraemos cabeceras y detectamos vulnerabilidades conocidas de IIS como WebDAV y posibles CVEs.
135	MSRPC	vuln, vulners, msrpc-enum	Detectamos vulnerabilidades RPC y enumeramos interfaces disponibles en el sistema.
139, 445	SMB (NetBIOS / Microsoft-DS)	smb-os-discovery, smb-enum-shares, smb-enum-users, smb-brute, smb-vuln-ms17-010	Descubrimos el sistema operativo remoto, compartimos recursos y detectamos vulnerabilidades SMB críticas como EternalBlue.
2103, 2107, 5040, 7680	tcpwrapped	banner, version, unusual-port, vuln, vulners	Identificamos servicios ocultos tras cortafuegos, obtenemos banners y versiones y buscamos vulnerabilidades asociadas.

Tabla 5.15. *Análisis de servicios agrupados y scripts NSE para host con múltiples puertos envueltos (tcpwrapped)*

5.1.11.4 Scripts usados para el IIS HewlettPackard Modelo 2

Puerto(s)	Servicio	Script(s)	Objetivo
80	HTTP (IIS 10.0)	http-title, http-enum, http-headers, http-vuln-*, http-vuln-cve*, http-iis-webdav-vuln	Identificamos el contenido web, cabeceras, rutas accesibles y posibles vulnerabilidades específicas del servidor IIS como WebDAV.
5040	Desconocido	banner, version, unusual-port, vuln, vulners	Obtenemos el banner del servicio, identificamos software oculto tras el puerto y comprobamos si existen vulnerabilidades asociadas.
9444	WSO2ESB Console (HTTPS)	http-title, http-enum, http-headers, vuln, vulners, ssl-enum-ciphers, ssl-cert	Enumeramos funcionalidades de consola, cabeceras HTTP, y evaluamos la seguridad del canal cifrado (TLS/SSL), incluyendo certificado y cifrados.

Tabla 5.16. *Análisis de servicios y scripts NSE aplicados a host con consola WSO2 y servidor IIS*

5.1.11.5 Scripts usados para el IIS HewlettPackard Modelo 3

Puerto(s)	Servicio	Script(s)	Objetivo
80	HTTP (IIS 10.0)	http-title, http-enum, http-headers, http-vuln-*, http-vuln-cve*, http-iis-webdav-vuln	Detectamos contenido expuesto, cabeceras inseguras, rutas accesibles y vulnerabilidades en IIS (como WebDAV).
135	MSRPC	vuln, vulners, msrpc-enum	Enumeramos interfaces RPC y detectamos posibles vulnerabilidades conocidas en el servicio.
139, 445	SMB (NetBIOS / Microsoft-DS)	smb-os-discovery, smb-enum-shares, smb-enum-users, smb-brute, smb-vuln-ms17-010	Descubrimos recursos y usuarios de red, detectamos el sistema remoto y evaluamos la vulnerabilidad EternalBlue.
5040, 5357, 7070, 7680, 10002	tcpwrapped	banner, version, unusual-port, vuln, vulners	Identificamos servicios ocultos tras cortafuegos, obtenemos banners, detectamos versiones y buscamos vulnerabilidades asociadas.

Tabla 5.17. *Análisis de servicios y scripts NSE aplicados a host con múltiples servicios tcpwrapped*

5.1.11.6 Hewlett Packard Modelo 4

Puerto	Servicio	Script(s)	Objetivo
623	HTTP (Intel AMT — User Notification Service)	http-enum, http-robots.txt, banner, http-title, http-headers, vuln, vulners	Enumeramos rutas y recursos expuestos por la interfaz de notificación; obtenemos banners para identificar versión; comprobamos cabeceras inseguras o información divulgada; y buscamos CVEs públicos asociados a la versión del servicio.
16992	HTTP (Intel AMT — Web GUI / SOAP over HTTP)	http-enum, ssl-cert (si responde TLS en otro puerto), http-title, amt-info (nmap-amt NSE si disponible), vuln, vulners, banner	Identificamos la presencia de la interfaz web de administración (AMT), recopilamos información de versión y certificado (si aplica), identificamos funcionalidades SOAP/WEBGUI y detectamos vulnerabilidades conocidas que permitan acceso remoto o escalado (por ejemplo vulnerabilidades AMT históricas).

Tabla 5.18. *Scripts NSE aplicados a puertos detectados en Hewlett Packard Modelo 4*

5.1.11.7 Hewlett Packard Modelo 5

Puerto	Servicio (según nmap)	Script(s) seleccionados	Objetivo (resumido)
80	HTTP (Microsoft IIS httpd 10.0)	http-enum, http-title, http-robots.txt.nse, http-headers, http-vuln-, http-vuln-cve	Enumeramos rutas y páginas y recursos expuestos; localizamos formularios/robots.txt; recopilamos cabeceras para detectar configuraciones inseguras; y buscamos CVEs/bugs conocidos en IIS y aplicaciones web.
623	HTTP (Intel AMT — User Notification Service httpd 7.1.3)	http-enum, http-title, banner, vuln, vulners, http-headers	Identificamos endpoints AMT/management, obtenemos banners y versiones y buscamos vulnerabilidades públicas asociadas a Intel AMT.
7070	RealServer / realserver (streaming)	banner, http-enum, http-title, vuln, vulners	Detectamos servidor de streaming (RealServer/Helix), extraemos banner y versión y comprobamos vulnerabilidades históricas de motores RealServer/RTSP.
7680	pando-pub (posible Delivery Optimization / WDU/servicio MS)	http-enum, http-headers, banner, vuln, vulners	Identificamos el servicio exacto (peers, delivery/optimization o componente propietario), obtenemos banner y buscamos CVEs o comportamientos P2P inseguros.
10002	documentum	banner, http-enum, http-headers, vuln, vulners	Confirmamos servicio Documentum (Content Server), obtenemos versión/banner y buscamos vulnerabilidades o errores de configuración que permitan acceso a contenido.
16992	HTTP (Intel AMT — Web GUI httpd 7.1.3)	http-enum, http-title, ssl-cert (si aplica), http-headers, vuln, vulners	Enumeramos la interfaz web de gestión AMT, analizamos certificados/TLS si aplica y buscamos CVEs que puedan permitir control remoto o divulgación de información.

Tabla 5.19. *Scripts NSE aplicados y para Hewlett Packard Modelo 5*

5.1.12 Escaneo de Puertos Activos de los Servidores

Analizamos los servidores, evaluando vulnerabilidades de forma concreta (no genérica como en la Fase 3). Al detectarse pocos perfiles de servidor, realizamos pruebas individualizadas (escaneos, revisión de configuraciones y versiones), documentando las vulnerabilidades encontradas en cada caso.

5.1.12.1 Hewlett Packard Enterprise Servidor

En el servidor de sistema operativo *Hewlett Packard*, identificamos los siguientes puertos activos: 21 (FTP), 80 (Microsoft IIS), 135 (MSRPC), 139 (NetBIOS-SSN), 445 (Microsoft-DS/SMB), 2099 (H.225 VoIP), 2382 (MS-OLAP), 3389 (RDP), 4042 (Tornado httpd), 4043 (Tornado httpd), 4718 (servicio propietario), 5357 (Microsoft HTTPAPI), 5985 (WinRM), 8391 (Microsoft HTTPAPI), 8888 (Jetty), 10447 (Microsoft IIS), 14222 (Microsoft SQL Server) y 20447 (Microsoft IIS).

Port	Protocol	State	Name	
21	tcp	open	ftp	FileZilla ftpd
80	tcp	open	http	Microsoft IIS httpd 10.0
135	tcp	open	msrpc	Microsoft Windows RPC
139	tcp	open	netbios-ssn	Microsoft Windows netbios-ssn
445	tcp	open	microsoft-ds	
2099	tcp	open	h2250-ann...	
2382	tcp	open	ms-olap3	
3389	tcp	open	ms-wbt-se...	Microsoft Terminal Services
4042	tcp	open	http	Tornado httpd 6.1
4043	tcp	open	http	Tornado httpd 6.1
4718	tcp	open		
5357	tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5985	tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8391	tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8888	tcp	open	http	Jetty 9.4.16.v20190411
10447	tcp	open	http	Microsoft IIS httpd 10.0
14222	tcp	open	ms-sql-s	Microsoft SQL Server
20447	tcp	open	http	Microsoft IIS httpd 10.0

Figura 5.21. *Puertos Activos del IIS Hewlett Packard*

5.1.12.2 Base de datos Microsoft SQL Server

En el servidor de sistema operativo *Hewlett Packard*, constatamos un servicio Microsoft SQL Server escuchando en el puerto 1433 (instancia SQL Server 2014 reportada) y un servicio RealServer/streaming en el puerto 7070.

Port	Protocol	State	Name	
21	tcp	open	ftp	FileZilla ftpd
80	tcp	open	http	Microsoft IIS httpd 10.0
135	tcp	open	msrpc	Microsoft Windows RPC
139	tcp	open	netbios-ssn	Microsoft Windows netbios-ssn
445	tcp	open	microsoft-ds	
2099	tcp	open	h2250-ann...	
2382	tcp	open	ms-olap3	
3389	tcp	open	ms-wbt-se...	Microsoft Terminal Services
4042	tcp	open	http	Tornado httpd 6.1
4043	tcp	open	http	Tornado httpd 6.1
4718	tcp	open		
5357	tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5985	tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8391	tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8888	tcp	open	http	Jetty 9.4.16.v20190411
10447	tcp	open	http	Microsoft IIS httpd 10.0
14222	tcp	open	ms-sql-s	Microsoft SQL Server
20447	tcp	open	http	Microsoft IIS httpd 10.0

Figura 5.22. *Puertos Activos del SQL Server*

5.1.13 Scripts usados para los Servidores

5.1.13.1 Scripts usados para Servidor Hewlett Packard Enterprise

Puerto	Servicio (según nmap)	Script(s) seleccionados	Objetivo (resumido)
21	FTP (FileZilla ftpd)	ftp-anon, ftp-brute, ftp-bounce, ftp-proftpd-backdoor, ftp-vsftpd-backdoor, vuln, vulners	Comprobar acceso anónimo o escritura; probar fuerza bruta; detectar backdoors conocidos y buscar CVEs.
80	HTTP (Microsoft IIS httpd 10.0)	http-enum, http-title, http-robots.txt, http-headers, http-brute, http-form-brute, http-vuln-*, vulners	Mapear rutas/formularios, verificar cabeceras y buscar vulnerabilidades en IIS y aplicaciones web.
135	MSRPC (Microsoft Windows RPC)	msrpc-enum, smb-os-discovery, vuln, vulners	Enumerar interfaces MSRPC/servicios expuestos; detectar vulnerabilidades RPC.

Tabla 5.20. *Scripts NSE aplicados y objetivo — Parte 1*

Puerto	Servicio (según nmap)	Script(s) seleccionados	Objetivo (resumido)
139	NetBIOS-SSN	smb-enum-shares, smb-enum-users, smb-enum-groups, smb-brute, smb-os-discovery, vulners	Enumeración de recursos compartidos, usuarios y grupos; comprobar vulnerabilidades SMB.
445	Microsoft-DS (SMB)	smb-enum-shares, smb-enum-users, smb-enum-groups, smb-os-discovery, smb-brute, smb-vuln-ms17-010, vuln, vulners	Enumerar shares/usuarios; probar credenciales; detectar vulnerabilidades críticas SMB (ej. MS17-010).
2099	H.225 / VoIP control	banner, vuln, vulners	Obtener banner e identificar producto; buscar CVEs en servicios H.225/VoIP.
2382	ms-olap3 (Microsoft Analysis Services / OLAP)	msrpc-enum, smb-os-discovery, vuln, vulners	Identificar instancia OLAP/Analysis Services; obtener versión y buscar vulnerabilidades.
3389	RDP (Microsoft Terminal Services)	rdp-enum-encryption, vuln, vulners, rdp-vuln-*	Determinar cifrado/NLA de RDP; detectar vulnerabilidades RDP conocidas.
4042	HTTP (Tornado httpd 6.1)	http-enum, http-title, http-headers, http-vuln-*, vuln, vulners	Mapear endpoints de la app Tornado; identificar versión y buscar CVEs.
4043	HTTP (Tornado httpd 6.1)	http-enum, http-title, http-headers, http-vuln-*, vuln, vulners	Mismo objetivo que 4042: enumeración y detección de vulnerabilidades.
4718	Servicio propietario / auxiliar	banner, http-enum, vuln, vulners	Identificar servicio vía banner/HTTP y buscar CVEs o problemas de configuración.

Tabla 5.21. *Scripts NSE aplicados y objetivo — Parte 2*

Puerto	Servicio (según nmap)	Script(s) seleccionados	Objetivo (resumido)
5357, 5985, 8391	HTTP (Microsoft HTTPAPI / SSDP / WinRM)	http-enum, http-headers, http-title, vuln, vulners	Mapear endpoints Microsoft HTTPAPI/WinRM/SSDP; comprobar cabeceras y CVEs.
8888	HTTP (Jetty 9.4.16.v20190411)	http-enum, http-title, http-headers, http-vuln-*, vuln, vulners	Enumerar aplicaciones en Jetty; identificar versión y buscar vulnerabilidades.
10447	HTTP (Microsoft IIS httpd 10.0)	http-enum, http-title, http-headers, http-vuln-*, vulners	Mapeo y búsqueda de CVEs en la instancia IIS en este puerto.
14222	ms-sql-s (Microsoft SQL Server)	ms-sql-info, ms-sql-ntlm-info, ms-sql-empty-password (si procede), vuln, vulners	Recopilar info de instancia SQL Server; detectar autenticación débil y CVEs aplicables.
20447	HTTP (Microsoft IIS httpd 10.0)	http-enum, http-title, http-headers, http-vuln-*, vulners	Enumeración y búsqueda de vulnerabilidades/configuraciones inseguras en IIS.

Tabla 5.22. *Scripts NSE aplicados y objetivo — Parte 3*

5.1.13.2 Scripts usados para la base de datos Microsoft SQL Server

Puerto	Servicio (detectado)	Script(s) seleccionados	Objetivo (resumido)
1433	Microsoft SQL Server 2014 (ms-sql-s)	ms-sql-info, ms-sql-config, ms-sql-query, ms-sql-empty-password, ms-sql-ntlm-info, vuln, vulners, port-states	Recopilar versión/instancias y configuración; listar bases y linked servers; probar autenticación débil (sa con contraseña vacía); obtener información divulgada por NTLM; y mapear vulnerabilidades públicas para priorizar pruebas.
7070	RealServer (realserver / streaming)	vuln, vulners, http-vuln-cve*, http-vuln-*, port-states, banner	Identificar banner/versión del servidor de streaming; comprobar CVEs conocidos (exposición a DoS y otras vulnerabilidades históricas en RealServer); y determinar el estado/alcance del puerto antes de pruebas intrusivas.

Tabla 5.23. *Scripts NSE aplicados y objetivo par la base de datos SQL Server*

5.1.14 Vulnerabilidades detectadas en la fase de Análisis Básico

En esta sección detallamos las vulnerabilidades más relevantes detectadas en la fase de Análisis Básico de las impresoras, incluyendo los puertos vulnerados y servicios asociados

Mitigación de Slowloris (CVE-2007-6750)

```
443     |tcp                    |open                   |https          |syn-ack        |               |               |
http-slowloris-check
  VULNERABLE:
  Slowloris DOS attack
    State: LIKELY VULNERABLE
    IDs:  CVE:CVE-2007-6750
      Slowloris tries to keep many connections to the target web server open and hold
      them open as long as possible.  It accomplishes this by opening connections to
      the target web server and sending a partial request. By doing so, it starves
      the http server's resources causing Denial Of Service.

    Disclosure date: 2009-09-17
    References:
      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
      http://ha.ckers.org/slowloris/
```

Figura 5.23. *Sloworis DOS Attack*

Para proteger un servidor web contra ataques Slowloris, recomendamos las siguientes medidas:

- **Apache:** activamos el módulo *mod_reqtimeout* (disponible a partir de la versión 2.2.15) para imponer tiempos límite mínimos a la recepción de cabeceras y cuerpo de peticiones HTTP, evitando que conexiones parciales se mantengan indefinidamente [143].
- **NGINX y servidores no bloqueantes:** aplicamos límites de conexión (*limit_conn_zone*) y de tasa (*limit_req_zone*); configuramos tiempos de espera de cabecera, cuerpo y keep-alive; funcionan mejor debido a su arquitectura asíncrona [89].

- **Balanceadores de carga / WAF / proxies:** utilizamos dispositivos o servicios que sólo reenvían peticiones HTTP completas o redirigen clientes tras validar cabeceras; así el servidor backend queda protegido [65].
- **Reglas de firewall / iptables:** limitamos el número de conexiones concurrentes por IP al puerto 80 (por ejemplo, con *-m connlimit*), reduciendo el riesgo de ataques Slowloris distribuidos [143].
- **Aumentar límites de conexión:** incrementamos parámetros como *MaxClients* (Apache) o el número máximo de clientes, aunque esto solo retrasa el ataque y no lo previene completamente [143].

Mitigación de vulnerabilidades en Genivia gSOAP (CVE-2017-9765, CVE-2019-7659, CVE-2020-1357[4–8], CVE-2021-21783)

```
vulners
  cpe:/a:genivia:gsoap:2.7:
        CVE-2021-21783  9.8     https://vulners.com/cve/CVE-2021-21783
        CVE-2020-13576  9.8     https://vulners.com/cve/CVE-2020-13576
        CVE-2019-7659   8.1     https://vulners.com/cve/CVE-2019-7659
        CVE-2017-9765   8.1     https://vulners.com/cve/CVE-2017-9765
        CVE-2020-13578  7.5     https://vulners.com/cve/CVE-2020-13578
        CVE-2020-13577  7.5     https://vulners.com/cve/CVE-2020-13577
        CVE-2020-13575  7.5     https://vulners.com/cve/CVE-2020-13575
        CVE-2020-13574  7.5     https://vulners.com/cve/CVE-2020-13574
        SSV:96284       6.8     https://vulners.com/seebug/SSV:96284    *EXPLOIT*
```

Figura 5.24. *Gsoap*

Para mitigar las vulnerabilidades críticas en Genivia gSOAP, recomendamos seguir estas medidas:

- **Actualizar a la versión más reciente de gSOAP (≥ 2.8.111)** Todas las vulnerabilidades críticas relacionadas con WS-Addressing y WS-Security, incluyendo CVE-2020-13576, CVE-2021-21783, CVE-2020-13574–78 y CVE-2019-7659, están resueltas en esa versión o posterior [131].
- **Recompilar sin soporte de cookies (parámetro *-DWITH_COOKIES*)** CVE-2019-7659 se evita eliminando el soporte de cookies en el servidor SOAP [91].
- **Desactivar plugins no necesarios (WS-Addressing / WS-Security)** Las vulnerabilidades más severas se localizan en estos plugins; si no son imprescindibles, es recomendable deshabilitarlos durante la generación del servicio [131].

Mitigación de LFI en phpMyAdmin grab_globals.lib.php (CVE-2005-3299)

```
http-phpmyadmin-dir-traversal
  VULNERABLE:
  phpMyAdmin grab_globals.lib.php subform Parameter Traversal Local File Inclusion
    State: LIKELY VULNERABLE
    IDs:  CVE:CVE-2005-3299
      PHP file inclusion vulnerability in grab_globals.lib.php in phpMyAdmin 2.6.4 and 2.6.4-pl1 allows remote attackers to include local files via the $__redirect parameter, possibly involving the subform array.

    Disclosure date: 2005-10-nil
    Extra information:
      ../../../../../etc/passwd not found.

    References:
      http://www.exploit-db.com/exploits/1244/
      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3299
```

Figura 5.25. *Php*

Para protegernos contra la vulnerabilidad CVE-2005-3299, que permite incluir archivos locales mediante el parámetro *$__redirect* en phpMyAdmin 2.6.4 y 2.6.4-pl1, recomendamos estas medidas:

- **Actualizar phpMyAdmin a 2.6.4-pl2 o versiones posteriores** La vulnerabilidad se corrige explícitamente en la versión 2.6.4-pl2 :contentReference. [133]
- **Deshabilitar parámetros no validados** Revisamos y eliminamos el uso del parámetro *$__redirect* o cualquier componente de entrada no sanitizado en aplicaciones personalizadas que utilicen grab_globals.lib.php. [133]
- **Bloquear accesos a archivos críticos desde la web** Mediante firewall, reglas en el servidor web o configuración de PHP, impedimos el acceso o inclusión a rutas como */etc/passwd*. [133]
- **Eliminar grab_globals.lib.php si no es necesario** En instalaciones modernas de phpMyAdmin, este archivo es obsoleto: eliminarlo evita el riesgo incluso en versiones afectadas. [133]

Mitigación de POODLE (CVE-2014-3566)

```
ssl-poodle
```

```
VULNERABLE:
SSL POODLE information leak
  State: VULNERABLE
  IDs:  BID:70574  CVE:CVE-2014-3566
        The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
        products, uses nondeterministic CBC padding, which makes it easier
        for man-in-the-middle attackers to obtain cleartext data via a
        padding-oracle attack, aka the "POODLE" issue.
  Disclosure date: 2014-10-14
  Check results:
    TLS_RSA_WITH_AES_128_CBC_SHA
  References:
    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
    https://www.securityfocus.com/bid/70574
    https://www.imperialviolet.org/2014/10/14/poodle.html
    https://www.openssl.org/~bodo/ssl-poodle.pdf
```

Figura 5.26. *SslPoodle*

Para prevenir la fuga de información por desbordamiento de CBC en SSL 3.0, recomendamos las siguientes acciones:

- **Deshabilitar completamente SSL 3.0** en servidor y cliente. En Apache, NGINX, IIS, JBoss, WebLogic, etc., lo configuramos para soportar sólo TLS 1.1/1.2/1.3. Evitamos cualquier mecanismo de downgrade incluido SSL 3.0 [27].
- **Habilitar TLS_FALLBACK_SCSV** para prevenir ataques de downgrade, aun si se conserva soporte temporal a SSL 3.0 [27].
- **Evitar cifrados CBC en SSL 3.0**, utilizando preferentemente RC4 como parche temporal mientras se completan las actualizaciones (aunque RC4 también tiene fallos, por lo que debe considerarse una solución transitoria) [27].
- **Aplicar parches y actualizaciones** OpenSSL ≥ 1.0.1j, NSS ≥ 3.16.2.3, 3.17.1 y LibreSSL ≥ 2.1.1 implementan TLS-FALLBACK-SCSV y deshabilitan SSL 3.0 por defecto [27].

Mitigación desmb-vuln-cve2009-3103

Para prevenir la vulnerabilidad CVE-2009-3103, que afecta a SMBv2 en Windows Server 2008 y Windows Vista SP1, recomendamos seguir las siguientes medidas:

- **Aplicar el boletín de seguridad MS09-050:**

  Microsoft publicó el boletín MS09-050 para corregir esta vulnerabilidad en SMBv2 [72].

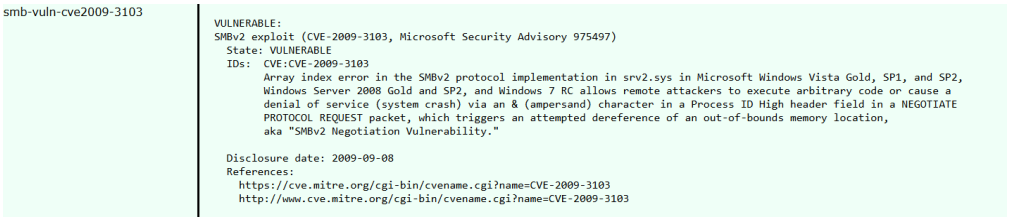


Figura 5.27. *Desmb-vuln-cve2009-3103*

- **Deshabilitar SMBv2 temporalmente:**

  Como medida temporal, podemos deshabilitar SMBv2 para mitigar el riesgo hasta aplicar el parche [72].

- **Actualizar a versiones no afectadas:**

  Migrar a versiones de Windows que no sean vulnerables (por ejemplo, Windows 7 RTM o Windows Server 2008 R2) evita la exposición [72].

- **Aplicar filtros de red:**

  Implementamos controles en red para bloquear tráfico SMBv2 no autorizado y prevenir que lleguen paquetes manipulados al servidor afectado [72].

Mitigación Diffie-Hellman

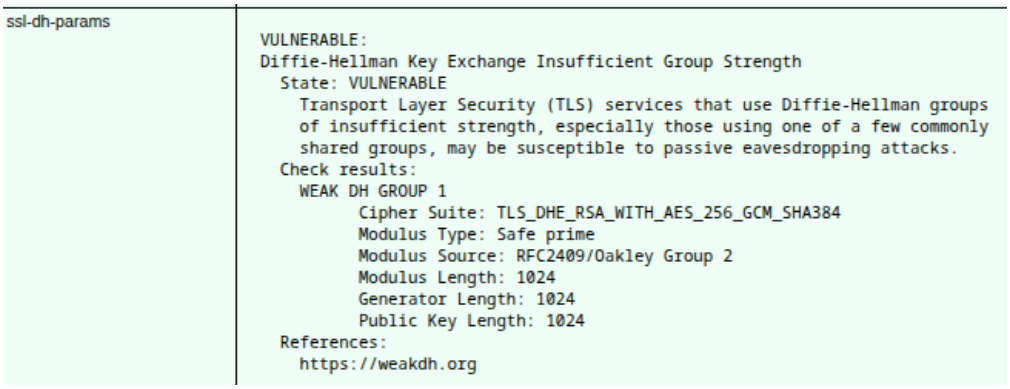


Figura 5.28. *Vulnerabilidad Diffie-Hellman*

Para prevenir la vulnerabilidad de Diffie-Hellman, recomendamos seguir las siguientes medidas:

- **Deshabilitar suites DH con tamaño de clave < 2048 bits**. Nos aseguramos de que en la configuración del servidor no estén permitidos grupos temporales con prime menor de 2048 bits. Esto incluye evitar "export cipher suites" y modos antiguos que dejen negociar grupos débiles [112].

- **Usar parámetros DH personalizados y seguros**. Generamos un nuevo archivo `dhparam` de al menos 2048 bits (preferiblemente 3072) para usar como grupo DH, en lugar de los parámetros por defecto compartidos que pueden ser comunes y susceptibles a ataques de precomputación [112].

- **Priorizar (Ephemeral) ECDHE / curvas elípticas**. Cambiamos la negociación de clave a ECDHE (o DH elíptico) cuando sea posible, ya que estos ofrecen mayor seguridad y resisten mejor los ataques modernos como Logjam [112].
- **Actualizar protocolos y bibliotecas TLS/SSL**. Nos aseguramos de usar versiones recientes del software que corrige vulnerabilidades DH débiles (OpenSSL, NSS, SChannel de Microsoft, Java, etc.). Muchas versiones recientes limpian los grupos débiles por defecto y permiten configurar parámetros más fuertes [112].
- **Revisar y restringir suites de cifrado**. Configuramos el servidor para permitir sólo suites que utilicen cifrado fuerte, evitando aquellas que usen cifrados CBC débiles, evitando Anonymous DH, DES, RC4, etc. Garantizamos que los algoritmos de intercambio de claves y firmas también sean seguros [112].

Mitigación de vulnerabilidades en OpenSSH 8.0 (CVE-2023-38408 y vulnerabilidades asociadas)

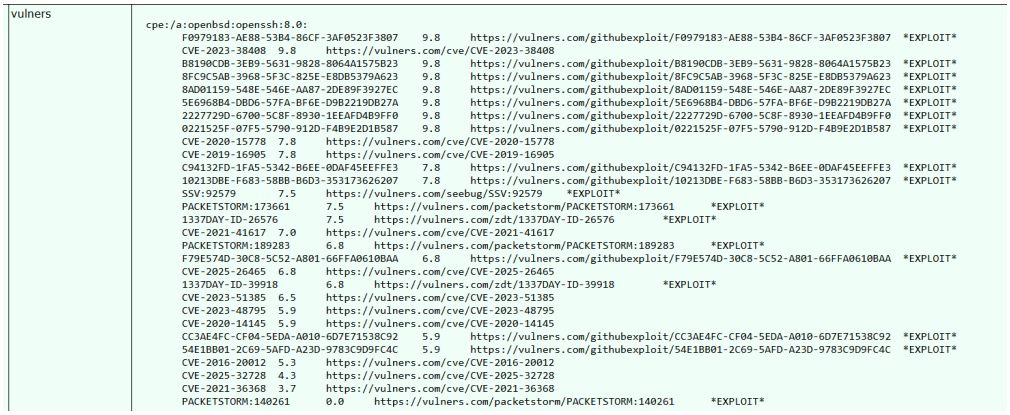


Figura 5.29. *Vulnerabilidades críticas en OpenSSH 8.0*

Para prevenir las vulnerabilidades detectadas en OpenSSH 8.0 (como CVE-2023-38408 y otras explotables), recomendamos adoptar las siguientes medidas:

- **Actualizar OpenSSH a una versión parcheada (≥ 9.3p2).** El fallo CVE-2023-38408 fue corregido en OpenSSH 9.3p2, por lo que actualizar elimina la vulnerabilidad del agente PKCS-11 con rutas inseguras. [95, 35]
- **Limitar o bloquear el uso de PKCS-11 / filtros de proveedores.** Inicializamos `ssh-agent` con una lista blanca vacía o restringida para los proveedores PKCS-11 (por ejemplo, `ssh-agent -P ''`) y reducimos el riesgo de que librerías inseguras sean cargadas. [35, 95]
- **Usar salto seguro (ProxyJump) en lugar de reenvío de agente.** En lugar de reenviar el agente entre saltos, usamos la directiva `ProxyJump` (o `-J`) para conectar varios hosts intermedios sin exponer el agente. [129, 128]

Mitigación HTTP verb tampering

Para prevenir la vulnerabilidad de HTTP verb tampering, recomendamos seguir las siguientes medidas:

- **Restringir los métodos HTTP permitidos solo a los estrictamente necesarios** — por ejemplo, permitir únicamente GET y POST si la aplicación no usa otros métodos [102].

Figura 5.30. Vulnerabilidad de HTTP verb tampering

- **Configurar reglas de autorización que apliquen a todos los métodos** — nos aseguramos de que cualquier método que solicite acceso a recursos protegidos sea chequeado por la lógica de autenticación/autorización [66].
- **Deshabilitar métodos inseguros o poco usados (TRACE, PUT, DELETE, etc.) si no se requieren**, y bloquear verbos no estándar o personalizados [102].

5.1.15 Vulnerabilidades encontradas en Impresoras y Escaners

- **Impresora HL-L5100DN, Impresora HL-L5100DN Modelo 2, HL-L5200-DW, Escaner PFU Modelo 2**
  - No detectamos vulnerabilidades en estas impresoras.
- **Impresora Brother HL-L5210-DN**
  - En el puerto 443 con servicio https, detectamos la vulnerabilidad 5.23, vulnerabilidad de tipo Sloworis DOS Attack.
- **Impresora HP Laserjet M404DN**
  - En el puerto 80, 631, 3910, 8080 con servicio http gSOAP detectamos la vulnerabilidad 5.24, vulnerabilidad de Genivia gSOAP.
  - En el puerto 631 con servicio http gSOAP, detectamos la vulnerabilidad 5.23, vulnerabilidad de tipo Sloworis DOS Attack.
  - En el 3910 con servicio http gSOAP, detectamos la vulnerabilidad 5.25, vulnerabilidad CVE-2005-3299 de php-MyAdmin.
- **Impresora Ricoh MP C401SR**
  - En el puerto 7444 con servicio desconocido, detectamos la vulnerabilidad 5.26 vulnerabilidad de SslPoodle.
- **Impresora Ricoh MP C401SR Modelo 2**
  - En el puerto 7444 con servicio desconocido, detectamos la vulnerabilidad 5.26 vulnerabilidad de SslPoodle.
  - Detectamos la vulnerabilidad 5.27, vulnerabilidad de tipo desmb-vuln-cve2009-3103, no determinamos el puerto.

- **Escaner PFU**
  - En el puerto 80 con servicio http, detectamos la vulnerabilidad 5.23, vulnerabilidad de tipo Sloworis DOS Attack.

5.1.16 Vulnerabilidades encontradas en las Cámaras

- **Cámara GrandStream Network**
  - En el puerto 80 con servicio http con el producto minittpd , detectamos la vulnerabilidad 5.23, vulnerabilidad de tipo Sloworis DOS Attack.

5.1.17 Vulnerabilidades encontradas en los dispositivos de red

- **Millenial Net**
  - No detectamos vulnerabilidades relevantes en el dispositivo.
- **TpLink**
  - Detectamos la vulnerabilidad 5.27, vulnerabilidad de tipo desmb-vuln-cve2009-3103, no determinamos el puerto.

5.1.18 Vulnerabilidades encontradas en Ordenadores

- **Hewlett Packward, Hon Hai Precision y Hewlett Packward iLO**
  - No detectamos vulnerabilidades en estos ordenadores.

5.1.19 Vulnerabilidades encontradas en IIS

- **IIS Asustek Compute,Dell, HewlettPackard Modelo 2, HewlettPackard Modelo 3 y HewlettPackard Modelo 5**
  - No detectamos vulnerabilidades en estos ordenadores.
- **IIS Hewlett Packard y IIS Dell Modelo 2**
  - Detectamos la vulnerabilidad 5.27, vulnerabilidad de tipo desmb-vuln-cve2009-3103, no determinamos el puerto.
- **IIS Hewlett Packard Modelo 4**
  - Detectamos la vulnerabilidad 5.30, vulnerabilidad de tipo HTTP verb tampering, en el puerto 16992.
  - Detectamos la vulnerabilidad 5.23, vulnerabilidad de tipo Sloworis DOS Attack, en el puerto 16992 y 623.

5.1.20 Vulnerabilidades encontradas en Servidor y Base de Datos

- **Servidor Hewlett Packard**
  - No detectamos vulnerabilidades en este servidor.
- **Base de Datos Microsoft SQL Server**
  - Detectamos la vulnerabilidad 5.26, vulnerabilidad de tipo SSL POODLE, en el puerto 1433.
  - Detectamos la vulnerabilidad 5.28, vulnerabilidad de tipo Diffie-Hellman, en el puerto 1433.

5.1.21 Vulnerabilidades encontradas en la Pagina Web

- **Página web**
  - Detectamos la vulnerabilidad 5.29, vulnerabilidad de tipo OpenSSH, en el puerto 22 con servicio ssh.

6

Auditoria a Redes Wifi

6.0.1 Airodump-ng, Aireplay-ng y Aircrack-ng

La empresa OFIAUTO no trabajaba con redes inalámbricas, por lo que imposibilita realizar un ataque al router inalámbrico tanto de **cifrado WEP como WPA** debido a su inexistencia para romper la contraseña debido a que los dispositivos con conexión a internet que tienen se conectan mediante cable red ethernet. En el caso de estas pruebas considerando su infraestructura de red se descartan, ya que tienen un impacto nulo en la auditoría.

Figura 6.1. *Imagen de impresora conectado por cable Ethernet OFIAUTO*

7

Ataques a Contraseñas

7.1 Mimikatz

7.1.0.1 Detección de claves

Antes de intentar exportar tickets, comprobamos que la sesión sobre la que trabajábamos tenía credenciales cargadas y permisos elevados. Para ello ejecutamos los comandos de Mimikatz que enumeran credenciales y aseguran privilegios de depuración:

privilege*::*debug Este comando intenta habilitar el privilegio de depuración en el proceso de Mimikatz, lo que es necesario para acceder a ciertos datos sensibles en memoria. Si tiene éxito, indica que Mimikatz tiene los permisos necesarios para operar a nivel del sistema. ***sekurlsa*::*logonpasswords***

El comando *sekurlsa::logonpasswords* muestra las credenciales y sesiones disponibles en memoria y requiere privilegios elevados. Véase en la figura 7.1.

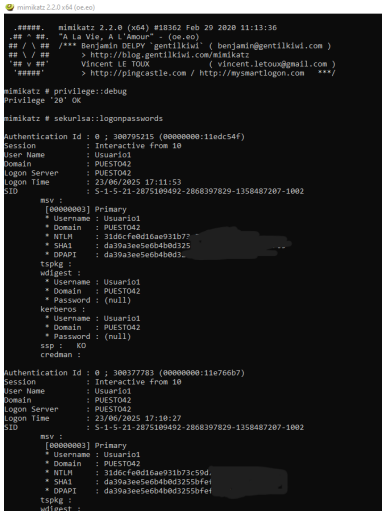


Figura 7.1. *Salida de sekurlsa::logonpasswords*

7.1.0.2 Identificación de TGT

En la Figura 7.2 observamos la salida de Mimikatz. Se muestran varios tickets para el usuario *Usuario1* en el dominio *PUESTO42*, incluyendo entradas bajo "Ticket Granting Service".

Aunque no aparece un "blob" binario o código TGT explícito en la pantalla, esto indica que el TGT sí está presente en memoria. Para extraerlo completamente debemos usar el comando: ***sekurlsa::tickets /export*** ***sekurlsa::tickets /export*** que podemos usar

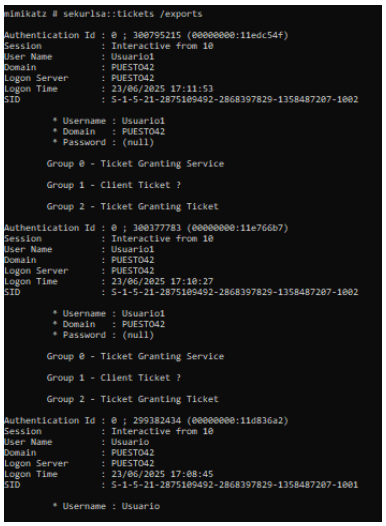


Figura 7.2. *Salida de sekurlsa::tickets /export mostrando el TGT en memoria*

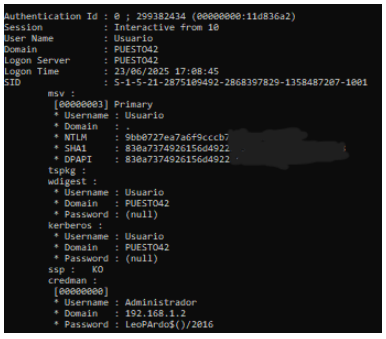


Figura 7.3. *Salida de sekurlsa::tickets /export mostrando contraseña en memoria*

para inyección de ticket ("pass-the-ticket") o análisis posterior. Como observamos en la Figura 7.3, también podemos extraer contraseñas en texto claro si están disponibles en memoria. [116]

7.1.0.3 Verificación de TGT en memoria

Por último, podemos verificar la presencia del TGT con: ***kerberos::ptt tgt*** o ver los tickets con ***kerberos::pttlist*** que lista los tickets Kerberos en memoria y confirma que el TGT está cargado y listo para usarse. [116] Como observamos en la siguiente figura, no se aprecia ningún código tgt generado ni tickets cargados en memoria.

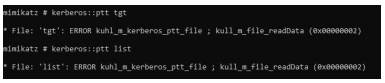


Figura 7.4. *Salida de kerberos::pttlist mostrando ausencia de tickets en memoria*

7.1.0.4 Conclusión: análisis de la extracción de credenciales con Mimikatz

Tras la secuencia de comprobación y extracción realizada (`privilege::debug` → `sekurlsa::logonpasswords` → `sekurlsa::tickets`
→ `sekurlsa::tickets /export`), alcanzamos la siguiente conclusión práctica y operativa.

- **Verificación previa necesaria.** Ejecutar `privilege::debug` y `sekurlsa::logonpasswords` es una práctica necesaria para confirmar que la herramienta dispone de los permisos y del contexto (sesiones con credenciales en memoria) requerido para intentar extraer tickets o contraseñas. Si estas comprobaciones no son positivas, las etapas posteriores carecen de sentido operativo.[116]

- **Impacto y priorización de mitigaciones.** Aunque en este caso concreto no conseguimos inyectar un ticket reutilizable con los comandos intentados, la detección de credenciales (contraseñas en texto claro) y la presencia de tickets en memoria confirman un riesgo real de escalada y movimiento lateral si un adversario logra exportar e inyectar un `.kirbi` válido. Por ello recomendamos como prioridades: restringir el uso de `SeDebugPrivilege`, habilitar LSA Protection / Credential Guard, monitorizar accesos a LSASS y creación de ficheros `.kirbi`, y desplegar detección EDR orientada a técnicas de volcado de credenciales. [116]

8

Auditoria aplicaciones Web

8.1 OWASP ZAP

En esta fase de la auditoría utilizamos la herramienta *OWASP Zed Attack Proxy (ZAP)* con el objetivo de analizar la seguridad de las aplicaciones web expuestas en los servicios HTTP que habíamos identificado previamente mediante el uso de *Legion*. Para ello, en primer lugar, iniciamos sesión en las páginas web auditadas con las credenciales disponibles, de manera que pudiésemos explorar y documentar en profundidad todas las secciones accesibles para un usuario autenticado.

Durante el proceso generamos informes que recogen de forma estructurada las vulnerabilidades encontradas, acompañadas de ejemplos concretos de las pruebas realizadas. La metodología que seguimos constó de tres etapas principales:

- Fase de *Ajax Spider*: llevamos a cabo un rastreo dinámico para identificar los recursos, formularios y rutas internas de la aplicación, permitiendo un mayor alcance en el posterior análisis.
- Escaneo pasivo: analizamos las peticiones y respuestas capturadas durante la navegación, sin modificar el tráfico, con el fin de detectar posibles problemas de configuración o debilidades de seguridad.
- Ataque activo a la aplicación: finalmente, ejecutamos un escaneo de vulnerabilidades con técnicas de inyección y explotación controlada, lo que nos permitió obtener un reporte detallado de los fallos más críticos presentes en la aplicación.

Servicios HTTP identificados

Services

Host	Port	Protocol	State	Version
192.168.1.2	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.2	5357	tcp	open	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
192.168.1.2	8888	tcp	open	Jetty 9.4.16.v20190411
192.168.1.2	4042	tcp	open	Tornado httpd 6.1
192.168.1.2	4043	tcp	open	Tornado httpd 6.1
192.168.1.2	5985	tcp	open	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
192.168.1.2	8391	tcp	open	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
192.168.1.2	10447	tcp	open	Microsoft IIS httpd 10.0
192.168.1.2	20447	tcp	open	Microsoft IIS httpd 10.0
192.168.1.13	80	tcp	open	HP Integrated Lights-Out web interface
192.168.1.13	443	tcp	open	HP Integrated Lights-Out web interface
192.168.1.22	80	tcp	open	Debut embedded httpd 1.30 (Brother/HP printer http admin)
192.168.1.22	443	tcp	open	Debut embedded httpd 1.30 (Brother/HP printer http admin)
192.168.1.22	631	tcp	open	Debut embedded httpd 1.30 (Brother/HP printer http admin)
192.168.1.25	80	tcp	open	Debut embedded httpd 1.30 (Brother/HP printer http admin)
192.168.1.25	443	tcp	open	Debut embedded httpd 1.30 (Brother/HP printer http admin)
192.168.1.25	631	tcp	open	Debut embedded httpd 1.30 (Brother/HP printer http admin)
192.168.1.30	80	tcp	open	
192.168.1.31	80	tcp	open	
192.168.1.32	80	tcp	open	
192.168.1.33	80	tcp	open	
192.168.1.37	80	tcp	open	

***Figura 8.1**. Puertos Activos de las Aplicaciones Web Parte 1*

192.168.1.41	80	tcp	open	lighttpd
192.168.1.42	80	tcp	open	lighttpd
192.168.1.43	80	tcp	open	lighttpd
192.168.1.99	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.198	80	tcp	open	lighttpd
192.168.1.199	8080	tcp	open	Knopflerfish httpd
192.168.1.203	80	tcp	open	
192.168.1.205	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.207	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.208	80	tcp	open	
192.168.1.210	80	tcp	open	gSOAP 2.7 (HP MFP printer)
192.168.1.210	8080	tcp	open	gSOAP 2.7 (HP MFP printer)
192.168.1.210	631	tcp	open	gSOAP 2.7 (HP MFP printer)
192.168.1.216	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.218	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.219	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.220	80	tcp	open	mini_httpd 1.30 26Oct2018
192.168.1.221	80	tcp	open	mini_httpd 1.30 26Oct2018
192.168.1.222	80	tcp	open	mini_httpd 1.30 26Oct2018
192.168.1.223	80	tcp	open	mini_httpd 1.30 26Oct2018

***Figura 8.2**. Puertos Activos de las Aplicaciones Web Parte 2*

192.168.1.221	80	tcp	open	mini_httpd 1.30 26Oct2018
192.168.1.222	80	tcp	open	mini_httpd 1.30 26Oct2018
192.168.1.223	80	tcp	open	mini_httpd 1.30 26Oct2018
192.168.1.225	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.226	80	tcp	open	
192.168.1.228	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.230	80	tcp	open	
192.168.1.232	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.239	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.240	16992	tcp	open	Intel Active Management Technology User Notification Service http admin 5.2.1
192.168.1.240	623	tcp	open	Intel Active Management Technology User Notification Service http admin 5.2.1
192.168.1.245	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.245	16992	tcp	open	Intel Active Management Technology User Notification Service httpd 7.1.3
192.168.1.245	623	tcp	open	Intel Active Management Technology User Notification Service httpd 7.1.3
192.168.1.249	80	tcp	open	Microsoft IIS httpd 10.0
192.168.1.251	80	tcp	open	Microsoft IIS httpd 10.0
217.76.142.216	80	tcp	open	Apache httpd
217.76.142.216	443	tcp	open	Apache httpd

Figura 8.3. *Puertos Activos de las Aplicaciones Web Parte 3*

8.1.1 Páginas web auditadas

En esta parte solo mostramos aquellas que estaban activas durante la auditoría. Auditamos las siguientes páginas web:

8.1.1.1 Impresoras y escáneres

- HL-5100DN

  `http://192.168.1.22`

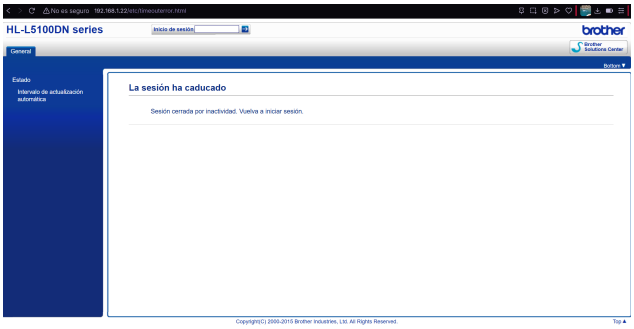


Figura 8.4. *Web HL-5100DN*

- HL-L5200DW

  `http://192.168.1.32`

- HL-5100DN Modelo 2

  `http://192.168.1.37`

- Ricoh MP C3003

  `http://192.168.1.199`

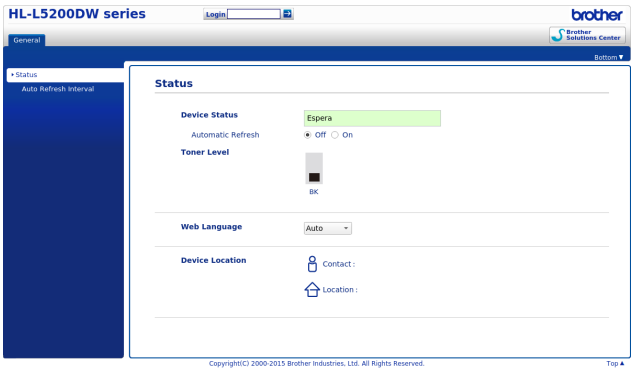


***Figura 8.5**. Web HL-L5200DW*

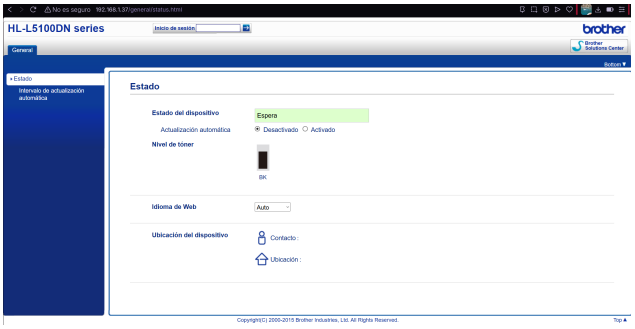


***Figura 8.6**. Web HL-5100DNM2*

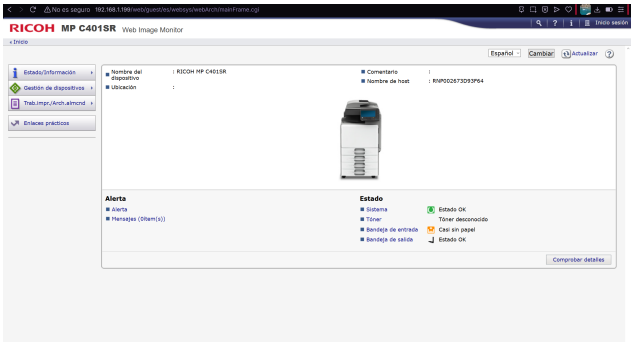


***Figura 8.7**. Web RICOH*

- **Ricoh MP C3003 Modelo 2**

  `http://192.168.1.101`

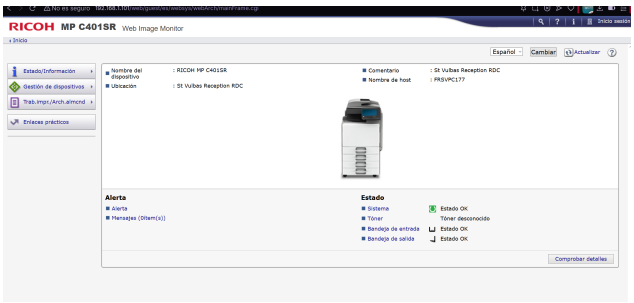


Figura 8.8. *Web RICOH Modelo 2*

- **HP LaserJet Pro MFP M130fw**

  `http://192.168.1.210`

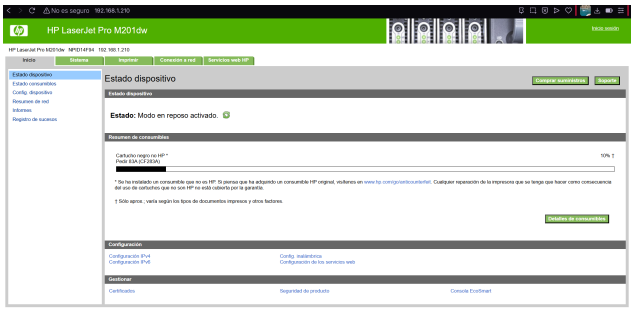


Figura 8.9. *Web HP-LASERJET*

- **Escáner PFU**

  `http://192.168.1.198`

Figura 8.10. *Web Escáner PFU*

- **Escáner PFU Modelo 2**

  `http://192.168.1.226`

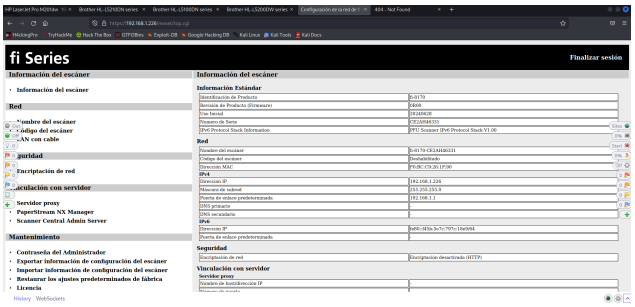


***Figura 8.11**. Web Escáner PFU Modelo 2*

8.1.1.2 Servidor y Base de Datos

- **Web Servidor Hewlett Packard**

  – IIS –>`http://192.168.1.2:80`

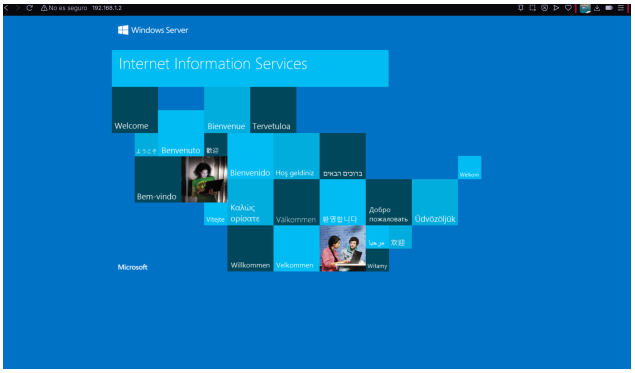


***Figura 8.12**. Web Escanner Servidor Hewlett Packard IIS Puerto 80*

  – Tornado httpd –>`http://192.168.1.2:4042`

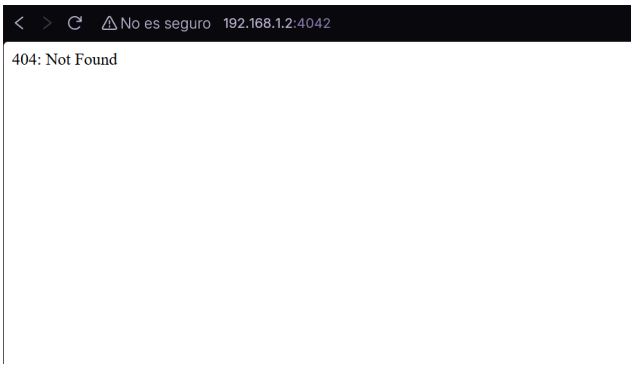


***Figura 8.13**. Web Escanner Servidor Hewlett Packard Tornado Puerto 4042*

– Tornado httpd –>`http://192.168.1.2:4043`

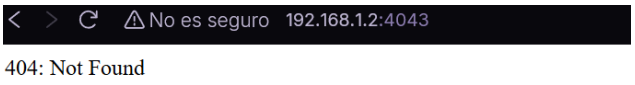


***Figura 8.14**. Web Escanner Servidor Hewlett Packard Tornado Puerto 4043*

– Microsoft HTTPAPI –>`http://192.168.1.2:5357`

– Microsoft HTTPAPI –>`http://192.168.1.2:5985`

Figura 8.15. *Web Escanner Servidor Hewlett Packard Microsoft HTTPAPI Puerto 5357*

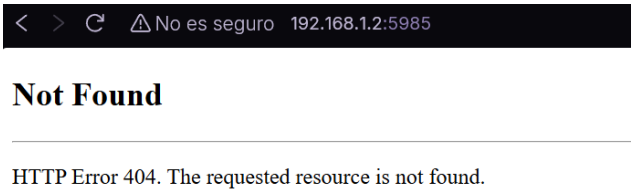


Figura 8.16. *Web Escanner Servidor Hewlett Packard Microsoft HTTPAPI Puerto 5985*

– IIS –>`http://192.168.1.2:10447`

***Figura 8.17**. Web Escanner Servidor Hewlett Packard IIS Puerto 10447*

– IIS –>`http://192.168.1.2:20447`

***Figura 8.18**. Web Escanner Servidor Hewlett Packard IIS Puerto 20447*

- Base de Datos no contiene servicio web.

8.1.1.3 IIS

- **Web IIS Hewlett Packard Modelo 3, Dell, Asus y Dell Modelo 2** no estaban activos sus servicios web.
- **Web IIS Hewlett Packard**

  `http://192.168.1.99`

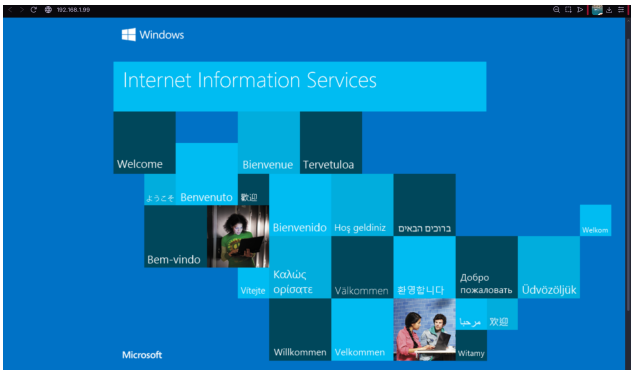


***Figura 8.19**. Web IIS Hewlett Packard*

- **Web IIS Hewlett Packard Modelo 4**

  `http://192.168.1.240:623`

- **Web IIS Hewlett Packard Modelo 4**

  `http://192.168.1.240:16992`

Figura 8.20. Web IIS Hewlett Packard Modelo 4 Puerto 623

Figura 8.21. Web IIS Hewlett Packard Modelo 4 Puerto 16992

- **Web IIS Hewlett Packard Modelo 5**

  `http://192.168.1.245:623`

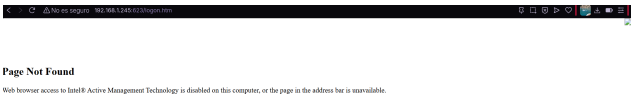


Figura 8.22. Web IIS Hewlett Packard Modelo 5 Puerto 623

- **Web IIS Hewlett Packard Modelo 5**

  `http://192.168.1.245:16992`

8.1.1.4 Ordenadores

- **Web Hewlett Packard y Hon Hai** no contienen servicio web.
- **Web IIS Hewlett Packard Enterprise**

  `http://192.168.1.13`

Figura 8.23. Web IIS Hewlett Packard Modelo 5 Puerto 16992

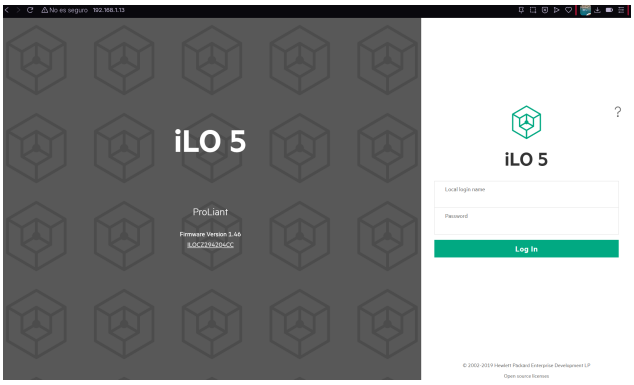


Figura 8.24. Web Hewlett Packard Ilo

8.1.1.5 Dispositivos de red

- **Tp-Link** no estaba activo su servicio web.
- **Web Millenial Net**

  `http://192.168.1.230`

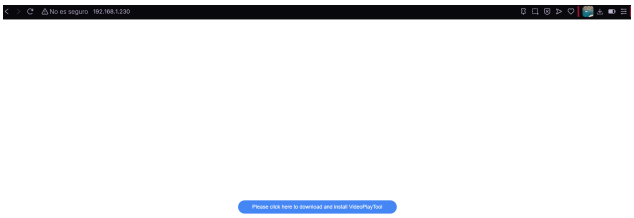


Figura 8.25. Web Millenial Net

8.1.1.6 Camaras

- **Web GrandStreamNetwork**

  `http://192.168.1.220`

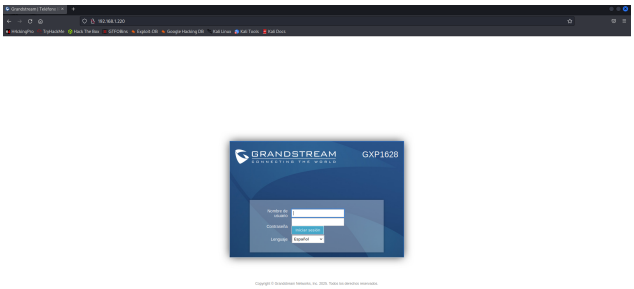


Figura 8.26. *Web GrandStreamNetwork*

8.1.2 Vulnerabilidades encontradas

A continuación, detallamos las vulnerabilidades más relevantes que detectamos durante la auditoría a aplicaciones web con OWASP ZAP, junto con una breve descripción y el riesgo asociado:

8.1.2.1 Mitigación de Redirección Externa

URL	http://192.168.1.22/general/status.html
Método	POST
Ataque	
Evidencia	
Otra información	Longitud URI de la cabecera de ubicación: 50 [/etc/passerror.html?url=%2Fgeneral%2Fstatus%2Ehtml]. Tamaño previsto de la respuesta: 350. Longitud del cuerpo de la respuesta: 7,579.

Figura 8.27. *Redirección Externa*

Para proteger una página web contra ataques de redirección externa, recomendamos las siguientes medidas:

1. **Validación de entrada con lista blanca (*"aceptar lo bueno conocido"*)**: Aceptamos únicamente valores que cumplan estrictamente con las especificaciones (longitud, tipo, formato, rango, reglas de negocio). Rechazamos o normalizamos cualquier entrada que no se ajuste rigurosamente. No confiamos únicamente en listas de denegación. [105]

2. **Mapeo mediante identificadores en lugar de URLs directas**: Usamos identificadores fijos (por ejemplo, numéricos) que se traducen internamente en una URL confiable. Rechazamos cualquier entrada que no corresponda al mapeo permitido. Ejemplo: ID 1 → "/login.asp", ID 2 → "https://www.example.com/". Herramientas como AccessReferenceMap de ESAPI facilitan este enfoque. [39]

3. **Lista blanca de dominios o URLs permitidos**: Solo redirigimos a destinos que estén explícitamente autorizados. Cualquier intento de redirección fuera de esta lista debe rechazarse. [124]

4. **Página intermedia de advertencia antes de redireccionar**: Mostramos al usuario una página que indique claramente que está abandonando el sitio, pidiendo confirmación mediante clic o con un retardo (timeout) antes de proceder. Prevenimos vulnerabilidades XSS en la generación de esa página. [39]

5. **Validación de todas las fuentes de entrada no confiables**: Consideramos todos los vectores de entrada (parámetros URL, cookies, encabezados, formularios ocultos, variables de entorno, contenido desde APIs, correos, bases de datos, etc.) y aplicamos validación rigurosa. [105]

6. **No confiar exclusivamente en listas negras**: Las listas de denegación pueden ayudarnos a filtrar patrones maliciosos conocidos, pero no son suficientes debido a nuevas formas de ataque. Debemos usarlas solo como apoyo adicional. [37]

8.1.2.2 Mitigación de ausencia de protección CSRF

URL	http://192.168.1.226/nwset/
Método	GET
Ataque	
Evidencia	<form name="frm" action="top.cgi" method="POST">
Otra información	No se ha encontrado ningún token Anti-CSRF [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] conocido en el siguiente formulario HTML: [Form 1: "admin_pass" "btn"].

Figura 8.28. *Ausencia de protección CSRF*

Para proteger una página web contra la ausencia de protección CSRF, recomendamos las siguientes medidas:

1. **Identificación de operaciones sensibles y exigencia de confirmación explícita**: Para operaciones de alto riesgo (por ejemplo, modificación de datos, transacciones, configuración crítica), implementamos un paso adicional donde el usuario confirme su intención (por ejemplo, mediante un diálogo o una página intermedia). Esto nos ayuda a impedir que una solicitud maliciosa ejecutada desde XSS pase inadvertida. [140]

2. **Evitar el uso de métodos GET para acciones que cambian el estado**: Según OWASP, las operaciones que modifican el estado del servidor debemos implementarlas exclusivamente a través de peticiones POST, NO GET, para evitar ataques inadvertidos mediante etiquetas como <img> o redirecciones automáticas. [104]

3. **Uso del control de gestión de sesión de ESAPI con soporte para CSRF**: El módulo de gestión de sesión de ESAPI incluye automáticamente un token CSRF que validamos en cada solicitud sensible. Este mecanismo asegura que solo solicitudes legítimas, originadas desde la sesión activa del usuario, sean aceptadas. [140]

4. **Asegurar que la solicitud provenga de la página esperada (validación de Referer/Origen)**: Debemos verificar que la petición sensible provenga desde la página legítima. Si bien las cabeceras Referer o Origin pueden suprimirse por motivos de privacidad o romper funcionalidad al estar desactivadas, su ausencia también podemos considerarla indicio de ataque y tratarla como sospechosa. [104]

5. **Prevención de bypass mediante XSS**: Las protecciones anti-CSRF pueden ser anuladas si existe una vulnerabilidad XSS. Por lo tanto, es crucial mitigar XSS en todas las áreas de entrada no confiables, garantizando que los tokens CSRF no sean expuestos o robados. [38]

URL	http://192.168.1.22/general/status.html
Método	POST
Ataque	
Evidencia	
Otra información	

Figura 8.29. Ausencia de protección CSP

8.1.2.3 Mitigación de cabecera CSP ausente o débil

Para proteger una página web contra la ausencia de protección CSP, recomendamos las siguientes medidas:

1. **Inyección de la cabecera CSP desde el servidor**: Configuramos el servidor para enviar una cabecera `Content-Security-Policy` en todas las respuestas HTTP (no solo en la página principal) para especificar qué recursos pueden cargarse. [123]
2. **Uso de política estricta con nonces o hashes**: Implementamos una política CSP estricta utilizando nonces (valores únicos por respuesta) o hashes para autorizar solo scripts específicos y prevenir la ejecución de código inyectado. [123]
3. **Restricción de carga de recursos externos**: Configuramos directivas como `default-src 'self'` y especificamos fuentes permitidas para scripts, estilos, imágenes y conexiones, reduciendo la superficie de ataque. [45]
4. **Prevención de ataques como XSS, clickjacking y exfiltración de datos**: Con una política CSP bien implementada, limitamos la ejecución de scripts no autorizados, protegemos contra la carga fuera de origen y evitamos que datos sean enviados a servidores maliciosos. [123]

8.1.2.4 Mitigación de exposición de listado de directorios

Para proteger una pagina web contra posibilidad de ver la explotación de directorios, recomendamos las siguientes medidas:

1. **Deshabilitar el listado de directorios en el servidor web**: Configuramos el servidor para que no muestre el contenido de un directorio si no existe un archivo índice. En Apache, por ejemplo, eliminamos la directiva `Indexes` o usamos `Options -Indexes`. En Nginx, establecemos `autoindex off`. En Tomcat desactivamos la opción `listings` en `web.xml`. [42]
2. **Restringir acceso a directorios sensibles mediante permisos y control de acceso**: Implementamos seguridad a nivel de servidor o sistema de archivos para asegurar que directorios críticos no sean accesibles públicamente. [42]
3. **Evitar almacenamiento público de archivos sensibles o backups**: No colocamos archivos de configuración, respaldos o metadatos dentro del directorio raíz público o en directorios accesibles sin protección. [42]

URL	http://192.168.1.226/license.txt/

Método	GET
Ataque	http://192.168.1.226/license.txt/
Evidencia	parent directory

***Figura 8.30**. Exposición de listado de directorios*

URL	http://192.168.1.22/admin/password.html
Método	GET
Ataque	
Evidencia	Set-Cookie: AuthCookie
Otra información	

***Figura 8.31**. Cookies sin el atributo Samesite*

8.1.2.5 Mitigación de Cookie sin el atributo Samesite

Para proteger una página web contra la posibilidad de que una cookie no tenga el atributo SameSite, recomendamos las siguientes medidas:

1. **Establecer explícitamente el atributo `SameSite` (idealmente `Lax` o `Strict`)**: Esto asegura que la cookie solo se envíe en contextos seguros. `Lax` ofrece un equilibrio aceptable para la mayoría de aplicaciones, mientras que `Strict` brinda máxima protección. [6]
2. **Configurar `SameSite=None` solo con la bandera `Secure`**: Cuando necesitamos enviar cookies en contextos entre sitios, nos aseguramos de incluir también `Secure`, ya que muchos navegadores modernos rechazan cookies `SameSite=None` sin esta directiva. [34]
3. **Entender la protección que ofrece `SameSite` contra CSRF**: Este atributo impide que navegadores envíen cookies sensibles en solicitudes iniciadas desde dominios externos, lo que refuerza la defensa frente a CSRF. [36]
4. **Evitar el comportamiento inseguro por defecto en navegadores**: Aunque algunos navegadores pueden aplicar SameSite = Lax por defecto si no se especifica, la configuración explícita por nuestra parte garantiza el comportamiento esperado y evita fugas accidentales. [6]
5. **Mitigar riesgos complementarios, como CSRF y exfiltración**: Establecer `SameSite = Lax / Strict` nos ayuda a evitar que cookies sean enviadas automáticamente en contextos cross-site, reduciendo significativamente los riesgos de CSRF y fugas de datos. [34, 36]

8.1.2.6 Mitigación: ocultar la versión en la cabecera `Server` de la respuesta HTTP

URL	http://192.168.1.22/common/css/common.css
Método	GET
Ataque	
Evidencia	debut/1.30

Figura 8.32. *Servidor filtra información HTTP*

Para proteger una página web contra la revelación de información sensible del servidor, recomendamos las siguientes medidas:

1. **Apache: limitar la cabecera `Server`** con `ServerTokens Prod` y `ServerSignature Off`, de modo que solo muestre "Apache" sin versión. [33]
2. **Nginx: desactivar la exposición de versión** con `server_tokens off;`. Para eliminar totalmente la cabecera `Server`, usamos un módulo como *Headers More* o lo hacemos a través de un proxy inverso/WAF. [125]

3. **Eliminar/reescribir la cabecera en un WAF o proxy inverso** (p.ej., ModSecurity, Nginx/Traefik/Envoy al frente) cuando el servidor origen no pueda cambiarse. [33, 125]

4. **Aplicar la medida también a páginas de error** (404, 500, etc.), donde la cabecera `Server` suele persistir. [125]

8.1.2.7 Falta encabezado X-Content-Type-Options

URL	http://192.168.1.22/general/reflesh.html
Método	POST
Ataque	
Evidencia	

Figura 8.33. *Falta encabezado X-Content-Type-Options*

Para proteger una página web contra la falta del encabezado X-Content-Type-Options, recomendamos las siguientes medidas:

1. **Agregar `X-Content-Type-Options: nosniff` en todas las respuestas HTTP**, incluyendo páginas de error (ej. 401, 403, 500), para forzar que el navegador respete el tipo de contenido declarado. [48]

2. **Declarar correctamente el encabezado `Content-Type` en todas las páginas**. Esto ayuda a prevenir que el navegador realice MIME-sniffing por falta de información o ambigüedad. [60]

3. **Evitar ataques de tipo MIME-sniffing**. Si un atacante sube un archivo malicioso con extensión de imagen, podría ser interpretado como ejecutable si no se controla el comportamiento del navegador. [49]

8.1.2.8 Gran redirección detectada (posible fuga de información confidencial)

URL	http://192.168.1.22/general/status.html
Método	POST
Ataque	
Evidencia	
Otra información	Longitud URI de la cabecera de ubicación: 50 [/etc/passerror.html?url=%2Fgeneral%2Fstatus%2Ehtml]. Tamaño previsto de la respuesta: 350. Longitud del cuerpo de la respuesta: 7,579.

Figura 8.34. *Gran redirección detectada (posible fuga de información confidencial)*

Para proteger una página web contra la gran redirección detectada (redirecciones inseguras), recomendamos las siguientes medidas:

1. **Validación estricta de destinos de redirección**: Solo permitimos redirecciones a destinos predefinidos o internos autorizados. Utilizamos una lista blanca en lugar de depender de parámetros enviados por el usuario. [62]

2. **Mapeo mediante identificadores internos**: En vez de aceptar rutas o URLs directamente, usamos IDs fijos que sean traducidos internamente a URLs seguras. Esto impide que un usuario manipule el destino con rutas arbitrarias. [39]

3. **Firewall de aplicaciones o WAF**: Usamos un firewall para detectar y bloquear redirecciones inseguras o poco comunes, especialmente cuando no se pueden corregir rápidamente en el código. [62]

4. **Página intermedia de advertencia para redirección externa**: Si detectamos que el destino de la redirección es externo o no confiable, mostramos al usuario una página intermedia que indique claramente a dónde se dirige y solicitamos confirmación antes de proceder. [39]

8.1.2.9 Mitigación de revelación de Ip privada

URL	http://192.168.1.22/general/find.html
Método	GET
Ataque	
Evidencia	192.168.1.25
Otra información	192.168.1.25 192.168.1.25 192.168.1.30 192.168.1.30 192.168.1.31 192.168.1.31 192.168.1.32 192.168.1.32 192.168.1.33 192.168.1.33 192.168.1.37 192.168.1.37 192.168.1.203 192.168.1.203 192.168.1.208 192.168.1.208 192.168.1.101 192.168.1.199 192.168.1.210

Figura 8.35. Fuga de información confidencial a través de la IP privada

Para proteger una página web contra la revelación de IP privada, recomendamos las siguientes medidas:

1. **Eliminar IPs privadas del contenido de respuesta**. Nos aseguramos de que no se filtren direcciones IP internas o nombres de host en el código visible al cliente (por ejemplo, en páginas, errores o mensajes de debug). [114]

2. **Utilizar comentarios del lado del servidor**. En lugar de insertar información sensible en comentarios HTML o JavaScript, usamos comentarios en tecnologías como JSP, ASP o PHP, que no serán visibles para el navegador del cliente. [114]

3. **Evitar exposición en mensajes generados dinámicamente**. Las IP internas no deben aparecer tampoco en logs, mensajes de error, trazas o vistas de depuración que puedan ser visibles en el entorno de producción. [114]

8.1.2.10 Mitigación de autenticación débil

URL	http://192.168.1.210/hp/device/info_config_AirPrint.html?menu=AirPrintStatus&tab=Networking
Método	GET
Ataque	
Evidencia	www-authenticate: Basic realm="HP LaserJet device (password only, no username required)@NPID14F94"
Otra información	

Figura 8.36. Fuga de información confidencial a través de la autenticación débil

Para proteger una página web contra la autenticación débil, recomendamos las siguientes medidas (con citas):

1. **Forzar uso de HTTPS (TLS) en todo el sitio**, redirigiendo todo HTTP a HTTPS. Usamos HSTS para que el navegador insista en HTTPS incluso si se escribe manualmente HTTP. [16]

2. **Evitar usar Basic o Digest como método principal**, priorizando autenticación basada en sesiones (cookies seguras) o tokens (Bearer, OAuth2) bajo HTTPS. [137]

8.1.2.11 Mitigación de credenciales capturadas

URL	http://192.168.1.210/set_config_password.html?tab=System&menu=Passwd
Método	GET
Ataque	
Evidencia	
Otra información	[GET] [http://192.168.1.210/set_config_password.html?tab=System&menu=Passwd] utiliza el mecanismo de autenticación inseguro [Basic], revelando el nombre de usuario [admin] y contraseña [].

Figura 8.37. *Fuga de contraseñas a través de un ataque*

Para proteger una página web contra las filtraciones de credenciales, recomendamos las siguientes medidas:

1. Implementamos el uso de protocolos seguros como HTTPS en lugar de HTTP para garantizar el cifrado de las credenciales en tránsito. [60]

2. Sustituimos la autenticación básica o Digest por mecanismos más seguros como OAuth 2.0, JWT o autenticación basada en certificados. [92]

3. Configuramos el servidor para que todas las comunicaciones requieran el uso de TLS 1.2 o superior, evitando versiones inseguras. [60]

4. Empleamos autenticación multifactor (MFA) para añadir una capa adicional de seguridad frente a robo de credenciales. [92]

5. Rotamos periódicamente las credenciales y aplicamos políticas de contraseñas robustas (longitud, complejidad y caducidad). [92]

6. Monitorizamos intentos de acceso sospechosos y aplicamos sistemas de detección de intrusiones (IDS/IPS) para detectar ataques Man-In-The-Middle. [60]

8.1.2.12 Mitigación de Librería JS vulnerable

Para proteger una página web contra el uso de una librería JS vulnerable, recomendamos las siguientes medidas:

1. **Actualizar a una versión segura**, como jQuery 3.5.0 o superior, donde se corrigieron vulnerabilidades como CVE-2020-11023. [28]

2. **Sanitizar HTML de entrada de usuario** antes de insertarlo en el DOM (por ejemplo, usando DOMPurify antes de .html()), para evitar inyección de código malicioso. [93]

URL	http://192.168.1.210/hp/device/jquery.js
Método	GET
Ataque	
Evidencia	* jQuery JavaScript Library v1.3.2

Figura 8.38. *Fuga de información confidencial a través de la librería JS vulnerable*

8.1.2.13 Mitigación de Inyección Remota de Comandos del Sistema Operativo

URL	http://192.168.1.199/web/guest/es/websys/webArch/waSearchFirst.cgi?wimToken=251565448%3Bsleep+1.0%3B
Método	GET
Ataque	251565448;sleep 15;
Evidencia	
Otra información	La regla de escaneo pudo controlar el tiempo de respuesta de la aplicación enviando [251565448;sleep 15;] al sistema operativo que ejecuta esta aplicación.

Figura 8.39. *Inyección remota de comandos del sistema operativo*

Para proteger una pagina web contra la inyección remota de comandos del sistema operativo, recomendamos las siguientes medidas:

1. Evitamos por completo la ejecución de comandos del sistema operativo desde el código de la aplicación cuando sea posible. Utilizamos APIs seguras o bibliotecas del lenguaje para lograr la funcionalidad requerida sin invocar comandos externos. [59]
2. Aplicamos validación fuerte de la entrada del usuario: usamos listas blancas (whitelists) de valores permitidos, nos aseguramos de que la entrada tiene el formato esperado (por ejemplo, números, rutas seguras, direcciones IP) y rechazamos cualquier dato que no coincida. [61]
3. Empleamos sanitización segura, usando funciones como `escapeshellarg()` (en PHP) u equivalentes, que evitan que metacaracteres de shell sean interpretados como comandos. [59]
4. Rechazamos o sanitizamos expresamente metacaracteres peligrosos como `;`, , como parte de la validación o sanitización. [59]
5. Adoptamos un enfoque de "Seguridad desde el Diseño": durante el desarrollo, evitamos funciones peligrosas como `system()` o similares, y las sustituimos por alternativas más seguras. [7]

8.1.2.14 Mitigación de falta encabezado AnticlickJacking

Para proteger una página web contra la falta de cabecera Anti-ClickJacking, recomendamos las siguientes medidas:

1. Usamos la cabecera `Content-Security-Policy` con la directiva `frame-ancestors`, por ejemplo:

```
Content-Security-Policy: frame-ancestors 'none';
```

   Esto bloquea completamente que la página sea mostrada dentro de un marco externo. [41]

URL	http://192.168.1.199/web/entry/es/address/adrsList.cgi
Método	GET
Ataque	
Evidencia	
Otra información	

Figura 8.40. *Falta encabezado AnticlickJacking*

2. Tenemos presente que la directiva CSP `frame-ancestors` tiene prioridad sobre `X-Frame-Options` en navegadores modernos, de modo que quienes soportan CSP ignorarán la directiva X-Frame-Options cuando ambas estén presentes. [31]

8.1.2.15 Mitigación de mostrar errores de depuración

URL	http://192.168.1.199/web/guest/es/websys/webArch/login.cgi
Método	GET
Ataque	
Evidencia	Internal Server Error
Otra información	

Figura 8.41. *Mostrar errores de depuración*

Para proteger una página web contra la exposición de errores de depuración, recomendamos las siguientes medidas:

1. Desactivamos el modo de depuración en producción (por ejemplo, `debug="false"` en ASP.NET). [71]
2. Mostramos mensajes de error genéricos al usuario, sin detalles internos. [71]
3. Registramos los errores detallados solo en logs internos seguros. [71]
4. Configuramos páginas de error personalizadas (404, 500, etc.) en el servidor o aplicación. [71]

8.1.2.16 Mitigación de mostrar errores de aplicación

Para proteger una pagina web contra la exposición de errores de divulgación, recomendamos las siguientes medidas:

1. No mostramos mensajes de error completos (como rutas de archivo o stack trace) al usuario. [71]
2. Mostramos errores genéricos al usuario final y guardamos detalles técnicos solo en logs internos seguros. [71]
3. Configuramos páginas de error personalizadas (por ejemplo, 500, fallo inesperado) en la aplicación o servidor. [71]

URL	http://192.168.1.199/web/entry/es/webdocbox
Método	GET
Ataque	
Evidencia	HTTP/1.1 500 Internal Server Error

Figura 8.42. Mostrar errores de divulgación

8.1.2.17 Mitigación de divulga información mediante un campo(s) de encabezado de respuesta HTTP X-Powered-By

URL	http://192.168.1.2/
Método	GET
Ataque	
Evidencia	X-Powered-By: ASP.NET

Figura 8.43. Mostrar errores de divulgación

Para proteger una pagina web contra la exposición de errores de divulgación, recomendamos las siguientes medidas:

- Quitamos el encabezado `X-Powered-By` desde el IIS Manager, sección "HTTP Response Headers". [71]
- En el archivo `web.config`, bajo la sección `system.webServer`, usamos `customHeaders` para eliminar el encabezado `X-Powered-By`. [71]
- Deshabilitamos encabezados de versión como `X-AspNet-Version` mediante la propiedad `enableVersionHeader = false` en la sección `httpRuntime`. [71]
- Eliminamos/ocultamos el encabezado `Server` usando la configuración de IIS (por ejemplo, en IIS 10 mediante `requestFiltering removeServerHeader = true`). [71]
- Usamos reglas de reescritura o un módulo/middleware que intercepte la respuesta antes de enviarla al cliente y elimine los encabezados sensibles. [71]

8.1.2.18 Mitigación de Inyección SQL

Para reducir el riesgo de inyección SQL, según lo recomendado por OWASP:

- Usamos consultas parametrizadas (prepared statements), de modo que los valores del usuario se traten como datos y no se concatenen al comando SQL. [100]
- Empleamos procedimientos almacenados con parámetros (sin construcción dinámica de SQL). [100]
- Validamos ("saneamiento") de entrada: permitimos solo los formatos esperados y rechazamos o filtramos caracteres peligrosos. [100]

Alto	Inyección SQL
Descripción	Inyección SQL puede ser posible.
URL	https://gestoriaofiauto.es/wp-login.php?action=lostpassword%27+AND+%271%27%3D%271%27+--+
Método	GET
Ataque	lostpassword' AND '1'='1' --
Evidencia	
Otra información	Los resultados de la página fueron manipulados con éxito utilizando las condiciones booleanas [lostpassword' AND '1'='1' --] y [lostpassword' AND '1'='2' --]. El valor del parámetro modificado fue , que fue eliminado del HTML para facilitar la comparación. Se devolvieron datos para el parámetro original. La vulnerabilidad se detectó al restringir con éxito los datos que se devolvían originalmente, manipulando el parámetro.

Figura 8.44. *Vulnerabilidad de inyección SQL*

- Aplicamos el principio de mínimo privilegio a la cuenta que accede a la base de datos (solo permisos necesarios). [100]

8.1.3 Vulnerabilidades encontradas en Impresoras y Escaners

- **Impresora HL-L5100DN**

Nombre	Nivel de riesgo
Redirección Externa	Alto
Ausencia de Tokens Anti-CSRF	Medio
Cabecera Content Security Policy (CSP) no configurada	Medio
Directory Browsing (Exploración de directorios)	Medio
Cookie sin el atributo SameSite	Bajo
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo
Falta encabezado X-Content-Type-Options	Bajo
Gran redirección detectada (posible fuga de información confidencial)	Bajo
Revelación de IP privada	Bajo

Figura 8.45. *Vulnerabilidades en Impresora HL-L5100DN*

- Impresora HP Laserjet M404DN

Nombre	Nivel de riesgo
Credenciales de Autenticación Capturadas	Alto
Ausencia de Tokens Anti-CSRF	Medio
Cabecera Content Security Policy (CSP) no configurada	Medio
Librería JS Vulnerable	Medio
Método de autenticación débil	Medio
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo
Revelación de IP privada	Bajo

Figura 8.46. *Vulnerabilidades en Impresora HP Laserjet M404DN*

- Impresora Brother HL-L5210-DN

Nombre	Nivel de riesgo
Ausencia de Tokens Anti-CSRF	Medio
Cabecera Content Security Policy (CSP) no configurada	Medio
Falta encabezado X-Content-Type-Options	Bajo

Figura 8.47. *Vulnerabilidades en Impresora Brother HL-L5210-DN*

- Impresora Brother HL-L5100DN Modelo 2

Nombre	Nivel de riesgo
Cabecera Content Security Policy (CSP) no configurada	Medio
Falta encabezado X-Content-Type-Options	Bajo
Gran redirección detectada (posible fuga de información confidencial)	Bajo
Revelación de IP privada	Bajo

Figura 8.48. *Vulnerabilidades en Impresora Brother HL-L5100DN Modelo 2*

- Impresora Brother HL-L5200DN

Nombre	Nivel de riesgo
Ausencia de Tokens Anti-CSRF	Medio
Cabecera Content Security Policy (CSP) no configurada	Medio
Falta encabezado X-Content-Type-Options	Bajo

Figura 8.49. *Vulnerabilidades en Impresora Brother HL-L5200DN*

- Escaner PFU

Nombre	Nivel de riesg
Cabecera Content Security Policy (CSP) no configurada	Medio

Figura 8.50. *Vulnerabilidades en Escaner PFU*

- Escaner PFU Modelo 2

Nombre	Nivel de riesgo
Ausencia de Tokens Anti-CSRF	Medio
Cabecera Content Security Policy (CSP) no configurada	Medio
Directory Browsing (Exploración de directorios)	Medio
Cookie sin el atributo SameSite	Bajo
Falta encabezado X-Content-Type-Options	Bajo
Revelación de IP privada	Bajo

Figura 8.51. *Vulnerabilidades en Escaner PFU Modelo 2*

- Impresora Ricoh MP C401SR y Ricoh MP C401SR Modelo 2

Nombre	Nivel de riesgo
Remote OS Command Injection (Inyección Remota de Comandos del Sistema Operativo)	Alto
Ausencia de Tokens Anti-CSRF	Medio
Cabecera Content Security Policy (CSP) no configurada	Medio
Falta de cabecera Anti-Clickjacking	Medio
Librería JS Vulnerable	Medio
Cookie Sin Flag HttpOnly	Bajo
Cookie sin el atributo SameSite	Bajo
Divulgación de Información - Mensajes de Error de Depuración	Bajo
Divulgación de error de aplicación	Bajo
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo
Falta encabezado X-Content-Type-Options	Bajo

Figura 8.52. *Vulnerabilidades en Impresora Ricoh MP C401SR*

8.1.4 Vulnerabilidades encontradas en las Cámaras

- Cámara GrandStream Network

Nombre	Nivel de riesgo
Cabecera Content Security Policy (CSP) no configurada	Medio
Falta de cabecera Anti-Clickjacking	Medio
Divulgación de Marcas de Tiempo - Unix	Bajo
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo
Falta encabezado X-Content-Type-Options	Bajo
Revelación de IP privada	Bajo

Figura 8.53. *Vulnerabilidades en Cámara GrandStream Network*

8.1.5 Vulnerabilidades encontradas en los dispositivos de red

- Millenial Net

- TpLink

Nombre	Nivel de riesgo
Cabecera Content Security Policy (CSP) no configurada	Medio
Falta de cabecera Anti-Clickjacking	Medio
Librería JS Vulnerable	Medio
Divulgación de Marcas de Tiempo - Unix	Bajo
Falta encabezado X-Content-Type-Options	Bajo

Figura 8.54. Vulnerabilidades en Millenial Net

- No detectamos vulnerabilidades en este dispositivo.

8.1.6 Vulnerabilidades encontradas en Ordenadores

- **Hewlett Packward y Hon Hai Precision**
  - No detectamos vulnerabilidades en estos ordenadores.
- **Hewlett Packard Enterprise**

Nombre	Nivel de riesgo
Cabecera Content Security Policy (CSP) no configurada	Medio
Divulgación de error de aplicación	Medio
Librería JS Vulnerable	Medio
Cookie Sin Flag HttpOnly	Bajo
Cookie sin el atributo SameSite	Bajo
Divulgación de Información - Mensajes de Error de Depuración	Bajo
Falta encabezado X-Content-Type-Options	Bajo
Revelación de IP privada	Bajo
Strict-Transport-Security Header No Establecido	Bajo

Figura 8.55. Vulnerabilidades en Hewlett Packard Enterprise

8.1.7 Vulnerabilidades encontradas en IIS

- **IIS Asustek Compute,Dell, HewlettPackard Modelo 2, HewlettPackard Modelo 3, Dell Modelo 2**
  - No detectamos vulnerabilidades en estos dispositivos.

- IIS Hewlett Packard

Nombre	Nivel de riesgo
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo

Figura 8.56. Vulnerabilidades en IIS Hewlett Packard

- IIS Hewlett Packard Modelo 4

Nombre	Nivel de riesgo
Cabecera Content Security Policy (CSP) no configurada	Medio
Credenciales de Autenticación Capturadas	Medio
Falta de cabecera Anti-Clickjacking	Medio
Método de autenticación débil	Medio
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo
Falta encabezado X-Content-Type-Options	Bajo
Tecnología Detectada - Digest	Informativo
Tecnología Detectada - Intel Active Management Technology	Informativo

Figura 8.57. Vulnerabilidades en IIS Hewlett Packard Modelo 4

- IIS Hewlett Packard Modelo 5

Nombre	Nivel de riesgo
Cabecera Content Security Policy (CSP) no configurada	Medio
Falta de cabecera Anti-Clickjacking	Medio
Método de autenticación débil	Medio
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo
Falta encabezado X-Content-Type-Options	Bajo
Tecnología Detectada - Digest	Informativo
Tecnología Detectada - Intel Active Management Technology	Informativo

Figura 8.58. Vulnerabilidades en IIS Hewlett Packard Modelo 5

8.1.8 Vulnerabilidades encontradas en Servidores y Bases de Datos

- Servidor Hewlett Packard

- **Base de Datos** no detectamos vulnerabilidades en este dispositivo.

Nombre	Nivel de riesgo	Número de Instancias
Cabecera Content Security Policy (CSP) no configurada	Medio	77
Credenciales de Autenticación Capturadas	Medio	1
Falta de cabecera Anti-Clickjacking	Medio	6
Librería JS Vulnerable	Medio	1
Método de autenticación débil	Medio	5
Divulgación de Marcas de Tiempo - Unix	Bajo	20
El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By'""	Bajo	4
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo	60
Falta encabezado X-Content-Type-Options	Bajo	69
Revelación de IP privada	Bajo	1

Figura 8.59. Vulnerabilidades en Servidor Hewlett Packard

8.1.9 Vulnerabilidades encontradas en la Pagina Web

- **Página web**

Nombre	Nivel de riesgo
Inyección SQL	Alto
Ausencia de Tokens Anti-CSRF	Medio
Cabecera Content Security Policy (CSP) no configurada	Medio
Falta de cabecera Anti-Clickjacking	Medio
Cookie sin el atributo SameSite	Bajo
Divulgación de Marcas de Tiempo - Unix	Bajo
Falta encabezado X-Content-Type-Options	Bajo
Strict-Transport-Security Header No Establecido	Bajo

Figura 8.60. Vulnerabilidades en Página Web

8.2 Sqlmap

Sqlmap es una herramienta de código abierto que utilizamos para la detección y explotación automatizada de vulnerabilidades de inyección SQL en aplicaciones web. En este proyecto, la autorización de pruebas se limitó exclusivamente a la **base de datos de la página web** auditada. Tras ejecutar las pruebas permitidas con *sqlmap* sobre los endpoints autorizados, que correspondían a la página web, **no identificamos vulnerabilidades de inyección SQL explotables** ni evidencias de compromiso en la base de datos evaluada.

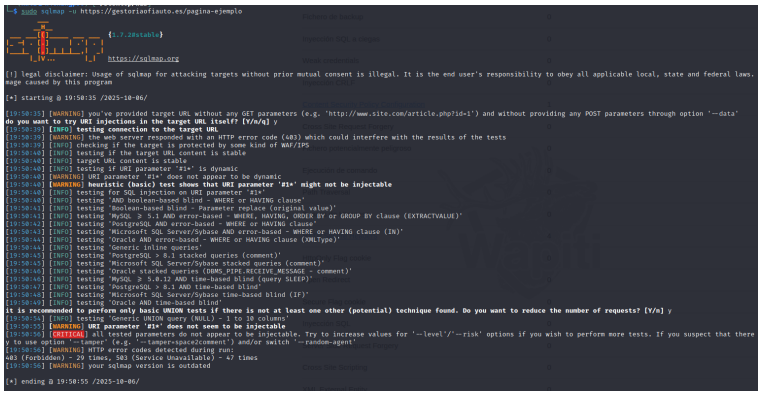


Figura 8.61. *Imagen de prueba sqlmap en página web con ausencia de vulnerabilidades SQL*

8.3 Nikto

En esta fase utilizamos la herramienta *Nikto* en su versión 2.5.0 para llevar a cabo un escaneo de vulnerabilidades web. El análisis nos permitió identificar cabeceras de seguridad ausentes, configuraciones inseguras y posibles puntos de ataque. Con ello obtuvimos una visión inicial de la superficie expuesta de los servidores analizados. Mediante la generación de informes detallados en formato HTML, facilitamos la revisión y el análisis posterior de los hallazgos.

El código que empleamos para el análisis de cada host fue el siguiente:

sudo nikto -h ip -Format html -o NombreHost.html

Donde *ip-Format* corresponde a la dirección IP del host que analizamos y *NombreHost* es el nombre que asignamos al informe generado.

8.3.1 Vulnerabilidades detectadas en la fase de Auditorias de Aplicaciones Web de Nikto

En esta sección detallamos las vulnerabilidades más relevantes detectadas en la fase de Auditorias de Aplicaciones Web de Nikto.

8.3.1.1 Mitigación de Falta de Cabecera sin tipo de contenido

Para proteger un servidor web contra cabecera sin tipo de contenido, recomendamos las siguientes medidas:

URI	/
HTTP Method	GET
Description	/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
Test Links	http://192.168.1.37:80/ http://192.168.1.37:80/
References	https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/

Figura 8.62. *Falta de Cabecera*

- Asegurar que el servidor web o servicio embebido en la impresora incluya siempre la cabecera `Content-Type` correcta y específica para cada recurso (por ejemplo, `application/ipp`, `text/html; charset=UTF-8`). [48]
- Incluir la cabecera `X-Content-Type-Options: nosniff` para evitar que los navegadores realicen MIME-sniffing y puedan interpretar incorrectamente (y quizá ejecutar) el contenido. [60]
- Verificar que todas las respuestas HTTP (tanto GET como POST) desde la impresora contienen efectivamente la cabecera `Content-Type`, especialmente en servicios como CUPS que manejan tipos específicos como `application/ipp`. [49]

8.3.1.2 Mitigación de vulnerabilidad de métodos permitidos

URI	/
HTTP Method	OPTIONS
Description	OPTIONS: Allowed HTTP Methods: POST, OPTIONS .
Test Links	http://192.168.1.37:80/ http://192.168.1.37:80/
References	

Figura 8.63. *Métodos permitidos*

Para mitigar las vulnerabilidades de metodos permitidos, recomendamos seguir estas medidas:

- Restringir los métodos HTTP permitidos a los estrictamente necesarios (por ejemplo, solo `POST` y `GET`), y devolver `405 Method Not Allowed` para los demás [138].
- Deshabilitar o proteger métodos potencialmente peligrosos como `PUT`, `DELETE`, `TRACE`, `CONNECT` que no sean requeridos por la impresora [136, 138].
- Si no es necesario, considerar desactivar el método `OPTIONS`, ya que puede facilitar la enumeración de métodos habilitados y aumentar la superficie de ataque. Solo mantenerlo activo si hay una razón funcional clara, como CORS o APIs REST [138].

Mitigación de falta de configuración de la cabecera X-Frame-Options

Para protegernos contra la vulnerabilidad de falta de configuración de la cabecera, recomendamos estas medidas mencionadas anteriormente en 8.40.

URI	/
HTTP Method	GET
Description	/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
Test Links	http://192.168.1.198:80/ http://192.168.1.198:80/
References	https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/

Figura 8.64. *Clickjacking*

URI	/#wp-config.php#
HTTP Method	GET
Description	/#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
Test Links	http://192.168.1.226:80/#wp-config.php# http://192.168.1.226:80/#wp-config.php#
References	

Figura 8.65. *Wp-Config*

8.3.1.3 Mitigación de fuga de información por acceso a un archivo con credenciales en wp-config.php

Para prevenir la fuga de información por acceso a un archivo con credenciales, recomendamos las siguientes acciones:

- Mover el archivo `wp-config.php` uno o más niveles por encima del directorio raíz (web root), de modo que WordPress todavía lo encuentre, pero que no sea accesible directamente desde web [119].
- Configurar el servidor (Apache, Nginx, etc.) o el archivo `.htaccess` para denegar explícitamente el acceso HTTP al archivo, generando una respuesta `403 Forbidden` [119, 148].
- Establecer permisos restrictivos en `wp-config.php`, como 400 o 600, limitando su lectura exclusivamente al propietario [119].
- Evitar renombrar el archivo mediante sufijos como `.bak` o `.old`, ya que esto puede impedir que se interprete como PHP y permitir su descarga en texto plano [119].
- Verificar que no existan vulnerabilidades de inclusión de archivos (LFI) o plugins inseguros que puedan exponer el contenido de `wp-config.php` [148].

8.3.1.4 Mitigación Cookie establiecida sin el atributo HttpOnly y Secure

URI	/
HTTP Method	GET
Description	/: Cookie cookieOnOffChecker created without the httponly flag.
Test Links	http://192.168.1.101:80/ http://192.168.1.101:80/
References	https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies

Figura 8.66. *Cookie On Off*

Para prevenir la vulnerabilidad de la cookie, recomendamos seguir las siguientes medidas:

- Establecer siempre el atributo `HttpOnly` en cookies sensibles (por ejemplo, de sesión o autenticación), para impedir que JavaScript pueda acceder a su valor [44, 103].
- Enviar todas las cookies junto con el atributo `Secure`, garantizando que solo se transmitan a través de conexiones HTTPS seguras [44, 103].
- Configurar atributos adicionales como `SameSite=Strict` o `SameSite=Lax` para mitigar ataques tipo CSRF [44, 103].
- Revisar y asegurar que las cookies sensibles expiren lo antes posible mediante `Max-Age` o `Expires` [44].
- Ajustar los atributos de `Domain` y `Path` al valor más restrictivo necesario para el funcionamiento, limitando el alcance de envío de cookies [44, 103].

8.3.1.5 Mitigación de revelación de información del servidor

Figura 8.67. *Revelación de Información - Cambio de Servidor*

Para prevenir la vulnerabilidad de la revelación de información del servidor , recomendamos seguir las siguientes medidas:

- Ocultar o modificar la cabecera `Server` para no revelar el nombre exacto o versión del servidor (por ejemplo, usar un valor genérico como "Server: secure-device"). [71]
- Configurar el servidor, si es posible, para desactivar la divulgación de versión mediante directivas como `servertokens off` en Nginx u opciones equivalentes en Apache. [71]
- Recompilar el software del servidor (como Nginx) con módulos que permitan el control del valor de la cabecera `Server`, como `HttpHeadersMoreModule`. [71]
- Usar un dispositivo intermedio (p. ej. proxy inverso o WAF) que interfiera la cabecera `Server`, reemplazándola o eliminándola antes de que llegue al cliente. [71]

8.3.1.6 Mitigación de revelación de información por medio de UPnP

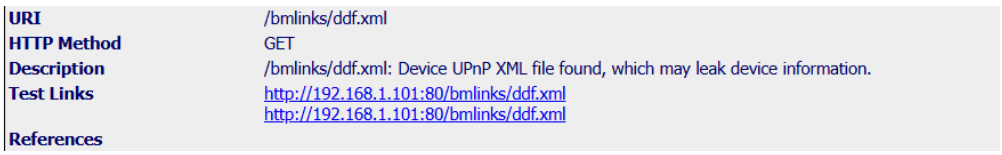


Figura 8.68. *Revelación de Información - UPnP*

Para prevenir la vulnerabilidad de la revelación de información del servidor, recomendamos seguir las siguientes medidas:

- Deshabilitar UPnP completamente si no es necesario, especialmente en routers o impresoras con interfaz web, reduciendo la superficie de ataque [121].
- Evitar que los archivos XML de descripción UPnP como `ddf.xml` estén accesibles externamente, restringiéndolos solo a la red local mediante configuración o firewall [134].
- Segmentar la red y aislar los dispositivos con UPnP en una VLAN dedicada, minimizando el impacto ante una posible explotación [121].
- Utilizar firewalls o IPS/WAF para bloquear tráfico UPnP o SSDP (puerto UDP 1900), impidiendo consultas remotas o automatizadas [121, 134].

8.3.1.7 Mitigación de cabecera Strict-Transport-Security (HSTS) no esté definida para TLS

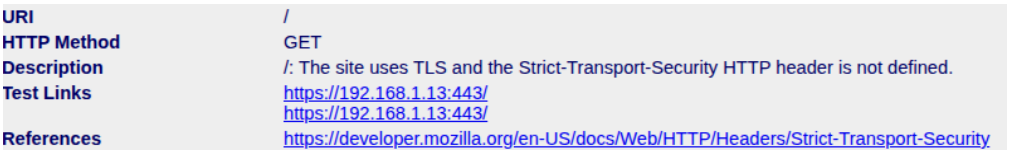


Figura 8.69. *Revelación de cabecera Strict-Transport-Security (HSTS)*

Para prevenir la vulnerabilidad de que la cabecera `Strict-Transport-Security` (HSTS) no esté definida para TLS, recomendamos seguir las siguientes medidas:

- Revisar que todos los dominios y subdominios del sitio tengan certificados TLS válidos, sin errores (nombre, caducidad, CA de confianza), ya que HSTS solo funciona correctamente si HTTPS está bien configurado [99, 19].
- Evitar contenido mixto ("mixed content"): asegurarnos de que todas las imágenes, scripts, hojas de estilo, recursos externos, etc., se carguen mediante HTTPS, para que no haya desestabilización de la página debido a recursos inseguros [70, 99].
- Implementar redirecciones permanentes (301) desde HTTP a HTTPS, de modo que no se permita acceso inseguro público y asegurar que las peticiones HTTP siempre acaben usando HTTPS antes de servir contenido [99, 19].

8.3.1.8 Mitigación de -AspNet-Version header: 4.0.30319

Figura 8.70. *Revelación de cabecera -AspNet-Version header: 4.0.30319*

Para prevenir la vulnerabilidad de la cabecera `X-AspNet-Version: 4.0.30319`, recomendamos seguir las siguientes medidas:

- Desactivar el envío del encabezado `X-AspNet-Version` (y también `X-AspNetMvc-Version` si existe) en la configuración de la aplicación ASP.NET o IIS [126, 94].
- En el archivo `web.config`, dentro de `<system.web>`, establecer la propiedad `enableVersionHeader` de `httpRuntime` a `false` [126, 94].
- Verificar que los parches y actualizaciones de ASP.NET / .NET Framework estén al día, para evitar que existan vulnerabilidades conocidas asociadas a versiones antiguas [126].

8.3.1.9 Mitigación de error de que el nombre del host no coincide con los nombres del certificado (CWE-297: Validación incorrecta de certificado con desajuste de host)

URI	/
HTTP Method	GET
Description	Hostname '192.168.1.2' does not match certificate's names: serverofiauto.
Test Links	https://192.168.1.2:8391/ https://192.168.1.2:8391/
References	https://cwe.mitre.org/data/definitions/297.html

Figura 8.71. *Revelación de error de que el nombre del host no coincide con los nombres del certificado (CWE-297: Validación incorrecta de certificado con desajuste de host)*

URI	/
HTTP Method	GET
Description	Hostname '192.168.1.2' does not match certificate's names: winserver.
Test Links	https://192.168.1.2:8888/ https://192.168.1.2:8888/
References	https://cwe.mitre.org/data/definitions/297.html

Figura 8.72. *Revelación de error de que el nombre del host no coincide con los nombres del certificado (CWE-297: Validación incorrecta de certificado con desajuste de host)*

Para prevenir la vulnerabilidad de que el nombre del host no coincide con los nombres del certificado, recomendamos seguir las siguientes medidas:

- Emitir un certificado TLS/SSL cuyo Common Name (CN) o los Subject Alternative Names (SAN) incluyan explícitamente el nombre o dirección que usarán los usuarios o clientes para acceder al servicio (por ejemplo "192.168.1.2" si lo acceden por IP, o "serverofiauto" si lo acceden por ese nombre) [58, 43].
- No depender solo del CN, usar SANs adecuadamente, ya que CN por sí solo puede no cubrir todos los nombres / IPs usadas [43, 58].
- Si se está accediendo al servidor por dirección IP, considerar emitir un certificado que incluya la IP en los SANs; muchos certificados solo cubren nombres de dominio y no direcciones IP [43].
- Configurar DNS y / o los nombres de host usados por los clientes para que coincidan con los nombres incluidos en el certificado, o bien usar nombres de host en vez de IP si esto facilita tener el certificado correcto [58].

- Verificar después de aplicar los cambios (instalar el certificado nuevo) con herramientas como "SSL Checker", navegadores web o con `openssl s-client` para asegurarnos de que al acceder por el host se presenta un certificado válido cuyo nombre coincide [58, 43].

8.3.1.10 Mitigación de Jetty versión antigua (9.4.16.v20190411)

Figura 8.73. *Revelación de Jetty versión antigua (9.4.16.v20190411)*

Para prevenir la vulnerabilidad de la versión antigua de Jetty (9.4.16.v20190411), recomendamos seguir las siguientes medidas:

- Actualizar Jetty a una versión segura; por ejemplo, Jetty 9.4.54, 10.0.20 o 11.0.20, donde ya se han corregido vulnerabilidades importantes [111, 40].
- Revisar los CVE conocidos para la versión actual y comprobar si alguno tiene exploit publicado o afecta el entorno, para priorizar la actualización [111].
- Aplicar parches oficiales y mantener Jetty actualizado como parte del ciclo de mantenimiento, incluyendo dependencias relacionadas (servlet, webapp, etc.) [111, 40].

8.3.1.11 Mitigación de devolver falsos positivos por medio de método JUNK

Figura 8.74. *Revelación de devolver falsos positivos por medio de método JUNK*

Para prevenir la vulnerabilidad de devolver falsos positivos por medio de método JUNK, recomendamos seguir las siguientes medidas:

- Configurar el servidor o aplicación para rechazar explícitamente métodos HTTP desconocidos o personalizados que no sean necesarios (por ejemplo "FOO", "BAR", "XJWD..."), devolviendo un estado 405 Method Not Allowed [130, 63].
- Revisar la configuración del servidor web / framework para asegurarnos de que solo los métodos necesarios estén habilitados (por ejemplo GET, POST, HEAD, OPTIONS si se requiere CORS), y deshabilitar los métodos inseguros o innecesarios como PUT, DELETE, TRACE, CONNECT, etc. [130, 63].
- Implementar restricción a nivel de seguridad, por ejemplo mediante filtros, reglas en el archivo de configuración (por ejemplo `web.xml` en aplicaciones Java/Servlets) usando "security-constraint" para bloquear métodos específicos [130].

8.3.1.12 Mitigación de Directory Traversal en Cisco ACS

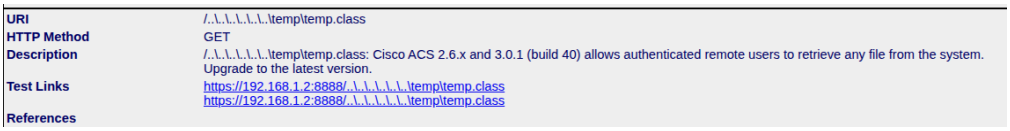


Figura 8.75. *Evidencia de vulnerabilidad de Directory Traversal en Cisco ACS*

Para prevenir la vulnerabilidad que permite a usuarios autenticados recuperar archivos arbitrarios del sistema mediante rutas como **/temp/temp.class**, recomendamos seguir las siguientes medidas:

- Actualizar Cisco ACS a la versión más reciente disponible, que contenga el parche que corrige la vulnerabilidad. Versiones antiguas como 2.6.x y 3.0.1 (build 40) están afectadas [29].
- Validar todas las entradas de las rutas (path inputs) usadas por la aplicación, para impedir secuencias de escape como `..`
  o uso de nombres que permitan escapar del directorio raíz. Filtrar o normalizar los paths antes de acceder al sistema de archivos. [29].
- Configurar adecuadamente los permisos del sistema de archivos de modo que los usuarios autenticados sólo tengan acceso de lectura/escritura a los ficheros estrictamente necesarios, y que los directorios sensibles (fuera del web root) no sean accesibles por la interfaz web. [29].
- Limitar el acceso al módulo o interfaz vulnerable a sólo usuarios de confianza, mediante reglas de firewall, listas de control de acceso (ACL) u otras barreras de red, mientras no apliquemos la actualización. [29].

8.3.1.13 Mitigación de cabecera Access-Control-Allow-Origin con wildcard y posible XSS

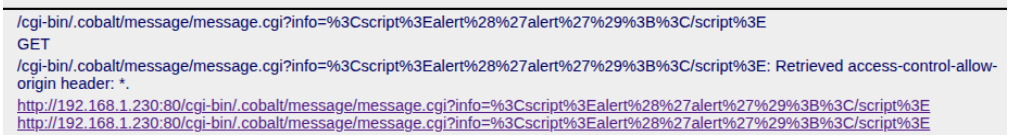


Figura 8.76. *Revelación de cabecera Access-Control-Allow-Origin: * en petición con posible inyección de script*

Para prevenir la vulnerabilidad que permite la cabecera `Access-Control-Allow-Origin: *` junto con inyección de contenido (como en el ejemplo `info=%3Cscript%3Ealert(...)%3C/script%3E`), recomendamos seguir las siguientes medidas:

- Restringir el valor de la cabecera `Access-Control-Allow-Origin` a orígenes específicos de confianza en lugar de usar el comodín `*`. Solo dominios seguros y verificados deberían poder hacer solicitudes cross-origin [101, 18].
- Evitar permitir credenciales (cookies, sesiones, autenticaciones) cuando se usa `Access-Control-Allow-Origin: *`. Si se requieren credenciales, la política CORS no debe usar `*` sino un origen concreto [101, 18].

- Validar y filtrar toda entrada que luego se refleje en la respuesta para evitar inyección de scripts o contenido no deseado. En particular, cuando se permite un parámetro como `info` que luego se envía al cliente, asegurarnos de escaparlo adecuadamente. [18]

8.3.1.14 Mitigación de uso de cabecera `Content-Type: text/plain` con contenido inseguro

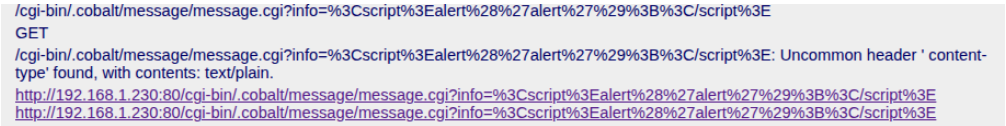


Figura 8.77. *Respuesta con `Content-Type: text/plain` permitiendo contenido de tipo script*

Para prevenir la vulnerabilidad asociada al uso de la cabecera Content-Type: text/plain cuando se devuelve contenido que puede incluir código (por ejemplo un script), recomendamos seguir las siguientes medidas:

- Establecer el Content-Type correcto acorde al contenido devuelto. Si la respuesta incluye HTML o fragmentos de código que el navegador pueda interpretar, usar text/html; charset=UTF-8 en vez de text/plain. [60]
- Añadir la cabecera de respuesta X-Content-Type-Options: nosniff para evitar que navegadores realicen "MIME sniffing" (interpretación automática del contenido), lo que puede permitir ejecución de scripts aunque el Content-Type sea text/plain. . [60]

8.3.1.15 Mitigación de exposición de la utilidad de configuración BIG-IP vía `bigconf.cgi`

Figura 8.78. *Revelación de acceso mediante `bigconf.cgi` en BIG-IP Configuration CGI*

Para prevenir la vulnerabilidad que permite a usuarios acceder a la utilidad de configuración (o archivos relacionados) mediante el script `/cgi-bin/bigconf.cgi`, recomendamos seguir las siguientes medidas:

- Actualizar BIG-IP a la versión más reciente que ya no incluya la vulnerabilidad de `bigconf.cgi` o que tenga parches publicados para ese script [122].
- Restringir el acceso al recurso `bigconf.cgi` de modo que exclusivamente usuarios con los permisos adecuados (por ejemplo administradores) lo puedan ejecutar, mediante autenticación sólida y autorizaciones [122].

- Deshabilitar o eliminar `bigconf.cgi` si no es necesario para la operación normal del sistema; si la funcionalidad que ofrece no es usada, eliminarla reduce la superficie de ataque [122].

8.3.1.16 Mitigación de ejecución arbitraria de comandos mediante `webdist.cgi`

Figura 8.79. *Revelación de vulnerabilidad de ejecución arbitraria de comandos con `webdist.cgi` (CVE-1999-0039)*

Para prevenir la vulnerabilidad que permite a atacantes remotos ejecutar comandos arbitrarios mediante el parámetro `distloc` de `webdist.cgi`, recomendamos seguir las siguientes medidas:

- Eliminar o deshabilitar el script `webdist.cgi` si no es estrictamente necesario para la operación del sistema. [122]
- Restringir el acceso a `webdist.cgi` a redes confiables mediante firewall, reglas de red, listas de control de acceso, etc. No exponerlo públicamente si no es imprescindible. [122]

8.3.1.17 Mitigación de divulgación de archivos arbitrarios mediante `pfdispaly.cgi`

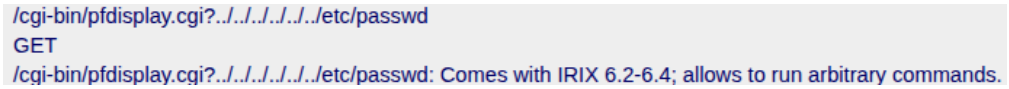


Figura 8.80. *Evidencia de Directory Traversal en `pfdispaly.cgi` (SGI IRIX)*

Para prevenir la vulnerabilidad que permite a usuarios remotos leer cualquier archivo del sistema usando rutas como ../../../../etc/passwd vía pfdispaly.cgi, recomendamos seguir las siguientes medidas:

- Validar la entrada del parámetro que recibe la ruta en el script para eliminar o rechazar secuencias de escape. [29]
- Ajustar los permisos de archivos y directorios sensibles de modo que el usuario que corre el servidor web (o CGI) no tenga acceso de lectura (o permisos innecesarios) sobre los ficheros fuera del directorio web root. [29]
- Restringir el acceso al script pfdispaly.cgi solo a usuarios autenticados y autorizados, o mediante restricciones de red (firewall, listas blancas de IP) si el uso público no es necesario. [29]

8.3.2 Vulnerabilidades encontradas en Impresoras y Escaners

- **Impresora HL-L5100DN, Impresora HL-L5200DW**
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.

- **Impresora Brother HL-L5210-DN, Impresora HL-L5100DN Modelo 2**
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.63, vulnerabilidad de tipo métodos permitidos, en el puerto 80 con servicio http.
- **Impresora HP Laserjet M404DN**
  - No encontramos vulnerabilidades en esta impresora.
- **Impresora Ricoh MP C401SR y Impresora Ricoh MP C401SR Modelo 2**
  - Hemos detectado la vulnerabilidad 8.66, vulnerabilidad de tipo CookieOnOff, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.67, vulnerabilidad de tipo cambio de servidor, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.68, vulnerabilidad de tipo filtración de información, en el puerto 80 con servicio http.
- **Escaner PFU Modelo 1**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.63, vulnerabilidad de tipo métodos permitidos, en el puerto 80 con servicio http.
- **Escaner PFU Modelo 2**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.63, vulnerabilidad de tipo métodos permitidos, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.65, vulnerabilidad de tipo filtración de credenciales, en el puerto 80 con servicio http.

8.3.3 Vulnerabilidades encontradas en las Cámaras

- **Cámara GrandStream Network**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.

8.3.4 Vulnerabilidades encontradas en los dispositivos de red

- **TpLink**
  - No detectamos vulnerabilidades relevantes en el dispositivo.
- **Millenial Net**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.74, vulnerabilidad de tipo falsos positivos, en el puerto 80 con servicio https.
  - Hemos detectado la vulnerabilidad 8.76, vulnerabilidad de tipo CORS, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.77, vulnerabilidad de tipo Content-Type, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.79, vulnerabilidad de tipo ejecución arbitraria de comandos, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.80, vulnerabilidad de tipo Directory Traversal, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.78, vulnerabilidad de tipo divulgación de la utilidad de configuración, en el puerto 80 con servicio http.

8.3.5 Vulnerabilidades encontradas en Ordenadores

- **Hewlett Packward y Hon Hai Precision**
  - No detectamos vulnerabilidades en estos ordenadores.
- **Hewlett Packward Enterprise**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.69, vulnerabilidad de tipo Tls, en el puerto 443 con servicio https.

8.3.6 Vulnerabilidades encontradas en IIS

- **IIS Asustek Compute,Dell, HewlettPackard Modelo 2, HewlettPackard Modelo 3**
  - No detectamos vulnerabilidades en estos ordenadores.
- **IIS Dell Modelo 2, IIS Hewlett Packard, IIS Hewlett Packward Modelo 5**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 80 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80 con servicio http.

- **IIS Hewlett Packard Modelo 4**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 623 y 16992 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 623 y 16992 con servicio http.
  - Hemos detectado la vulnerabilidad 8.63, vulnerabilidad de tipo métodos permitidos, en el puerto 623 y 16992 con servicio http.

8.3.7 Vulnerabilidades encontradas en Servidor y Base de Datos

- **Base de Datos**
  - No detectamos vulnerabilidades.
- **Servidor**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, en el puerto 80, 4042, 4043, 8391,8888, 10447 y 20447 con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, en el puerto 80, 4042, 4043, 8391,8888, 10447 y 20447 con servicio http.
  - Hemos detectado la vulnerabilidad 8.69, vulnerabilidad de tipo Tls, en el puerto 8391,8888 con servicio https.
  - Hemos detectado la vulnerabilidad 8.71 vulnerabilidad de tipo e certificado , en el puerto 8391 y 8888 con servicio http.
  - Hemos detectado la vulnerabilidad 8.63, vulnerabilidad de tipo métodos permitidos, en el puerto 8888 con servicio http.
  - Hemos detectado la vulnerabilidad 8.75, vulnerabilidad de tipo directorio trasversal, en el puerto 8888 con servicio http.
  - Hemos detectado la vulnerabilidad 8.65, vulnerabilidad de tipo filtración de credenciales, en el puerto 8888 con servicio http.

8.3.8 Vulnerabilidades encontradas en la Pagina Web

- **Página web**
  - Hemos detectado la vulnerabilidad 8.40, vulnerabilidad de tipo ClickJacking, con servicio http.
  - Hemos detectado la vulnerabilidad 8.63, vulnerabilidad de tipo métodos permitidos, con servicio http.
  - Hemos detectado la vulnerabilidad 8.62, vulnerabilidad de tipo MIME, con servicio http.

8.4 Wapiti

En esta fase utilizamos la herramienta *Wapiti* para realizar un escaneo de vulnerabilidades web. Wapiti recorre (crawls) las páginas públicas de la aplicación, extrae enlaces y formularios, y actúa como un fuzzer que inyecta payloads para identificar debilidades en los endpoints analizados. [46] Los resultados obtenidos los contrastamos con las imágenes y salidas previas de Nmap/Legion para centrar el análisis en los servicios HTTP detectados (ver figura 8.1, 8.2 y 8.3).

Durante el análisis con Wapiti resaltamos vulnerabilidades y anomalías que no aparecían en los escaneos de las anteriores herramientas, lo que evidencia la complementariedad entre el mapeo de puertos/servicios (Nmap/Legion) y el escaneo dinámico de aplicaciones (Wapiti).

El comando que utilizamos para el análisis del host (formato adoptado en los informes) fue el siguiente:

```
sudo wapiti -u http://192.168.1.2 -o ./reporte_Servidor.html -f html
```

Donde:

- `-u http://192.168.1.2` es la URL/host objetivo (lo tomamos del listado de servicios HTTP detectados por Legion).
- `-o ./reporte_Servidor.html` especifica el fichero de salida en formato HTML.
- `-f html` indica el formato de informe (HTML).

8.4.1 Vulnerabilidades encontradas

8.4.1.1 Mitigación de consumo de recursos sin control

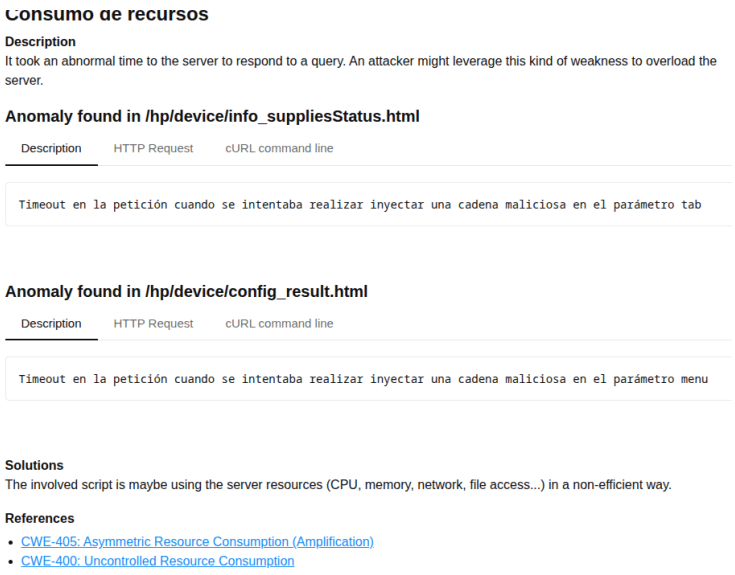


Figura 8.81. *Ausencia de protección contra el consumo de recursos*

En la imagen 8.81 observamos que la aplicación web no implementa mecanismos para limitar el consumo de recursos por petición. Durante las pruebas, al inyectar cadenas maliciosas en parámetros como `tab` o `menu`, la petición HTTP sufrió un **timeout**, lo que indica un uso descontrolado de recursos por parte del servidor.

Para proteger una página web contra la ausencia de protección contra el consumo de recursos, recomendamos las siguientes medidas:

- Imponer límites de tiempo (timeouts) y uso máximo de CPU/memoria por petición. [98]
- Validar y sanear parámetros (longitud máxima, formato esperado). [98]
- Controlar la complejidad de operaciones internas (evitar loops costosos, recursividad excesiva). [98]

8.4.1.2 Mitigación de errores internos (HTTP 500) inducidos por inyección

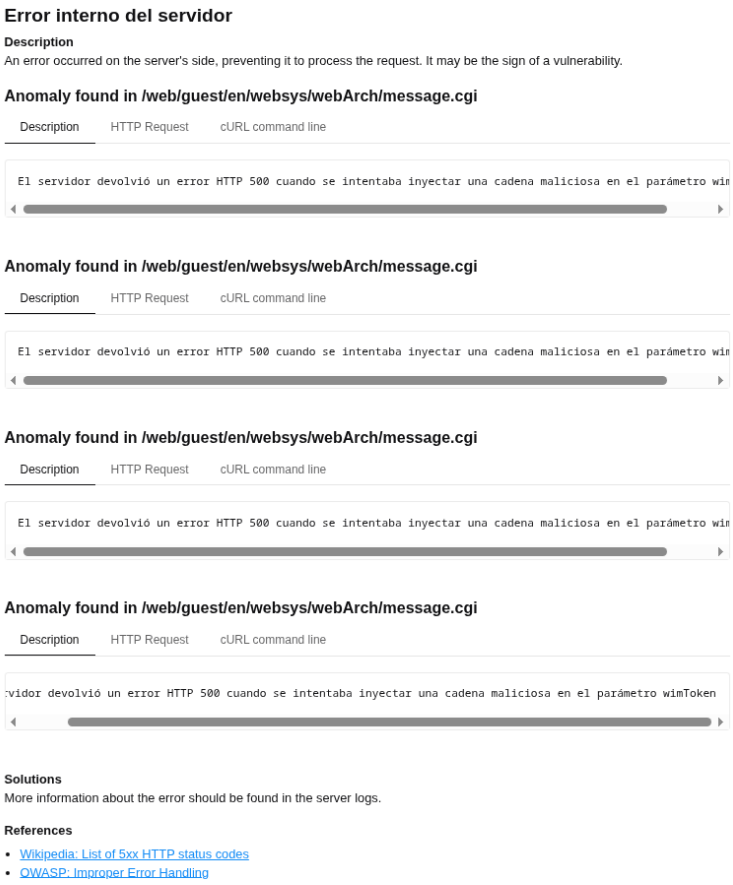


Figura 8.82. *Error interno del servidor ante inyección de datos maliciosos*

En la figura 8.82 observamos que al inyectar cadenas maliciosas en el parámetro `wimToken` del endpoint `/web/guest/en/websys/webAr` el servidor responde con un **HTTP 500**. Esto sugiere que el procesamiento interno falla ante entrada inesperada, revelando un manejo de errores insuficiente.

Para mitigar este tipo de vulnerabilidad, sugerimos las siguientes acciones:

- Validar y sanear el parámetro `wimToken` (longitud, caracteres permitidos, formato) antes de procesarlo internamente. [54]
- Implementar manejo de excepciones seguro: capturar errores internos y retornar respuestas genéricas al cliente sin exponer detalles internos. [54]
- Registrar los errores internamente (logs) para diagnóstico, pero no mostrar trazas al cliente. [54]

8.4.1.3 Mitigación de inyección SQL a ciegas

Figura 8.83. *Detección de inyección SQL a ciegas en* `login.cgi`

En la figura 8.83 mostramos que la aplicación permite una *inyección SQL a ciegas* en el parámetro `userid_work` del endpoint `/web/guest/en/websys/webArch/login.cgi`. Aunque la respuesta no revela datos (no hay error explícito ni resultados visibles), podemos inferir la vulnerabilidad observando cambios en comportamiento o tiempos de respuesta ante condiciones booleanas o retrasos establecidos.

Para evitar este tipo de vulnerabilidad, recomendamos las siguientes medidas:

- Emplear consultas preparadas (parameterized queries) o sentencias parametrizadas en lugar de concatenar directamente cadenas de usuario. [54]
- Validar y filtrar las entradas del usuario (lista blanca, longitud máxima, caracteres permitidos) antes de pasarlas a la capa de datos. [54]
- Limitar el privilegio de acceso a la base de datos (mínimos permisos necesarios) para que una explotación no decante en una toma total del sistema. [15]
- Implementar mecanismos de detección de patrones sospechosos en tiempos de respuesta (delay-based) e introducir límites de tasa (rate limiting) para dificultar la extracción por fuerza bruta. [54]
- Usar firewall de aplicación web (WAF) para filtrar peticiones con payloads SQL comunes y bloquear solicitudes maliciosas antes de llegar al servidor. [15]

8.4.2 Vulnerabilidades encontradas en Impresoras y Escaners

- **Impresora HL-L5100DN, Impresora HL-L5200DW, Impresora Brother HL-L5210-DN, Impresora HL-L5100DN Modelo 2, Escaner PFU Modelo 1 y Escaner PFU Modelo 2**
  - No detectamos vulnerabilidades relevantes en estos dispositivos.
- **Impresora HP Laserjet M404DN**
  - Detectamos la vulnerabilidad de la figura 8.81, de tipo consumo de recursos.
- **Impresora Ricoh MP C401SR y Impresora Ricoh MP C401SR Modelo 2**
  - Detectamos la vulnerabilidad 8.83, de tipo inyección SQL a ciegas, en el puerto 80 con servicio http.
  - Detectamos la vulnerabilidad 8.82, de tipo error interno del servidor, en el puerto 80 con servicio http.

8.4.3 Vulnerabilidades encontradas en las Cámaras

- **Cámara GrandStream Network**
  - No detectamos vulnerabilidades relevantes en este dispositivo.

8.4.4 Vulnerabilidades encontradas en los dispositivos de red

- **TpLink y Millenial Net**
  - No detectamos vulnerabilidades relevantes en estos dispositivos.

8.4.5 Vulnerabilidades encontradas en Ordenadores

- **Hewlett Packward, Hewlett Packward Enterprise y Hon Hai Precision**
  - No detectamos vulnerabilidades relevantes en estos ordenadores.

8.4.6 Vulnerabilidades encontradas en IIS

- **IIS Asustek Compute,Dell, IIS Dell Modelo 2,IIS Hewlett Packard,IIS HewlettPackard Modelo 2, IIS HewlettPackard Modelo 3,IIS Hewlett Packard Modelo 4,IIS Hewlett Packard Modelo 5**
  - No detectamos vulnerabilidades relevantes en estos dispositivos.

8.4.7 Vulnerabilidades encontradas en Servidor y Base de Datos

- **Base de Datos y Servidor**
  - No detectamos vulnerabilidades relevantes en estos dispositivos.

8.4.8 Vulnerabilidades encontradas en Aplicación Web

- **Aplicación Web**
  - No detectamos vulnerabilidades relevantes en este dispositivo.

8.5 WPScan

En este apartado describimos, de forma técnica, las vulnerabilidades detectadas por el escaneo con WPScan. Solo lo detectamos en la página web, en los demás dispositivos no encontramos ningún servicio web basado en WordPress. Para cada entrada incluimos: una breve descripción, el impacto potencial, las medidas de mitigación recomendadas y referencias (CVE / fuentes públicas).

Con WPScan identificamos **15** alertas en los plugins detectados y varias advertencias/informaciones adicionales (XML-RPC habilitado, `readme.html` accesible, WP-Cron externo). Las entradas más críticas afectan principalmente al plugin *Elementor* y al plugin *ocean-extra*. A continuación detallamos las vulnerabilidades individuales.

8.5.0.1 Vulnerabilidades en *Elementor*

8.5.0.2 Elementor $<$ 3.24.0 — Authenticated (Contributor+) Stored XSS en parámetro URL (CVE-2024-5416)

Descripción: Un usuario con rol *contributor* puede inyectar contenido malicioso persistente (Cross-Site Scripting almacenado) a través de un parámetro URL en determinados widgets.

Impacto: ejecución de JavaScript en el navegador de administradores u otros usuarios que visualicen la página; robo de cookies/sesiones, redirecciones maliciosas, defacement.

Mitigación:

- Actualizar Elementor a versión ≥ 3.24.0. [76]
- Revisar y limitar privilegios del rol *contributor* (evitar que editen widgets que renderizan HTML sin filtrado). [76]
- Sanear y escapar todas las entradas que se muestren en vistas públicas o de administración. [76]
- Auditar entradas recientes creadas por usuarios *contributor*. [76]

8.5.0.3 Elementor 3.24.6 — Exposición de información (CVE-2024-6757)

Descripción: La función encargada de obtener el texto alternativo de imágenes podía exponer metadatos, rutas u otros datos sensibles.

Impacto: divulgación de información útil para la enumeración y la planificación de ataques posteriores.

Mitigación:

- Actualizar a versión ≥ 3.24.6. [78]
- Revisar la información mostrada en metadatos de imágenes y limitar los datos expuestos. [78]
- Aplicar control de acceso donde proceda y sanitizar las salidas. [78]

8.5.0.4 Elementor < 3.25.8 — Contributor+ Stored XSS (CVE-2024-8236)

Descripción: Vulnerabilidad de XSS persistente explotable por usuarios con rol *contributor*.

Impacto: ejecución de scripts en contexto de usuarios que visualicen el contenido afectado.

Mitigación:

- Actualizar a versión ≥ 3.25.8. [79]
- Validar y sanear las entradas de usuario en los widgets/configuraciones susceptibles. [79]
- Auditar contenido previamente creado por roles contribuyentes. [79]

8.5.0.5 Elementor < 3.25.10 — Contributor+ Stored XSS vía *Typography Settings* (CVE-2024-10453)

Descripción: En los ajustes tipográficos permitía introducir código malicioso que se almacenaba y luego se ejecutaba al renderizar la página.

Impacto: XSS persistente mediante configuraciones visuales; riesgo para usuarios y administradores.

Mitigación:

- Actualizar a versión ≥ 3.25.10. [73]
- Validar y filtrar los valores recibidos en las configuraciones tipográficas. [73]
- Auditar ajustes tipográficos ya existentes para detectar posibles payloads. [73]

8.5.0.6 Elementor < 3.27.5 — Contributor+ Stored XSS (CVE-2024-13445)

Descripción: Nueva instancia de XSS persistente explotable por roles *contributor* en widgets o parámetros internos.

Impacto: ejecución de scripts cuando usuarios visualicen contenido comprometido.

Mitigación:

- Actualizar a versión ≥ 3.27.5. [74]
- Aplicar saneamiento y escape de todas las entradas. [74]
- Auditar contenido generado por roles de bajo privilegio. [74]

8.5.0.7 Elementor < 3.25.11 — Contributor+ Stored XSS (CVE-2024-54444)

Descripción: Variante adicional de XSS persistente corregida en la versión indicada.

Impacto: ejecución de scripts maliciosos al mostrar contenido afectado.

Mitigación:

- Actualizar a versión ≥ 3.25.11. [77]
- Sanear y escapar todas las entradas de usuario. [77]
- Auditar contenido existente para detectar inyecciones previas. [77]

8.5.0.8 Elementor $<$ 3.29.1 — Contributor+ Stored XSS (CVE-2024-50555 y CVE-2025-3075)

Descripción: Dos vulnerabilidades tipo XSS persistente corregidas en la versión 3.29.1, explotables por usuarios con rol *contributor*. [75, 80]

Impacto: ejecución de scripts cuando se visualiza contenido infectado; riesgos de manipulación de sesión, redirecciones maliciosas, defacement.

Mitigación:

- Actualizar Elementor a versión ≥ 3.29.1. [75, 80]
- Auditar contenido aportado por usuarios *contributor*. [75, 80]
- Garantizar saneamiento y escape adecuados para todas las rutas de entrada. [75, 80]

8.5.0.9 Elementor $<$ 3.30.3 — Contributor+ Stored XSS vía *Text Path Widget* (CVE-2025-4566)

Descripción: El widget *Text Path* permitía la inyección y almacenamiento de payloads XSS, los cuales se ejecutaban al renderizar la página. [84]

Impacto: ejecución de código malicioso en el contexto de los usuarios que visitan páginas afectadas.

Mitigación:

- Actualizar a versión ≥ 3.30.3. [84]
- Restringir el uso del widget *Text Path* si no es imprescindible. [84]
- Validar y sanear todos los parámetros del widget antes de su renderización. [84]
- Auditar páginas que usen este widget para detectar inyecciones anteriores. [84]

8.5.0.10 Elementor $<$ 3.30.3 — Admin+ Arbitrary File Read vía Image Import (CVE-2025-8081)

Descripción: La funcionalidad de importación de imágenes podía ser usada por un actor con privilegios administrativos para leer ficheros arbitrarios del servidor (por ejemplo `.env`, backups o archivos de configuración). [86]

Impacto: divulgación de archivos sensibles del servidor, lo que puede derivar en el compromiso completo del sitio o del servidor.

Mitigación:

- Actualizar a versión ≥ 3.30.3. [86]
- Revisar permisos de ficheros y del usuario que ejecuta el proceso web, limitando lecturas solo a rutas seguras. [86]
- Evitar otorgar privilegios de admin a cuentas no necesarias. [86]
- Auditar logs de importación de imágenes y accesos a rutas sensibles. [86]

8.5.1 Vulnerabilidades en *ocean-extra*

8.5.1.1 Ocean Extra < 2.4.7 — Contributor+ Stored XSS vía Shortcode (CVE-2025-3457)

Descripción: Shortcodes gestionados por el plugin permitían que un usuario con rol *contributor* guardara contenido malicioso que luego se interpretaba como script al visualizar la página. [81]

Impacto: XSS persistente al renderizar páginas que usan esos shortcodes; riesgo de robo de sesión, relevancia para defacement u otras manipulaciones. [81]

Mitigación:

- Actualizar Ocean Extra a versión ≥ 2.4.7. [81]
- Deshabilitar shortcodes inseguros o no utilizados. [81]
- Validar y sanear todos los parámetros aceptados por los shortcodes. [81]
- Auditar contenido existente que use esos shortcodes para detectar inyecciones previas. [81]

8.5.1.2 Ocean Extra < 2.4.7 — Contributor+ Stored XSS vía `ocean_gallery_id` (CVE-2025-3458)

Descripción: El parámetro `ocean_gallery_id` no validaba ni escapaba correctamente su contenido, permitiendo la inyección persistente de código malicioso. [82]

Impacto: ejecución de scripts al renderizar la página que use ese parámetro. [82]

Mitigación:

- Actualizar a versión ≥ 2.4.7. [82]
- Sanitizar los valores recibidos para `ocean_gallery_id`. [82]
- Revisar contenido existente que use dicho parámetro para detectar inyecciones. [82]

8.5.1.3 Ocean Extra < 2.4.7 — Ejecución arbitraria de shortcodes sin autenticación (CVE-2025-3472)

Descripción: Permite la ejecución de shortcodes arbitrarios sin autenticación previa: un atacante remoto podría invocar funciones del plugin usando parámetros públicos que no se validan adecuadamente. [83]

Impacto: ejecución no autorizada de funcionalidades del plugin, con posibilidad de divulgación, modificación o ejecución de operaciones no previstas. [83]

Mitigación:

- Actualizar a versión ≥ 2.4.7. [83]
- Si no hay parche disponible, desactivar Ocean Extra hasta aplicar la corrección. [83]
- Revisar todas las páginas que usen shortcodes y eliminar o restringir los que no sean imprescindibles. [83]
- Implementar controles de validación antes de ejecutar do-shortcode o funciones similares. [83]

8.5.1.4 Ocean Extra < 2.4.9 — Authenticated (Contributor+) Stored XSS (CVE-2025-49068)

Descripción: Variante de XSS persistente explotable por usuarios con rol *contributor*, corregida en la versión 2.4.9. [85]

Impacto: ejecución de scripts maliciosos al visualizar contenido afectado. [85]

Mitigación:

- Actualizar a versión ≥ 2.4.9. [85]
- Validar y sanear todas las entradas de usuario asociadas. [85]
- Auditar contenido existente para detectar inyecciones. [85]

8.5.1.5 Ocean Extra < 2.5.0 — Contributor+ Stored XSS (CVE-2025-9499)

Descripción: Vulnerabilidad de XSS persistente presente en versiones anteriores a 2.5.0, explotable por usuarios con rol *contributor*. [87]

Impacto: ejecución de scripts maliciosos al mostrar contenido afectado, con riesgos de robo de sesión, manipulación del DOM y otros efectos. [87]

Mitigación:

- Actualizar a versión ≥ 2.5.0. [87]
- Auditar las entradas o shortcodes usados por usuarios para detectar inyecciones. [87]
- Validar y sanear todos los parámetros del plugin que puedan admitir inyección. [87]

8.5.2 Advertencias e información complementaria

8.5.2.1 Plugins desactualizados

Descripción: Los plugins *elementor* (versión detectada 3.23.3) y *ocean-extra* (versión detectada 2.3.0) están desactualizados respecto a las versiones corregidas indicadas en los avisos.

Riesgo: mantener versiones antiguas incrementa la probabilidad de explotación por vulnerabilidades ya conocidas.

Mitigación: plan de actualización coordinado (staging → producción), comprobación de compatibilidades, y backups previos.

8.5.2.2 XML-RPC habilitado (`/xmlrpc.php`)

Descripción: El endpoint `/xmlrpc.php` está accesible. XML-RPC provee funcionalidades (p. ej. trackbacks, publicación remota) pero también vectores de ataque (pingback DDoS, intentos de fuerza bruta a través de métodos remotos).

Mitigación: si no es necesario, deshabilitar XML-RPC; en caso contrario, aplicar restricciones por IP o WAF, y monitorizar patrones de uso anómalos.

8.5.2.3 `readme.html` accesible

Descripción: El fichero `readme.html` de WordPress está accesible públicamente, lo que puede revelar la versión de WordPress y facilitar reconocimiento.

Mitigación: eliminar o restringir el acceso al fichero (regla en servidor o en .htaccess), y evitar exponer información de versiones.

8.5.2.4 WP-Cron externo activo (`/wp-cron.php`)

Descripción: La ejecución de WP-Cron vía invocaciones externas puede ser mal aprovechada para generar carga o DoS.
Mitigación: configurar un cron real en el servidor y desactivar las invocaciones externas a **wp-cron.php** y programar tareas con **crontab**.

8.5.3 Recomendaciones operativas y orden de prioridad

1. **Actualizar inmediatamente** los plugins afectados elementor y ocean-extra a las versiones mínimas indicadas o a la última versión estable disponible. Realizar la actualización primero en **staging** y verificar compatibilidad antes de aplicar en producción.

2. **Revisar roles y permisos**: especialmente el rol **contributor** (restringir capacidad de añadir shortcodes, widgets o contenido que pueda ser interpretado como HTML).

3. **Desactivar XML-RPC** si no es requerido; proteger `wp-cron.php` y eliminar `readme.html`.

4. **Auditoría y monitoreo**: revisar logs de acceso, uploads e importaciones de imágenes; buscar indicadores de explotación (scripts añadidos, usuarios desconocidos, cambios en shortcodes).

5. **Hardening del servidor**: revisar permisos de ficheros (evitar acceso de lectura a ficheros sensibles por parte del proceso web), aplicar WAF/IDS y reglas para filtrar payloads XSS comunes.

6. **Backup y plan de recuperación**: antes de actualizar, realizar backup completo (ficheros + base de datos) y definir plan para rollback.

9

Conclusiones y Líneas Futuras

9.0.1 Conclusiones

Este Trabajo Fin de Grado ha demostrado, mediante una metodología de auditoría sistemática y reproducible, nuestra capacidad para identificar y priorizar vulnerabilidades reales en el entorno analizado. La combinación de herramientas especializadas —entre ellas Nmap, OWASP ZAP, Nikto, Wapiti, WPScan y sqlmap— junto con nuestras verificaciones manuales y contrastes cruzados, nos permitió descubrir debilidades en diversos activos (aplicaciones web, dispositivos de red y servicios expuestos), clasificarlas por criticidad y asociarlas a causas raíz técnicas y organizativas.

A partir de estos hallazgos, propusimos un plan de remediación que incluye ajustes de configuración, endurecimiento de cabeceras, cierres de puertos y actualización de componentes, junto con acciones estructurales (ciclo de gestión de parches, hardening por defecto, segmentación de red, autenticación robusta y monitorización continua). La ejecución de estas medidas no solo reduce la superficie de ataque y el riesgo operativo, sino que sienta las bases de una mejora continua alineada con las buenas prácticas de seguridad y gobierno del dato.

Finalmente, todos los resultados se reflejan y documentan en el apéndice B siguiente, donde incluimos las evidencias de las pruebas, el detalle por activo y vulnerabilidad, así como las recomendaciones específicas de mitigación y su priorización temporal.

9.0.2 Líneas futuras

Como líneas futuras de trabajo, consideramos diversas acciones que permitirían ampliar y consolidar los resultados obtenidos en esta auditoría. En primer lugar, podríamos incorporar herramientas adicionales de análisis automatizado y correlación de eventos para mejorar la detección de vulnerabilidades emergentes. Asimismo, sería conveniente integrar un sistema de gestión de incidencias vinculado con los hallazgos, de manera que se facilite el seguimiento del ciclo completo de corrección y validación.

Otra línea de mejora consistiría en la implantación de auditorías periódicas, con un enfoque continuo que permita evaluar

la eficacia de las medidas de seguridad aplicadas y detectar desviaciones a tiempo. Finalmente, contemplamos la posibilidad de ampliar el alcance del análisis a infraestructuras en la nube y entornos híbridos, reforzando así la cobertura y resiliencia del sistema frente a amenazas futuras.

Apéndice A. Formulario de autorización

FORMULARIO DE AUTORIZACIÓN DE PENTESTING

Índice

1. Formulario de autorización de Pentesting

Cliente: OFIAUTO
Nombre:José Francisco Roldán
Puesto: Director General
Fecha: 29 de mayo de 2025

Autoriza a José Sánchez-Rosso Almoguera para llevar a cabo las actividades de verificación de seguridad de la aplicación(es) y sistema (s) que se describen) a continuación, según las siguientes condiciones:

- Ámbito de las pruebas (Activos autorizados):
    - https://gestoriaofiauto.es/
    - Infraestructura de red interna, incluyendo routers, switches y puntos de acceso inalámbrico.
    - Equipos y estaciones de trabajo (ordenadores) empleados por el personal de la empresa, incluyendo tanto equipos de escritorio como portátiles integrados en la red corporativa.
    - Aplicaciones web y servicios internos asociados a la operativa de la organización.
    - Sistemas de autenticación y control de acceso (LDAP, Active Directory, etc.).

- Condiciones:
    - Las pruebas serán internas y se realizarán desde dentro de la red del cliente utilizando Internet y para acceder a los activos incluidos en el alcance.
    - Se realizarán varios TEST de penetración:
        - Caja negra
        - Caja blanca

- Planificación temporal:
    - Fechas: Del 29 de mayo al 6 de junio del 2025. Días laborables comprendidos entre el L-V.
    - Horario: Preferiblemente de 16.00h a 20.00h

- Teléfonos de soporte en caso de caída de servicio:
        - Por parte del equipo de producto o cliente:
            - Nombre: Jose Francisco Roldán
            - Teléfono: 666545545
            - Email: juan@gestoriaofiauto.es

        - **PENTESTINGS SÁNCHEZ-ROSSO ALMOGUERA**
            - Nombre: José Sánchez-Rosso Almoguera
            - Teléfono: 640359248
            - Email: jalmoguera2003@gmail.com

2. Tipos de pruebas(Pentesting)

El servicio de tests de intrusión nos permite comprobar cómo de segura es la infraestructura IT para medir tanto su seguridad a nivel técnico como si se satisfacen los niveles regidos por la política organizativa. Estos tests son realizados por expertos en seguridad sobre todos los servicios expuestos a internet y aquellos que puedan accederse remotamente.
Se pueden realizar dos tipos distintos de Tests de Intrusión, Externo o Interno en función desde dónde se lanzan estos ataques.

- El **test de intrusión externo** es una simulación controlada de un ataque informático realizada desde fuera de la red corporativa (por ejemplo, desde Internet). Su objetivo es identificar vulnerabilidades expuestas al público.
    - **Objetivo: Simular el rol de un atacante externo que no tiene acceso previo a la red interna.**

- El **test de intrusión interno** es una simulación controlada de un ataque realizada desde dentro de la red interna de la organización. Se realiza bajo el supuesto de que un atacante ha conseguido acceso a la red interna (por ejemplo, mediante phishing o un dispositivo comprometido).
    - **Objetivo: Simular el comportamiento de un empleado malintencionado o de un atacante que logró acceso a la red interna.**

En el ámbito de los test de intrusión, existen tres enfoques clásicos conocidos como pruebas de caja negra, caja blanca y caja gris. Estos enfoques se diferencian según el nivel de conocimiento que el evaluador tiene sobre el sistema antes de comenzar las pruebas a las que se somete, se definen por las siguientes características:

- **Caja Negra (Black-box):**
    - El pentester no tiene conocimiento previo del sistema.
    - Simula el comportamiento de un atacante externo.
    - Es el tipo más cercano a un ataque real.
    - Requiere mayor tiempo de reconocimiento.
    - Se evalúa la seguridad externa y resistencia de la infraestructura frente a atacantes desconocidos.
- **Caja Blanca (White-box):**
    - El evaluador tiene acceso completo a la información del sistema: código fuente, configuraciones y arquitectura.
    - Permite un análisis exhaustivo y profundo, con ello identificar problemas de seguridad o hardware
    - Es eficaz para comprobar la calidad del código detectando así fallos lógicos
    - Detecta vulnerabilidades internas, midiendo la efectividad de las medidas de seguridad implementada de forma interna
- **Caja Gris (Grey-box):**
    - Combinación de elementos de la caja blanca y negra.
    - Se dispone de información parcial, como credenciales limitadas o diagramas de red.
    - Simula un atacante con acceso restringido,con algún tipo de acceso o información privilegiada, cómo un empleado con permisos básicos,
    - Ofrece un equilibrio entre realismo y profundidad técnica, es decir, se combina la perspectiva interna y externa de manera equilibrada habilitando una evaluación más completa de las vulnerabilidades.

3. Herramientas Autorizadas (Pentesting)

A continuación, se detallan las herramientas empleadas durante la auditoría de seguridad, organizadas por fases, junto con su propósito, alcance y posibles impacto a modo de gráfica:

- **Fase 1: Recogida de Información**
  - **Lynis:** Herramienta de auditoría de seguridad para sistemas Unix/Linux. Realiza un análisis exhaustivo del sistema, identificando posibles vulnerabilidades y configuraciones inseguras. Impacto mínimo en el rendimiento del sistema.
- **Fase 2: Análisis de Vulnerabilidades**
  - **Legion:** Marco de pruebas de penetración de red semi-automatizado y altamente extensible. Permite la detección, reconocimiento y explotación de sistemas de información, utilizando más de 100 scripts auto-programados.
- **Fase 3: Auditoría de Redes Inalámbricas**
  - **Airodump-ng:** Herramienta para la captura de paquetes en redes Wi-Fi, útil para recolectar IVs de WEP y handshakes de WPA/WPA2. Requiere modo monitor en la interfaz inalámbrica.
  - **Aireplay-ng:** Utilidad que permite la inyección de paquetes en redes inalámbricas. Se emplea para generar tráfico artificial o forzar la desconexión de clientes legítimos (ataque de desautenticación), facilitando la captura de handshakes o incrementando el tráfico para obtener más IVs.
  - **Aircrack-ng:** Programa destinado al descifrado de contraseñas Wi-Fi. Utiliza los IVs capturados para romper claves WEP mediante ataques estadísticos y descifra handshakes WPA/WPA2 mediante ataques de diccionario o fuerza bru
- **Fase 4: Ataques a Contraseñas**
  - **Mimikatz:** Herramienta para sistemas Windows que permite extraer contraseñas, hashes y tickets Kerberos directamente de la memoria. Su uso está restringido a entornos controlados y con las debidas autorizaciones.
- **Fase 5: Auditoría de Aplicaciones Web**
  - **OWASP ZAP:** Herramienta de código abierto para pruebas de seguridad en aplicaciones web. Ayuda a identificar vulnerabilidades comunes como inyecciones SQL y XSS.
  - **Nikto:** Herramienta que realiza pruebas exhaustivas contra servidores web, escaneando múltiples vulnerabilidades y configuraciones incorrectas.
  - **SQLMap:** Herramienta de pruebas de penetración que automatiza la detección y explotación de vulnerabilidades de inyección SQL en aplicaciones web.
  - **Wapiti:** Herramienta ligera pero potente para escanear aplicaciones web en busca de vulnerabilidades como inyecciones SQL, XSS y divulgación de archivos.
  - **WPScan:** Herramienta especializada en la detección de vulnerabilidades en sitios web que utilizan WordPress.

Herramienta	Fase	Riesgo e Impacto
Lynis	Recogida de Información	Impacto mínimo en el sistema. Análisis pasivo de configuraciones y vulnerabilidades.
Legion	Análisis de Vulnerabilidades	Posible carga sobre el sistema si se ejecutan múltiples scripts simultáneamente. Detecta y explota vulnerabilidades.
Airodump-ng	Redes Inalámbricas	Captura pasiva de tráfico Wi-Fi. Requiere modo monitor. Bajo impacto directo.
Aireplay-ng	Redes Inalámbricas	Puede desconectar usuarios legítimos (DoS temporal). Riesgo moderado si se abusa.
Aircrack-ng	Redes Inalámbricas	Ataques estadísticos o de diccionario sobre claves Wi-Fi. Impacto nulo si se usa sobre tráfico capturado.
Mimikatz	Ataques a Contraseñas	Extrae credenciales de la memoria. Riesgo crítico si se ejecuta fuera de entorno controlado.
OWASP ZAP	Aplicaciones Web	Escaneo automatizado de vulnerabilidades. Bajo riesgo si se configura correctamente.
Nikto	Aplicaciones Web	Escaneo intensivo de servidores web. Puede generar mucho tráfico. Riesgo bajo si se limita el alcance.
SQLMap	Aplicaciones Web	Automatiza inyecciones SQL. Riesgo moderado si accede a datos reales.
Wapiti	Aplicaciones Web	Escaneo ligero de vulnerabilidades web. Impacto bajo.

Dada la naturaleza de las tareas a ejecutar, existe la posibilidad de que, de manera no intencionada, se produzcan efectos colaterales no deseados. En tal caso, el auditor de Pentestings Sánchez-Rosso Almoguera informará de inmediato al teléfono de soporte correspondiente o se pondrá a disposición del cliente para proporcionar la información requerida sobre las acciones realizadas.

4. Restricciones y conformidades

Se harán vigentes desde el 29 de mayo de 2025 al 6 de junio de 2025, ambos inclusive.
En caso de conformidad con la concesión de esta autorización el Cliente declara lo siguiente:

- El cliente es dueño de los sistemas donde se realiza la auditoría de vulnerabilidades y el suscrito tiene la autoridad adecuada para poder llevar a cabo las actividades de verificación de seguridad de aplicaciones.

- El cliente ha creado una copia de seguridad completa de todos los sistemas dentro del ámbito de las pruebas de vulnerabilidades, y se ha comprobado que el procedimiento de copia de seguridad permitirá al cliente restaurar los sistemas a su estado pre-test.

- El servicio implica necesariamente el uso de herramientas y técnicas diseñadas para detectar las vulnerabilidades de seguridad, y que es imposible identificar y eliminar todos los riesgos que implica el uso de estas herramientas y técnicas.

- El cliente se compromete a no modificar el sistema o aplicativo durante la ejecución del pentesting. Si esto sucediera se entenderá como un cambio de alcance, de tal modo que el pentesting se dará por finalizado y se entregará el informe al cliente con los resultados obtenidos hasta ese momento, con lo que el Auditor cumplirá totalmente con el alcance inicialmente acordado. Si el cliente desea realizar un pentesting al sistema o aplicativo que ha modificado deberá volver a negociarlo con José Sánchez-Rosso Almoguera considerándose en todo caso como el encargo de un nuevo trabajo.

- Reconoce los posibles efectos colaterales de las pruebas y acepta sus riesgos.

Cualquier cambio a las limitaciones y condiciones descritas deberán realizarse por escrito y ser aceptado por ambas partes.

5. Acuerdo de confidencialidad

Se considerará Información Confidencial toda aquella información, ya sea oral, escrita, gráfica o en cualquier otro formato, que la Parte Reveladora proporcione a la Parte Receptora y que esté relacionada con la auditoría de seguridad, incluyendo, pero no limitándose a, datos técnicos, configuraciones de sistemas, credenciales de acceso, informes de vulnerabilidades, y cualquier otra información que, por su naturaleza, deba ser tratada como confidencial.

Obligaciones del Auditor

El auditor se compromete a:

- Utilizar la información confidencial exclusivamente para los fines establecidos en el alcance de la auditoría.
- No divulgar, copiar ni reproducir dicha información sin el consentimiento expreso y por escrito del cliente.
- Implementar medidas de seguridad adecuadas para proteger la información confidencial contra accesos no autorizados.
- Destruir o devolver toda la información confidencial al finalizar la auditoría, según lo acordado con el cliente.

Exclusiones

Las obligaciones de confidencialidad no se aplicarán a la información que:

- Sea de dominio público al momento de su divulgación o que llegue a serlo sin incumplimiento del acuerdo.
- Ha sido obtenida legalmente de terceros sin restricciones de confidencialidad.
- Deba ser divulgada por mandato legal o judicial, en cuyo caso se notificará previamente al cliente, salvo prohibición legal, cuya comunicación o uso sin restricciones haya sido aprobada por la empresa cliente.

Derechos de Propiedad Intelectual

Este acuerdo no implica la concesión de derechos de propiedad intelectual sobre la información confidencial. Todos los derechos, títulos e intereses sobre dicha información permanecerán con el emisor. El receptor no adquirirá ningún derecho de propiedad intelectual por el acceso a la información confidencial.

Duración del Acuerdo

La obligación de confidencialidad permanecerá en vigor durante un **período de tres (3) años** a partir de la finalización de la auditoría.

Incumplimiento

En caso de incumplimiento de las obligaciones establecidas en el presente acuerdo, la Parte Receptora será responsable de los daños y perjuicios que dicho incumplimiento pudiera ocasionar a la Parte Reveladora.

Legislación Aplicable

El presente acuerdo se regirá e interpretará de acuerdo con las leyes de España. Para la resolución de cualquier controversia derivada del mismo, las Partes se someten a la jurisdicción de los tribunales de Málaga.

POR EL CLIENTE	**PENTESTINGS SÁNCHEZ-ROSSO ALMOGUERA**
Fdo:Jose Francisco Roldán Fecha: 29 de mayo de 2025	Fdo:Jose Sánchez-Rosso Almoguera Fecha: 29 de mayo de 2025

Apéndice B. Informe de Auditoría

AUDITORÍA DE PENTESTING

Índice

1.Introducción

Este informe documenta la auditoría de seguridad realizada sobre la infraestructura y los servicios tecnológicos de la organización, con el objetivo de identificar vulnerabilidades, estimar su impacto y proponer medidas de mitigación priorizadas. La motivación principal es reducir la superficie de ataque, elevar la resiliencia ante incidentes y alinear los controles técnicos con las buenas prácticas del sector.

El trabajo se ha desarrollado bajo un enfoque metodológico sistemático: reconocimiento y enumeración de activos y superficies expuestas; análisis de configuración y políticas de seguridad; escaneo pasivo y activo para descubrir debilidades; verificación y correlación de hallazgos; y evaluación de riesgos para priorizar acciones. Este ciclo se ha aplicado tanto a componentes de aplicación web como a equipos y servicios de red (impresoras, cámaras IP, servidores, servicios web/IIS y otros dispositivos relevantes).

En cuanto a herramientas, se han empleado utilidades consolidadas en la industria para cada fase del proceso. Para la capa web, se han utilizado analizadores de seguridad orientados a descubrir fallos de configuración, carencias de cabeceras de protección y posibles vulnerabilidades de aplicación. Para la capa red/sistemas, se han ejecutado escaneos de puertos y servicios, detección de versiones, y verificación básica de configuraciones débiles. Cuando ha sido necesario, se han realizado pruebas no intrusivas adicionales para confirmar la explotabilidad de ciertos hallazgos sin comprometer la disponibilidad de los sistemas.

Los criterios de severidad y priorización se han establecido combinando el impacto potencial, la probabilidad de explotación y el contexto operativo de la organización. Se referencia el uso de estándares de facto (p. ej., CVSS para puntuar severidad) y de guías de buenas prácticas para mitigaciones (cabeceras HTTP de seguridad, políticas de endurecimiento en servidores, segmentación de red, control de accesos, gestión de parches y versiones, etc.).

El alcance ha comprendido los activos y entornos expresamente autorizados por la organización, respetando las restricciones de tiempo, ventana operativa y no intrusión destructiva. No se han realizado pruebas fuera de dicho alcance ni actividades que pudieran degradar los servicios en producción. Cualquier limitación detectada (por ejemplo, falta de visibilidad en determinados segmentos o restricciones de credenciales) se indica en el cuerpo del informe para contextualizar los resultados.

2.Dispositivos Analizados clasificados por Tipo e IP

Se seccionó de la siguiente manera, los dispositivos:

- Impresoras y Escáneres
  - **Brother HL-L5100DN**->192.168.1.22
  - **Brother HL-L5210-DN**->192.168.1.30
  - **Brother HL-L5200DW**->192.168.1.32
  - **Brother HL-L5100DN (Modelo 2)**->192.168.1.37
  - **Ricoh MP C401SR (Modelo 2)**->192.168.1.101
  - **Ricoh MP C401SR**->192.168.1.199
  - **Escáner PFU**->192.168.1.198
  - **Escáner PFU (Modelo 2)**->192.168.1.226
  - **HPLaserJet M404DN**->192.169.1.210

- Cámaras
  - **Cámara GrandStream Network**->192.168.1.220

- Dispositivos de red
  - **TpLink-**>192.168.1.207
  - **Millenial Net**->192.168.1.230

- Ordenadores
  - **Hewlett Packard**->192.168.1.12
  - **Hewlett Packard iLO**->192.168.1.13
  - **HonHai Precision**->192.168.1.202

- Servicios IIS
  - **IIS Hewlett Packard** ->192.168.1.99
  - **IIS Asustek Computer**->192.168.1.205
  - **IIS Dell**->192.168.1.207
  - **IIS Hewlett Packard (Modelo 2)**->192.168.1.225
  - **IIS Dell (Modelo 2)**->192.168.1.219 228
  - **IIS Hewlett Packard (Modelo 3)**->192.168.1.239
  - **IIS Hewlett Packard (Modelo 4)**-> 192.168.1.240
  - **IIS Hewlett Packard (Modelo 5**)-> 192.168.1.245

- Servidores y Base de Datos
  - **Servidor Hewlett Packard**->192.168.1.2
  - **Microsoft SQL Server (Base de Datos)**->192.168.1.10

3. Vulnerabilidades encontradas

A continuación, se detallan todas las vulnerabilidades identificadas durante la auditoría, junto con las acciones específicas que deben llevarse a cabo para mitigarlas. Se propone este conjunto de medidas prácticas y priorizadas con el objetivo de reducir la superficie de ataque, reforzar la seguridad de los servicios expuestos, y minimizar el riesgo de explotación exitosa.

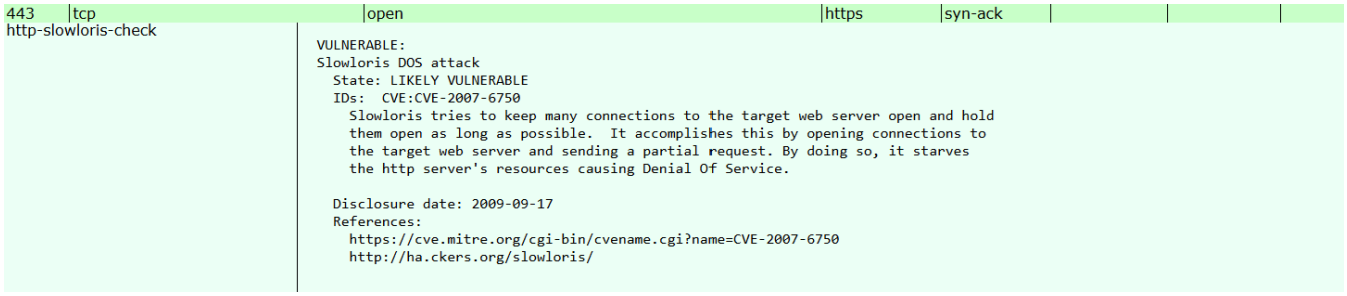
- Las vulnerabilidades fueron **detectadas** empleando diferentes herramientas de auditoría, escáneres y pruebas manuales bajo distintos vectore.

- Para cada vulnerabilidad se sugiere una **medida de mitigación concreta**, debidamente priorizada según su riesgo, para que se pueda aplicar en el entorno real de la empresa.

- Estas medidas no son solo correctivas sino también preventivas: buscan endurecer configuraciones, eliminar puntos débiles de exposición o reforzar la validación y el control (cabeceras HTTP, atributos de cookies, mecanismos de autenticación, control de entrada de datos, etc.).

- Es importante que tras aplicar las mitigaciones, se vuelvan a realizar pruebas (re-testing) con las mismas herramientas para verificar que la vulnerabilidad haya sido efectivamente mitigada sin introducir regresiones.

Mitigación de Slowloris



Para proteger un servidor web contra ataques Slowloris, se recomiendan las siguientes medidas:

- **Apache:** activar el módulo mod_reqtimeout (disponible a partir de la versión 2.2.15) para imponer tiempos límite mínimos a la recepción de cabeceras y cuerpo de peticiones HTTP, evitando que conexiones parciales se mantengan indefinidamente [Using mod_qos and mod_reqtimeout to mitigate Slowloris attacks | Liquid Web].

- **NGINX servidores bloqueantes**:aplicar límites de conexión(limit_conn_zone) y de tasa (limit_req_zone); configurar tiempos de espera de cabecera, cuerpo y keep-alive; funcionan mejor debido a su arquitectura asíncrona [Mitigating DDoS Attacks with NGINX – NGINX Community Blog].

- **Balanceadores de carga / WAF / proxies**: utilizar dispositivos o servicios que sólo reenvían peticiones HTTP completas o redirigen clientes tras validar cabeceras; así el servidor backend queda protegido [IBM Documentation].

- **Reglas de firewall / iptables**: limitar el número de conexiones concurrentes por IP al puerto 80 (por ejemplo, con-m connlimit), reduciendo el riesgo de ataques Slowloris distribuidos [Using mod_qos and mod_reqtimeout to mitigate Slowloris attacks | Liquid Web].

Mitigación de vulnerabilidades en Genivia gSOAP (CVE-2017-9765, CVE-2019-7659, CVE-2020-1357, CVE-2021-21783)

```
vulners
    cpe:/a:genivia:gsoap:2.7:
          CVE-2021-21783  9.8     https://vulners.com/cve/CVE-2021-21783
          CVE-2020-13576  9.8     https://vulners.com/cve/CVE-2020-13576
          CVE-2019-7659   8.1     https://vulners.com/cve/CVE-2019-7659
          CVE-2017-9765   8.1     https://vulners.com/cve/CVE-2017-9765
          CVE-2020-13578  7.5     https://vulners.com/cve/CVE-2020-13578
          CVE-2020-13577  7.5     https://vulners.com/cve/CVE-2020-13577
          CVE-2020-13575  7.5     https://vulners.com/cve/CVE-2020-13575
          CVE-2020-13574  7.5     https://vulners.com/cve/CVE-2020-13574
          SSV:96284       6.8     https://vulners.com/seebug/SSV:96284    *EXPLOIT*
```

Para mitigar las vulnerabilidades críticas en Genivia gSOAP, se recomienda seguir estas me didas:

- **Actualizar a la versión más reciente de gSOAP (≥ 2.8.111) :**Todas las vulnerabilidades críticas relacionadas con WS-Addressing y WS-Security, incluyendo CVE-2020 13576, CVE-2021-21783, CVE-2020-13574–78 y CVE-2019-7659, están resueltas en esa versión o posterior [Vulnerability Spotlight: Multiple vulnerabilities in Genivia gSOAP].

- **Recompilar sin soporte de cookies (parámetro-DWITH_COOKIES):** CVE-2019 7659 se evita eliminando el soporte de cookies en el servidor SOAP [NVD - CVE-2019-7659].

- **Desactivar plugins nonecesarios (WS-Addressing / WS-Security)**: Las vulnerabilidades más severas se localizan en estos plugins; si no son imprescindibles, es recomendable deshabilitarlos durante la generación del servicio [Vulnerability Spotlight: Multiple vulnerabilities in Genivia gSOAP]

Mitigación de vulnerabilidades en OpenSSH 8.0 (CVE-2023-38408 y vulnerabilidades asociadas)

vulners

```
cpe:/a:openbsd:openssh:8.0:
      F0979183-AE88-53B4-86CF-3AF0523F3807    9.8     https://vulners.com/githubexploit/F0979183-AE88-53B4-86CF-3AF0523F3807  *EXPLOIT*
      CVE-2023-38408  9.8     https://vulners.com/cve/CVE-2023-38408
      B8190CDB-3EB9-5631-9828-8064A1575B23    9.8     https://vulners.com/githubexploit/B8190CDB-3EB9-5631-9828-8064A1575B23  *EXPLOIT*
      8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8     https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-825E-E8DB5379A623  *EXPLOIT*
      8AD01159-548E-546E-AA87-2DE89F3927EC    9.8     https://vulners.com/githubexploit/8AD01159-548E-546E-AA87-2DE89F3927EC  *EXPLOIT*
      5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A    9.8     https://vulners.com/githubexploit/5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A  *EXPLOIT*
      2227729D-6700-5C8F-8930-1EEAFD4B9FF0    9.8     https://vulners.com/githubexploit/2227729D-6700-5C8F-8930-1EEAFD4B9FF0  *EXPLOIT*
      0221525F-07F5-5790-912D-F4B9E2D1B587    9.8     https://vulners.com/githubexploit/0221525F-07F5-5790-912D-F4B9E2D1B587  *EXPLOIT*
      CVE-2020-15778  7.8     https://vulners.com/cve/CVE-2020-15778
      CVE-2019-16905  7.8     https://vulners.com/cve/CVE-2019-16905
      C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3    7.8     https://vulners.com/githubexploit/C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3  *EXPLOIT*
      10213DBE-F683-58BB-B6D3-353173626207    7.8     https://vulners.com/githubexploit/10213DBE-F683-58BB-B6D3-353173626207  *EXPLOIT*
      SSV:92579       7.5     https://vulners.com/seebug/SSV:92579    *EXPLOIT*
      PACKETSTORM:173661      7.5     https://vulners.com/packetstorm/PACKETSTORM:173661      *EXPLOIT*
      1337DAY-ID-26576        7.5     https://vulners.com/zdt/1337DAY-ID-26576        *EXPLOIT*
      CVE-2021-41617  7.0     https://vulners.com/cve/CVE-2021-41617
      PACKETSTORM:189283      6.8     https://vulners.com/packetstorm/PACKETSTORM:189283      *EXPLOIT*
      F79E574D-30C8-5C52-A801-66FFA0610BAA    6.8     https://vulners.com/githubexploit/F79E574D-30C8-5C52-A801-66FFA0610BAA  *EXPLOIT*
      CVE-2025-26465  6.8     https://vulners.com/cve/CVE-2025-26465
      1337DAY-ID-39918        6.8     https://vulners.com/zdt/1337DAY-ID-39918        *EXPLOIT*
      CVE-2023-51385  6.5     https://vulners.com/cve/CVE-2023-51385
      CVE-2023-48795  5.9     https://vulners.com/cve/CVE-2023-48795
      CVE-2020-14145  5.9     https://vulners.com/cve/CVE-2020-14145
      CC3AE4FC-CF04-5EDA-A010-6D7E71538C92    5.9     https://vulners.com/githubexploit/CC3AE4FC-CF04-5EDA-A010-6D7E71538C92  *EXPLOIT*
      54E1BB01-2C69-5AFD-A23D-9783C9D9FC4C    5.9     https://vulners.com/githubexploit/54E1BB01-2C69-5AFD-A23D-9783C9D9FC4C  *EXPLOIT*
      CVE-2016-20012  5.3     https://vulners.com/cve/CVE-2016-20012
      CVE-2025-32728  4.3     https://vulners.com/cve/CVE-2025-32728
      CVE-2021-36368  3.7     https://vulners.com/cve/CVE-2021-36368
      PACKETSTORM:140261      0.0     https://vulners.com/packetstorm/PACKETSTORM:140261      *EXPLOIT*
```

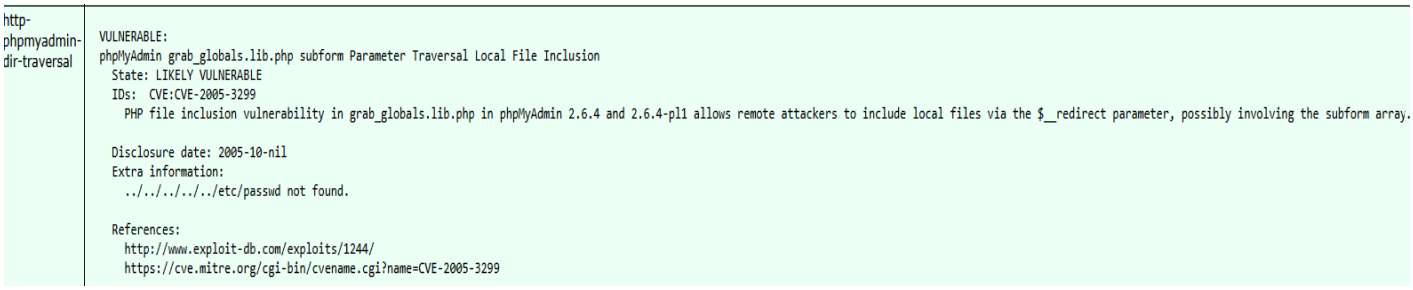
Para mitigar las vulnerabilidades críticas en OpenSSH 8.0, se recomienda seguir estas me didas:

- **Actualizar OpenSSH a una versión parcheada (≥ 9.3p2) :** El fallo CVE-2023-38408 fue corregido en OpenSSH 9.3p2, por lo que actualizar elimina la vulnerabilidad del agente PKCS-11 con rutas inseguras. [72, 73]

- **Limitar o bloquear el uso de PKCS-11/filtros de proveedores:**.Inicializar ssh-agent con una lista blanca vacía o restringida para los proveedores PKCS-11 (por ejemplo, ssh-agent-P ") reduce el riesgo de que librerías inseguras sean cargadas. [72, 73]

- **Usar salto seguro (ProxyJump)en lugar de reenvío de agente.)**:Enlugardereenviar el agente entre saltos, usar la directiva ProxyJump (o-J) permite conectar varios hosts intermedios sin exponer el agente. [74, 75]

Mitigación de LFI en phpMyAdmin grab_globals.lib.php (CVE-2005-3299)



```
http-phpmyadmin-dir-traversal

VULNERABLE:
phpMyAdmin grab_globals.lib.php subform Parameter Traversal Local File Inclusion
  State: LIKELY VULNERABLE
  IDs:  CVE:CVE-2005-3299
    PHP file inclusion vulnerability in grab_globals.lib.php in phpMyAdmin 2.6.4 and 2.6.4-pl1 allows remote attackers to include local files via the $__redirect parameter, possibly involving the subform array.

  Disclosure date: 2005-10-nil
  Extra information:
    ../../../../../etc/passwd not found.

  References:
    http://www.exploit-db.com/exploits/1244/
    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3299
```

Para protegerse contra la vulnerabilidad CVE-2005-3299, que permite incluir archivos locales mediante el parámetro $__redirect en phpMyAdmin 2.6.4 y 2.6.4-pl1, se recomiendan estas medidas:

- **Actualizar phpMyAdmin** a 2.6.4-pl2 o versiones posteriores La vulnerabilidad se corrige explícitamente en la versión 2.6.4-pl2 :contentReference. [phpMyAdmin - Security - PMASA-2005-4]
- **Deshabilitar parámetros no validados:** Revisar y eliminar el uso del parámetro $__redirect o cualquier componente de entrada no sanitizado en aplicaciones personalizadas que utilicen grab_globals.lib.php. [phpMyAdmin - Security - PMASA-2005-4]
- **Bloquear accesos a archivos críticos desde la web:** Mediante firewall, reglas en el servidor web o configuración de PHP, impedir el acceso o inclusión a rutas como /etc/passwd. [phpMyAdmin - Security - PMASA-2005-4]
- **Eliminar grab_globals.lib.php**: sino necesario instalaciones modernas phpMyAdmin, este archivo es obsoleto: eliminarlo evita el riesgo incluso en versiones afecta das. [phpMyAdmin - Security - PMASA-2005-4]

Mitigación de POODLE (CVE-2014-3566)

ssl-poodle

```
VULNERABLE:
SSL POODLE information leak
  State: VULNERABLE
  IDs:  BID:70574  CVE:CVE-2014-3566
        The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
        products, uses nondeterministic CBC padding, which makes it easier
        for man-in-the-middle attackers to obtain cleartext data via a
        padding-oracle attack, aka the "POODLE" issue.
  Disclosure date: 2014-10-14
  Check results:
    TLS_RSA_WITH_AES_128_CBC_SHA
  References:
    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
    https://www.securityfocus.com/bid/70574
    https://www.imperialviolet.org/2014/10/14/poodle.html
    https://www.openssl.org/~bodo/ssl-poodle.pdf
```

Para prevenir la fuga de información por desbordamiento de CBC en SSL 3.0, se recomiendan las siguientes acciones:

- **Deshabilitar completamente SSL 3.0:** en servidor y cliente. En Apache, NGINX, IIS,JBoss, WebLogic, etc., configúrelo para soportar sólo TLS 1.1/1.2/1.3. Evite cualquier mecanismo de downgrade incluido SSL 3.0 [SSL 3.0 Protocol Vulnerability and POODLE Attack | CISA].

- **Habilitar TLS_FALLBACK_SCSV:** para prevenir ataques de downgrade, aún si se conserva soporte temporal a SSL 3.0 [SSL 3.0 Protocol Vulnerability and POODLE Attack | CISA].

- **Evitar cifrados CBC en SSL 3.0**: utilizando preferentemente RC4 como parche temporal mientras se completan las actualizaciones (aunque RC4 también tiene fallos, por lo que debe considerarse una solución transitoria) [SSL 3.0 Protocol Vulnerability and POODLE Attack | CISA].

- **Aplicar parches y actualizaciones:** OpenSSL ≥ 1.0.1j, NSS ≥ 3.16.2.3, 3.17.1 y LibreSSL ≥ 2.1.1 implementan TLS-FALLBACK-SCSV y deshabilitan SSL 3.0 por defecto [SSL 3.0 Protocol Vulnerability and POODLE Attack | CISA].

Mitigación desmb-vuln-cve2009-3103

```
smb-vuln-cve2009-3103

VULNERABLE:
SMBv2 exploit (CVE-2009-3103, Microsoft Security Advisory 975497)
  State: VULNERABLE
  IDs:  CVE:CVE-2009-3103
        Array index error in the SMBv2 protocol implementation in srv2.sys in Microsoft Windows Vista Gold, SP1, and SP2,
        Windows Server 2008 Gold and SP2, and Windows 7 RC allows remote attackers to execute arbitrary code or cause a
        denial of service (system crash) via an & (ampersand) character in a Process ID High header field in a NEGOTIATE
        PROTOCOL REQUEST packet, which triggers an attempted dereference of an out-of-bounds memory location,
        aka "SMBv2 Negotiation Vulnerability."

  Disclosure date: 2009-09-08
  References:
    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103
    http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103
```

Para prevenir la vulnerabilidad CVE-2009-3103, que afecta a SMBv2 en Windows Server 2008 y Windows Vista SP1, se recomienda seguir las siguientes medidas:

- **Aplicar el boletín de seguridad MS09-050:** Microsoft publicó el boletín MS09-050 para corregir esta vulnerabilidad en SMBv2 [Boletín de seguridad de Microsoft MS09-050: crítico | Microsoft Learn].
- **Deshabilitar SMBv2 temporalmente**: Como medida temporal, se puede deshabilitar SMBv2 para mitigar el riesgo hasta aplicar el parche [Boletín de seguridad de Microsoft MS09-050: crítico | Microsoft Learn].
- **Actualizar a versiones no afectadas:** Migrar a versiones de Windows que no sean vulnerables (por ejemplo, Windows 7 RTM o Windows Server 2008 R2) evita la exposición [Boletín de seguridad de Microsoft MS09-050: crítico | Microsoft Learn].
- **Aplicar filtros de red:** Implementar controles en red para bloquear tráfico SMBv2 no autorizado y prevenir que lleguen paquetes manipulados al servidor afectado [Boletín de seguridad de Microsoft MS09-050: crítico | Microsoft Learn].

Mitigación Diffie-Hellman

ssl-dh-params	

```
VULNERABLE:
Diffie-Hellman Key Exchange Insufficient Group Strength
  State: VULNERABLE
    Transport Layer Security (TLS) services that use Diffie-Hellman groups
    of insufficient strength, especially those using one of a few commonly
    shared groups, may be susceptible to passive eavesdropping attacks.
  Check results:
    WEAK DH GROUP 1
          Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
          Modulus Type: Safe prime
          Modulus Source: RFC2409/Oakley Group 2
          Modulus Length: 1024
          Generator Length: 1024
          Public Key Length: 1024
  References:
    https://weakdh.org
```

Para prevenir la vulnerabilidad de Diffie-Hellman, se recomienda seguir las siguientes medidas:

- **Deshabilitar suites DH con tamaño de clave <2048 bits**: Asegúrate de que en la configuración del servidor no estén permitidos grupos temporales con prime menor de 2048 bits. Esto incluye evitar "export cipher suites" y modos antiguos que dejen negociar grupos débiles [Logjam: PFS Deployment Guide].
- **Usar parámetros DH personalizados y seguros:** Genera un nuevo archivo dhparam de al menos 2048 bits (preferiblemente 3072) para usar como grupo DH, en lugar de los parámetros por defecto compartidos que pueden ser comunes y susceptibles a ataques de precomputación [Logjam: PFS Deployment Guide].
- **Priorizar (Ephemeral) ECDHE / curvas elípticas:** Cambiar la negociación de clave a ECDHE (o DH elíptico) cuando sea posible, ya que estos ofrecen mayor seguridad y resisten mejor los ataques modernos como Logjam [Logjam: PFS Deployment Guide].
- **Actualizar protocolos y bibliotecas TLS/SSL.**: Asegúrate de usar versiones recientes del software que corrige vulnerabilidades DH débiles (OpenSSL, NSS, SChannel de Microsoft, Java, etc.). Muchas versiones recientes limpian los grupos débiles por defecto y permiten configurar parámetros más fuertes [Logjam: PFS Deployment Guide].
- **Revisar y restringir suites de cifrado:** Configurar el servidor para permitir sólo suites que utilicen cifrado fuerte, evitar aquellas que usen cifrados CBC débiles, evitar Anonymous DH, DES, RC4, etc. Garantizar que los algoritmos de intercambio de claves y firmas también sean seguros [Logjam: PFS Deployment Guide].

Mitigación HTTP verb tampering

http-method-tamper	VULNERABLE: Authentication bypass by HTTP verb tampering State: VULNERABLE (Exploitable) This web server contains password protected resources vulnerable to authentication bypass vulnerabilities via HTTP verb tampering. This is often found in web servers that only limit access to th common HTTP methods and in misconfigured .htaccess files. Extra information: URIs suspected to be vulnerable to HTTP verb tampering: /index.htm [GENERIC] References: http://www.imperva.com/resources/glossary/http_verb_tampering.html https://www.owasp.org/index.php/Testing_for_HTTP_Methods_and_XST_%28OWASP-CM-008%29 http://capec.mitre.org/data/definitions/274.html http://www.mkit.com.ar/labs/htexploit/

Para prevenir la vulnerabilidad de HTTP verb tampering, se recomienda seguir las siguientes medidas:

- **Restringir los métodos HTTP permitidos solo a los estrictamente necesarios** — por ejemplo, permitir únicamente GET y POST si la aplicación no usa otros métodos [WSTG - v4.1 | OWASP Foundation].
- **Configurar reglas de autorización que apliquen a todos los métodos** — asegurarse de que cualquier método que solicite acceso a recursos protegidos sea chequeado por la lógica de autenticación/autorización [Http verb tempering: bypassing web authentication and authorization | Infosec].
- **Deshabilitar métodos inseguros o poco usados** (TRACE, PUT, DELETE, etc.) si no se requieren, y bloquear verbos no estándar o personalizados [WSTG - v4.1 | OWASP Foundation].

Mitigación de Redirección Externa

URL	http://192.168.1.22/general/status.html
Método	POST
Ataque	https://4333334623948260514.owasp.org
Evidencia	https://4333334623948260514.owasp.org
Otra información	La respuesta contiene una redirección en su encabezado de ubicación que permite configurar una URL externa.
URL	http://192.168.1.22/general/status.html
Método	POST
Ataque	https://5709342025478090871.owasp.org
Evidencia	https://5709342025478090871.owasp.org
Otra información	La respuesta contiene una redirección en su encabezado de ubicación que permite configurar una URL externa.

Para proteger una pagina web contra ataques de redirección externa, se recomiendan las siguientes medidas:

- **Validación de entrada con lista blanca ("aceptar lo bueno conocido"):** Aceptar únicamente valores que cumplan estrictamente con las especificaciones (longitud, tipo, formato, rango, reglas de negocio). Rechazar o normalizar cualquier entrada que no se ajuste rigurosamente. No confiar únicamente en listas de denegación. [Input Validation - OWASP Cheat Sheet Series]
- **Mapeo mediante identificadores en lugar de URLs directas:** Usar identificadores fijos (por ejemplo, numéricos) que se traducen internamente en una URL confiable. Rechazar cualquier entrada que no corresponda al mapeo permitido. Ejemplo: ID 1 → "/login.asp", ID 2 → "https://www.example.com/" Herramientas como AccessReferenceMap de ESAPI facilitan este enfoque. [Input Validation - OWASP Cheat Sheet Series]
- **Lista blanca de dominios o URLs permitidos:** Solo redirigir a destinos que estén explícitamente autorizados. Cualquier intento de redirección fuera de esta lista debe rechazarse. [Unvalidated Redirects and Forwards - OWASP Cheat Sheet Series]
- **Página intermedia de advertencia antes de redireccionar:** Mostrar al usuario una página que indique claramente que está abandonando el sitio, pidiendo confirmación 147 vía clic o con un retardo (timeout) antes de proceder. Prevenir vulnerabilidades XSS en la generación de esa página. [Input Validation - OWASP Cheat Sheet Series]
- **Validación de todas las fuentes de entrada no confiables:** Considerar todos los vectores de entrada (parámetros URL, cookies, encabezados, formularios ocultos, variables de entorno, contenido desde APIs, correos, bases de datos, etc.) y aplicar validación rigurosa. [Unvalidated Redirects and Forwards - OWASP Cheat Sheet Series]
- **No confiar exclusivamente en listas negras:** las listas de denegación pueden ayudar a filtrar patrones maliciosos conocidos, pero no son suficientes debido a nuevas formas de ataque. Se deben usar solo como apoyo adicional. [Unvalidated Redirects and Forwards - OWASP Cheat Sheet Series]

Mitigación de ausencia de protección CSRF

URL	http://192.168.1.226/nwset/
Método	GET
Ataque	
Evidencia	<form name="frm" action="top.cgi" method="POST">
Otra información	No se ha encontrado ningún token Anti-CSRF [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] conocido en el siguiente formulario HTML: [Form 1: "admin_pass" "btn"].

Para proteger una página web contra la ausencia de protección CSRF, se recomiendan las siguientes medidas:

- **Identificación de operaciones sensibles y exigencia de confirmación explícita:** Para operaciones de alto riesgo (por ejemplo, modificación de datos, transacciones, configuración crítica), implementar un paso adicional donde el usuario confirme su intención (por ejemplo, mediante un diálogo o una página intermedia). Esto ayuda a impedir que una solicitud maliciosa ejecutada desde XSS pase inadvertida. [Cross-Site Request Forgery Prevention - OWASP Cheat Sheet Series]
- **Evitar el uso de métodos GET para acciones que cambian el estado:** Según OWASP, las operaciones que modifican el estado del servidor deben implementarse exclusivamente a través de peticiones POST, NO GET, para evitar ataques inadvertidos mediante etiquetas como o redirecciones automáticas. [Cross-Site Request Forgery Prevention - OWASP Cheat Sheet Series]
- **Uso del control de gestión de sesión de ESAPI con soporte para CSRF:** El módulo de gestión de sesión de ESAPI incluye automáticamente un token CSRF que puede ser validado en cada solicitud sensible. Este mecanismo asegura que sólo solicitudes legítimas, originadas desde la sesión activa del usuario, sean aceptadas. [Cross-Site Request Forgery Prevention - OWASP Cheat Sheet Series]
- **Asegurar que la solicitud provenga de la página esperada (validación de Referer/Origen):** Se debe verificar que la petición sensible provenga desde la página legítima. Si bien las cabeceras Referer o Origin pueden suprimirse por motivos de privacidad o romper funcionalidad al estar desactivadas, su ausencia puede también considerarse indicio de ataque y tratarse como sospechosa. [Cross-Site Request Forgery Prevention - OWASP Cheat Sheet Series]
- **Prevención de bypass mediante XSS**: Las protecciones anti-CSRF pueden ser anuladas si existe una vulnerabilidad XSS. Por lo tanto, es crucial mitigar XSS en todas las áreas de entrada no confiables, garantizando que los tokens CSRF no sean expuestos o robados. [Cross-Site Request Forgery Prevention - OWASP Cheat Sheet Series]

Mitigación de cabecera CSP ausente o débil

URL	http://192.168.1.22/general/status.html
Método	POST
Ataque	
Evidencia	
Otra información	

Para proteger una página web contra la ausencia de protección CSP, se recomiendan las siguientes medidas:

- **Inyección de la cabecera CSP desde el servidor:** Configurar el servidor para enviar una cabecera Content-Security-Policy en todas las respuestas HTTP (no solo en la página principal) para especificar qué recursos pueden cargarse. [Content Security Policy - OWASP Cheat Sheet Series]
- **Uso de política estricta con nonces o hashes:** Implementar una política CSP estricta utilizando nonces (valores únicos por respuesta) o hashes para autorizar solo scripts específicos y prevenir la ejecución de código inyectado. [Content Security Policy - OWASP Cheat Sheet Series]
- **Restricción de carga de recursos externos:** Configurar directivas como default‑src 'self' y especificar fuentes permitidas para scripts, estilos, imágenes y conexiones, reduciendo la superficie de ataque. [Content Security Policy (CSP) - HTTP | MDN]
- **Prevención de ataques como XSS, clickjacking y exfiltración de datos:** Una política CSP bien implementada limita la ejecución de scripts no autorizados, protege contra la carga fuera de origen y evita que datos sean enviados a servidores maliciosos. [Content Security Policy - OWASP Cheat Sheet Series]

Mitigación de exposición de listado de directorios

URL	http://192.168.1.226/license.txt/

Método	GET
Ataque	http://192.168.1.226/license.txt/
Evidencia	parent directory

Para proteger una pagina web contra posibilidad de ver la explotación de directorios, se recomiendan las siguientes medidas:

- **Deshabilitar el listado de directorios en el servidor web:** Configurar el servidor para que no muestre el contenido de un directorio si no existe un archivo índice. En Apache, por ejemplo, eliminar la directiva Indexes o usar Options -Indexes. En Nginx, establecer autoindex off. En Tomcat desactivar la opción listings en web.xml. [Disabling Directory Listing on Your Web Server – And Why It Matters | Acunetix]
- **Restringir acceso a directorios sensibles mediante permisos y control de acceso:** Implementar seguridad a nivel de servidor o sistema de archivos para asegurar que directorios críticos no sean accesibles públicamente. [Disabling Directory Listing on Your Web Server – And Why It Matters | Acunetix]
- **Evitar almacenamiento público de archivos sensibles o backups:** No colocar archivos de configuración, respaldos o metadatos dentro del directorio raíz público o en directorios accesibles sin protección. [Disabling Directory Listing on Your Web Server – And Why It Matters | Acunetix]

Mitigación de Cookie sin el atributo Samesite

URL	http://192.168.1.22/admin/password.html
Método	GET
Ataque	
Evidencia	Set-Cookie: AuthCookie
Otra información	

Para proteger una página web contra la posibilidad de que una cookie no tenga el atributo SameSite, se recomiendan las siguientes medidas:

- **Establecer explícitamente el atributo SameSite (idealmente Lax o Strict):** Esto asegura que la cookie solo se envíe en contextos seguros. Lax ofrece un equilibrio aceptable para la mayoría de aplicaciones, mientras que Strict brinda máxima protección. [Cookie Without SameSite Flag Detected | Tenable®]
- **Configurar SameSite=None solo con la bandera Secure:** Cuando se necesita enviar cookies en contextos entre sitios, asegúrese de incluir también Secure, ya que muchos navegadores modernos rechazan cookies SameSite=None sin esta directiva. [Cookie Without SameSite Flag Detected | Tenable®]
- **Entender la protección que ofrece SameSite contra CSRF:** Este atributo impide que navegadores envíen cookies sensibles en solicitudes iniciadas desde dominios externos, lo que refuerza la defensa frente a CSRF. [CWE - CWE-1275: Sensitive Cookie with Improper SameSite Attribute (4.18)]
- **Evitar el comportamiento inseguro por defecto en navegadores:** Aunque algunos navegadores pueden aplicar SameSite = Lax por defecto si no se especifica, la configuración explícita por parte del desarrollador garantiza el comportamiento esperado y evita fugas accidentales. [Cookie Without SameSite Flag Detected | Tenable®]
- **Mitigar riesgos complementarios, como CSRF y exfiltración:** Establecer SameSite = Lax / Strict ayuda a evitar que cookies sean enviadas automáticamente en contextos cross-site, reduciendo significativamente los riesgos de CSRF y fugas de datos. [CWE - CWE-1275: Sensitive Cookie with Improper SameSite Attribute (4.18)]

Mitigación ocultar la versión en la cabecera Server de la respuesta HTTP

URL	http://192.168.1.22/common/css/common.css
Método	GET
Ataque	
Evidencia	debut/1.30

Para proteger una página web contra la revelación de información sensible del servidor, se recomiendan las siguientes medidas:

- **Apache:** limitar la cabecera Server con ServerTokens Prod y ServerSignature Off, de modo que solo muestre "Apache" sin versión. [Configuring Your Web Server to Not Disclose Its Identity | Acunetix]
- **Nginx:** desactivar la exposición de versión con server_tokens off;. Para eliminar totalmente la cabecera Server, usar un módulo como Headers More o hacerlo a través de un proxy inverso/WAF. [nginx - Can I hide all server / os info? - Server Fault]
- **Eliminar/reescribir la cabecera en un WAF o proxy inverso** (p.ej., ModSecurity, Nginx/Traefik/Envoy al frente) cuando el servidor origen no pueda cambiarse. nginx - Can I hide all server / os info? - Server Fault]
- **Aplicar la medida también a páginas de error** (404, 500, etc.), donde la cabecera Server suele persistir. [nginx - Can I hide all server / os info? - Server Fault]

Mitigación Falta encabezado X-Content-Type-Options

URL	http://192.168.1.22/general/reflesh.html
Método	POST
Ataque	
Evidencia	

Para proteger una página web contra la falta del encabezado X-Content-Type-Options, se recomiendan las siguientes medidas:

- **Agregar X-Content-Type-Options:** nosniff en todas las respuestas HTTP, incluyendo páginas de error (ej. 401, 403, 500), para forzar que el navegador respete el tipo de contenido declarado. [ZAP – X-Content-Type-Options Header Missing]
- **Declarar correctamente el encabezado Content-Type en todas las páginas:** Esto ayuda a prevenir que el navegador realice MIME-sniffing por falta de información o ambigüedad. [HTTP Headers - OWASP Cheat Sheet Series]
- **Evitar ataques de tipo MIME-sniffing:** Si un atacante sube un archivo malicioso con extensión de imagen, podría ser interpretado como ejecutable si no se controla el comportamiento del navegador. [HTTP Security Headers: An Easy Way To Harden Your Web Applications]

Mitigación de gran redirección detectada

URL	http://192.168.1.22/general/status.html
Método	POST
Ataque	
Evidencia	
Otra información	Longitud URI de la cabecera de ubicación: 50 [/etc/passerror.html?url=%2Fgeneral%2Fstatus%2Ehtml]. Tamaño previsto de la respuesta: 350. Longitud del cuerpo de la respuesta: 7,579.

Para proteger una página web contra la gran redirección detectada (redirecciones inseguras), se recomiendan las siguientes medidas:

- **Validación estricta de destinos de redirección:** Solo permitir redirecciones a destinos predefinidos o internos autorizados. Utilizar una lista blanca en lugar de depender de parámetros enviados por el usuario. [Unvalidated Redirects and Forwards - OWASP Cheat Sheet Series]
- **Mapeo mediante identificadores internos:** En vez de aceptar rutas o URLs directamente, usar IDs fijos que sean traducidos internamente a URLs seguras. Esto impide que un usuario manipule el destino con rutas arbitrarias. [CWE - CWE-601: URL Redirection to Untrusted Site ('Open Redirect') (4.18)]
- **Firewall de aplicaciones o WAF:** Usar un firewall para detectar y bloquear redirecciones inseguras o poco comunes, especialmente cuando no se pueden corregir rápidamente en el código. [Unvalidated Redirects and Forwards - OWASP Cheat Sheet Series]
- **Página intermedia de advertencia para redirección externa:** Si se detecta que el destino de la redirección es externo o no confiable, mostrar al usuario una página intermedia que indique claramente a dónde se dirige y solicitar confirmación antes de proceder. [CWE - CWE-601: URL Redirection to Untrusted Site ('Open Redirect') (4.18)]

Mitigación de divulga información mediante un campo(s) de encabezado de respuesta HTTP X-Powered-By

URL	http://192.168.1.2/
Método	GET
Ataque	
Evidencia	X-Powered-By: ASP.NET

Para proteger una pagina web contra la exposición de errores de divulgación, se recomiendan las siguientes medidas:

- **Quitar el encabezado X-Powered-By desde el IIS:** Manager, sección “HTTP Response Headers”. [104] En el archivo web.config, bajo la sección system.webServer, usar customHeaders para eliminar el encabezado X-Powered-By. [Error Handling - OWASP Cheat Sheet Series]
- **Deshabilitar encabezados de versión:** como X-AspNet-Version mediante la propiedad enableVersionHeader = false en la sección httpRuntime. [Error Handling - OWASP Cheat Sheet Series]
- **Eliminar/ocultar encabezado Server:** usando la configuración de IIS (por ejemplo, en IIS 10 mediante requestFiltering removeServerHeader = true). [Error Handling - OWASP Cheat Sheet Series]
- **Usar reglas de reescritura o un módulo/middleware:** que intercepte la respuesta antes de enviarla al cliente y elimine los encabezados sensibles. [Error Handling - OWASP Cheat Sheet Series]

Mitigación de revelación de Ip privada

URL	http://192.168.1.22/general/find.html
Método	GET
Ataque	
Evidencia	192.168.1.25
Otra información	192.168.1.25 192.168.1.25 192.168.1.30 192.168.1.30 192.168.1.31 192.168.1.31 192.168.1.32 192.168.1.32 192.168.1.33 192.168.1.33 192.168.1.37 192.168.1.37 192.168.1.203 192.168.1.203 192.168.1.208 192.168.1.208 192.168.1.101 192.168.1.199 192.168.1.210

Para proteger una página web contra la revelación de IP privada, se recomiendan las siguientes medidas:

- **Eliminar IPs privadas del contenido de respuesta:** Asegúrese de que no se filtren direcciones IP internas o nombres de host en el código visible al cliente (por ejemplo, en páginas, errores o mensajes de debug). [ZAP – Private IP Disclosure]
- **Utilizar comentarios del lado del servidor**: En lugar de insertar información sensible en comentarios HTML o JavaScript, use comentarios en tecnologías como JSP, ASP o PHP, que no serán visibles para el navegador del cliente. [ZAP – Private IP Disclosure]
- **Evitar exposición en mensajes generados dinámicamente**: Las IP internas no deben aparecer tampoco en logs, mensajes de error, trazas o vistas de depuración que puedan ser visibles en el entorno de producción. [ZAP – Private IP Disclosure]

Mitigación de autenticación débil

URL	http://192.168.1.210/hp/device/info_config_AirPrint.html?menu=AirPrintStatus&tab=Networking
Método	GET
Ataque	
Evidencia	www-authenticate: Basic realm="HP LaserJet device (password only, no username required)@NPID14F94"
Otra información	

Para proteger una página web contra la autenticación débil, se recomiendan las siguientes medidas:

- **Forzar uso de HTTPS (TLS) en todo el sitio,** redirigiendo todo HTTP a HTTPS. Use HSTS para que el navegador insista en HTTPS incluso si se escribe manualmente HTTP. [Basic authentication over HTTP - Vulnerabilities - Acunetix]
- **Evitar usar Basic o Digest como método principal,** priorizando autenticación basada en sesiones (cookies seguras) o tokens (Bearer, OAuth2) bajo HTTPS. [authentication - TLS to secure Basic HTTP Auth - Information Security Stack Exchange]

Mitigación de credenciales capturadas

URL	http://192.168.1.210/set_config_password.html?tab=System&menu=Passwd
Método	GET
Ataque	
Evidencia	
Otra información	[GET] [http://192.168.1.210/set_config_password.html?tab=System&menu=Passwd] utiliza el mecanismo de autenticación inseguro [Basic], revelando el nombre de usuario [admin] y contraseña [].

Para proteger una página web contra las filtraciones de credenciales, se recomiendan las siguientes medidas:

- **Implementar el uso de protocolos seguros** como HTTPS en lugar de HTTP para garantizar el cifrado de las credenciales en tránsito. [HTTP Headers - OWASP Cheat Sheet Series]
- **Sustituir la autenticación básica o Digest** por mecanismos más seguros como OAuth 2.0, JWT o autenticación basada en certificados. [OAuth 2.0 Security Best Current Practice]
- **Configurar el servidor** para que todas las comunicaciones requieran el uso de TLS 1.2 o superior, evitando versiones inseguras. [HTTP Headers - OWASP Cheat Sheet Series]
- **Emplear autenticación multifactor (MFA)** para añadir una capa adicional de seguridad frente a robo de credenciales. Rotar periódicamente las credenciales y aplicar políticas de contraseñas robustas (longitud, complejidad y caducidad). [OAuth 2.0 Security Best Current Practice]
- **Monitorizar intentos de acceso sospechosos y aplicar sistemas de detección de intrusiones** (IDS/IPS) para detectar ataques Man-In-The-Middle. [HTTP Headers - OWASP Cheat Sheet Series]

Mitigación de Librería JS vulnerable

URL	http://192.168.1.210/hp/device/jquery.js
Método	GET
Ataque	
Evidencia	* jQuery JavaScript Library v1.3.2

Para proteger una página web contra el uso de una librería JS vulnerable, se recomiendan las siguientes medidas:

- **Actualizar a una versión segura**, como jQuery 3.5.0 o superior, donde se corrigieron vulnerabilidades como CVE-2020-11023. [CISA Adds Five-Year-Old jQuery XSS Flaw to Exploited Vulnerabilities List]
- **Sanitizar HTML de entrada de usuario** antes de insertarlo en el DOM (por ejemplo, usando DOMPurify antes de .html()), para evitar inyección de código malicioso. [Security Goals & Threat Model · cure53/DOMPurify Wiki · GitHub]

Mitigación de Inyección Remota de Comandos del Sistema Operativo

URL	http://192.168.1.199/web/guest/es/websys/webArch/waSearchFirst.cgi?wimToken=251565448%3Bsleep+1.0%3B
Método	GET
Ataque	251565448;sleep 15;
Evidencia	
Otra información	La regla de escaneo pudo controlar el tiempo de respuesta de la aplicación enviando [251565448;sleep 15;] al sistema operativo que ejecuta esta aplicación.

Para proteger una pagina web contra la inyección remota de comandos del sistema operativo, se recomiendan las siguientes medidas:

- **Evitar por completo la ejecución de comandos del sistema operativo** desde el código de la aplicación cuando sea posible. Utilizar APIs seguras o bibliotecas del lenguaje para lograr la funcionalidad requerida sin invocar comandos externos. [OS Command Injection Defense - OWASP Cheat Sheet Series]
- **Aplicar validación fuerte de la entrada del usuario:** usar listas blancas (whitelists) de valores permitidos, asegurarse de que la entrada tiene el formato esperado (por ejemplo, números, rutas seguras, direcciones IP) y rechazar cualquier dato que no coincida. [OAuth 2.0 Security Best Current Practice]
- **Emplear sanitización segura,** usando funciones como escapeshellarg() (en PHP) u equivalentes, que evitan que metacaracteres de shell sean interpretados como comandos. [OS Command Injection Defense - OWASP Cheat Sheet Series]
- **Rechazar o sanitizar expresamente metacaracteres peligrosos** como ;, , como parte de la validación o sanitización. [OS Command Injection Defense - OWASP Cheat Sheet Series]

Mitigación de falta encabezado AnticlickJacking

URL	http://192.168.1.199/web/entry/es/address/adrsList.cgi
Método	GET
Ataque	
Evidencia	
Otra información	

Para proteger una página web contra la falta de cabecera Anti-ClickJacking, se recomiendan las siguientes medidas:

- **Usar la cabecera Content-Security-Policy** con la directiva frame-ancestors, por ejemplo: Content-Security-Policy: frame-ancestors 'none'; Esto bloquea completamente que la página sea mostrada dentro de un marco externo. [Clickjacking Defense - OWASP Cheat Sheet Series]
- **Tener presente que la directiva CSP** frame-ancestors tiene prioridad sobre X-Frame-Options en navegadores modernos, de modo que quienes soportan CSP ignorarán la directiva XFrame-Options cuando ambas estén presentes. [Clickjacking Defense - OWASP Cheat Sheet Series]

Mitigación de mostrar errores de aplicación

URL	http://192.168.1.199/web/entry/es/webdocbox
Método	GET
Ataque	
Evidencia	HTTP/1.1 500 Internal Server Error

Para proteger una pagina web contra la exposición de errores de divulgación, se recomiendan las siguientes medidas:

- **No mostrar mensajes de error completos** (como rutas de archivo o stack trace) al usuario. [Error Handling - OWASP Cheat Sheet Series]
- **Mostrar errores genéricos al usuario final** y **guardar detalles técnicos solo en logs internos seguros**. [Error Handling - OWASP Cheat Sheet Series]
- **Configurar páginas de error personalizadas** (por ejemplo, 500, fallo inesperado) en la aplicación o servidor. [Error Handling - OWASP Cheat Sheet Series]

Mitigación de vulnerabilidad de métodos permitidos

URI	/
HTTP Method	OPTIONS
Description	OPTIONS: Allowed HTTP Methods: POST, OPTIONS .
Test Links	http://192.168.1.37:80/ http://192.168.1.37:80/
References	

Para mitigar las vulnerabilidades de metodos permitidos, se recomienda seguir estas medidas:

- **Restringir los métodos HTTP permitidos** a los estrictamente necesarios (por ejemplo, solo POST y GET), y devolver 405 Method Not Allowed para los demás [Why should the OPTIONS method not be allowed on an HTTP server? - Information Security Stack Exchange].
- **Deshabilitar o proteger métodos** potencialmente peligrosos como PUT, DELETE, TRACE, CONNECT que no sean requeridos por la impresora [Why should the OPTIONS method not be allowed on an HTTP server? - Information Security Stack Exchange].
- **Si no es necesario, considerar desactivar el método OPTIONS**, ya que puede facilitar la enumeración de métodos habilitados y aumentar la superficie de ataque. Solo mantenerlo activo si hay una razón funcional clara, como CORS o APIs REST [Why should the OPTIONS method not be allowed on an HTTP server? - Information Security Stack Exchange].

Mitigación de falta de configuración de la cabecera X-Frame-Options

URI	/#wp-config.php#
HTTP Method	GET
Description	/#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
Test Links	http://192.168.1.226:80/#wp-config.php# http://192.168.1.226:80/#wp-config.php#
References	

Para prevenir la fuga de información por acceso a un archivo con credenciales, se recomiendan las siguientes acciones:

- **Mover el archivo wp-config.php** uno o más niveles por encima del directorio raíz (web root), de modo que WordPress todavía lo encuentre, pero que no sea accesible directamente desde web [Clickjacking Defense - OWASP Cheat Sheet Series].
- **Configurar el servidor (Apache, Nginx, etc.) o el archivo .htaccess** para denegar explícitamente el acceso HTTP al archivo, generando una respuesta 403 Forbidden [107, 108].
- **Establecer permisos restrictivos en wp-config.php,** como 400 o 600, limitando su lectura exclusivamente al propietario . Evitar renombrar el archivo mediante sufijos como .bak o .old, ya que esto puede impedir que se interprete como PHP y permitir su descarga en texto plano [Clickjacking Defense - OWASP Cheat Sheet Series]

Mitigación Cookie establecida sin el atributo HttpOnly y Secure

URI	/
HTTP Method	GET
Description	/: Cookie cookieOnOffChecker created without the httponly flag.
Test Links	http://192.168.1.101:80/ http://192.168.1.101:80/
References	https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies

Para prevenir la vulnerabilidad de la cookie, se recomienda seguir las siguientes medidas:

- **Establecer siempre el atributo HttpOnly en cookies sensibles** (por ejemplo, de sesión o autenticación), para impedir que JavaScript pueda acceder a su valor [Secure cookie configuration - Security | MDN, WSTG - v4.1 | OWASP Foundation].
- **Enviar todas las cookies junto con el atributo Secure**, garantizando que solo se transmitan a través de conexiones HTTPS seguras [Secure cookie configuration - Security | MDN, WSTG - v4.1 | OWASP Foundation].
- **Configurar atributos adicionales** como SameSite=Strict o SameSite=Lax para mitigar ataques tipo CSRF [Secure cookie configuration - Security | MDN, WSTG - v4.1 | OWASP Foundation].
- **Revisar y asegurar que las cookies sensibles** expiren lo antes posible mediante Max-Age o Expires [109]. Ajustar los atributos de Domain y Path al valor más restrictivo necesario para el funcionamiento, limitando el alcance de envío de cookies [Secure cookie configuration - Security | MDN, WSTG - v4.1 | OWASP Foundation].

Mitigación de revelación de información del servidor

URI	/
HTTP Method	GET
Description	: Server banner changed from 'Web-Server/3.0' to 'RICOH SERVER/1.0'.
Test Links	http://192.168.1.101:80/ http://192.168.1.101:80/
References	

Para prevenir la vulnerabilidad de la revelación de información del servidor , se recomienda seguir las siguientes medidas:

- **Ocultar o modificar la cabecera Serve**r para no revelar el nombre exacto o versión del servidor (por ejemplo, usar un valor genérico como “Server: secure-device”). [Error Handling - OWASP Cheat Sheet Series]
- **Configurar el servidor,** si es posible, para desactivar la divulgación de versión mediante directivas como servertokens off en Nginx u opciones equivalentes en Apache. [Error Handling - OWASP Cheat Sheet Series]
- **Recompilar el software del servidor** (como Nginx) con módulos que permitan el control del valor de la cabecera Server, como HttpHeadersMoreModule. [Error Handling - OWASP Cheat Sheet Series]
- **Usar un dispositivo intermedio** (p. ej. proxy inverso o WAF) que interfiera la cabecera Server, reemplazándola o eliminándola antes de que llegue al cliente. [Error Handling - OWASP Cheat Sheet Series]

Mitigación de revelación de información por medio de UPnP

URI	/bmlinks/ddf.xml
HTTP Method	GET
Description	/bmlinks/ddf.xml: Device UPnP XML file found, which may leak device information.
Test Links	http://192.168.1.101:80/bmlinks/ddf.xml http://192.168.1.101:80/bmlinks/ddf.xml
References	

Para prevenir la vulnerabilidad de la revelación de información del servidor, se recomienda seguir las siguientes medidas:

- **Deshabilitar UPnP completamente si no es necesario**, especialmente en routers o impresoras con interfaz web, reduciendo la superficie de ataque [What Is UPnP and Why Is It a Security Risk? - SecurityScorecard].
- **Evitar que los archivos XML de descripción UPnP** como ddf.xml estén accesibles externamente, restringiéndolos solo a la red local mediante configuración o firewall [[DIY] Learn How to Secure WP-Config File In 5 Minutes].
- **Segmentar la red y aislar los dispositivos con UPnP** en una VLAN dedicada, minimizando el impacto ante una posible explotación [What Is UPnP and Why Is It a Security Risk? - SecurityScorecard].
- **Utilizar firewalls o IPS/WAF** para bloquear tráfico UPnP o SSDP (puerto UDP 1900), impidiendo consultas remotas o automatizadas [What Is UPnP and Why Is It a Security Risk? - SecurityScorecard, [DIY] Learn How to Secure WP-Config File In 5 Minutes].

Mitigación de cabecera Strict-Transport-Security (HSTS) no esté definida para TLS

URI	/
HTTP Method	GET
Description	/: The site uses TLS and the Strict-Transport-Security HTTP header is not defined.
Test Links	https://192.168.1.13:443/ https://192.168.1.13:443/
References	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security

Para prevenir la vulnerabilidad de que la cabecera Strict-Transport-Security (HSTS) no esté definida para TLS, se recomienda seguir las siguientes medidas:

- **Revisar que todos los dominios y subdominios** del sitio tengan certificados TLS válidos, sin errores (nombre, caducidad, CA de confianza), ya que HSTS solo funciona correctamente si HTTPS está bien configurado [HTTP Strict Transport Security - OWASP Cheat Sheet Series, The Importance of a Proper HTTP Strict Transport Security Implementation on Your Web Server | Qualys].
- **Evitar contenido mixto ("mixed content"):** asegurarse de que todas las imágenes, scripts, hojas de estilo, recursos externos, etc., se carguen mediante HTTPS, para que no haya desestabilización de la página debido a recursos inseguros [HTTP Strict Transport Security - OWASP Cheat Sheet Series, How to Quickly Fix Mixed Content Warnings (HTTPS/SSL)].
- **Implementar redirecciones permanentes (301) desde HTTP a HTTPS**, de modo que no se permita acceso inseguro público y asegurar que las peticiones HTTP siempre acaben usando HTTPS antes de servir contenido [HTTP Strict Transport Security - OWASP Cheat Sheet Series, The Importance of a Proper HTTP Strict Transport Security Implementation on Your Web Server | Qualys].

Migación de error de que el nombre del host no coincide con los nombres del certificado (CWE-297: Validación incorrecta de certificado con desajuste de host)

URI	/
HTTP Method	GET
Description	Hostname '192.168.1.2' does not match certificate's names: winserver.
Test Links	https://192.168.1.2:8888/ https://192.168.1.2:8888/
References	https://cwe.mitre.org/data/definitions/297.html

URI	/
HTTP Method	GET
Description	Hostname '192.168.1.2' does not match certificate's names: serverofiauto.
Test Links	https://192.168.1.2:8391/ https://192.168.1.2:8391/
References	https://cwe.mitre.org/data/definitions/297.html

Para prevenir la vulnerabilidad de que el nombre del host no coincide con los nombres del certificado, se recomienda seguir las siguientes medidas:

- **Emitir un certificado TLS/SSL cuyo Common Name (CN) o los Subject Alternative Names (SAN)** incluyan explícitamente el nombre o dirección que usarán los usuarios o clientes para acceder al servicio (por ejemplo "192.168.1.2" si lo acceden por IP, o "serverofiauto" si lo acceden por ese nombre) [Qué es un error de falta de coincidencia de nombre común SSL, What is the Subject Alternative Name (SAN)? - DNSimple Help].
- **No depender solo del CN, usar SANs adecuadamente**, ya que CN por sí solo puede no cubrir todos los nombres / IPs usadas [Qué es un error de falta de coincidencia de nombre común SSL, What is the Subject Alternative Name (SAN)? - DNSimple Help].
- **Si se está accediendo al servidor por dirección IP,** considerar emitir un certificado que incluya la IP en los SANs; muchos certificados solo cubren nombres de dominio y no direcciones IP [What is the Subject Alternative Name (SAN)? - DNSimple Help].
- **Configurar DNS y / o los nombres de host usados** por los clientes para que coincidan con los nombres incluidos en el certificado, o bien usar nombres de host en vez de IP si esto facilita tener el certificado correcto [Qué es un error de falta de coincidencia de nombre común SSL].
- **Verificar después de aplicar los cambios (instalar el certificado nuevo)** con herramientas como "SSL Checker", navegadores web o con openssl s-client para asegurarse de que al acceder por el host se presenta un certificado válido cuyo nombre coincide [Qué es un error de falta de coincidencia de nombre común SSL, What is the Subject Alternative Name (SAN)? - DNSimple Help]

Mitigación de devolver falsos positivos por medio de método JUNK

URI	/
HTTP Method	XJWDFUNS
Description	/: Web Server returns a valid response with junk HTTP methods which may cause false positives.
Test Links	https://192.168.1.2:8888/ https://192.168.1.2:8888/
References	

Para prevenir la vulnerabilidad de devolver falsos positivos por medio de método JUNK, se recomienda seguir las siguientes medidas:

- **Configurar el servidor o aplicación para rechazar explícitamente métodos HTTP** desconocidos o personalizados que no sean necesarios (por ejemplo "FOO", "BAR", "XJWD…"), devolviendo un estado 405 Method Not Allowed [web application - How can I test that I have correctly disabled unnecessary HTTP methods? - Information Security Stack Exchange , 405 Method Not Allowed: What It Is and How to Fix It?].
- **Revisar la configuración del servidor web / framework** para asegurarse de que solo los métodos necesarios estén habilitados (por ejemplo GET, POST, HEAD, OPTIONS si se requiere CORS), y deshabilitar los métodos inseguros o innecesarios como PUT, DELETE, TRACE, CONNECT, etc. [web application - How can I test that I have correctly disabled unnecessary HTTP methods? - Information Security Stack Exchange , 405 Method Not Allowed: What It Is and How to Fix It?].
- **Implementar restricción a nivel de seguridad,** por ejemplo mediante filtros, reglas en el archivo de configuración (por ejemplo web.xml en aplicaciones Java/Servlets) usando "security-constraint" para bloquear métodos específicos [web application - How can I test that I have correctly disabled unnecessary HTTP methods? - Information Security Stack Exchange].

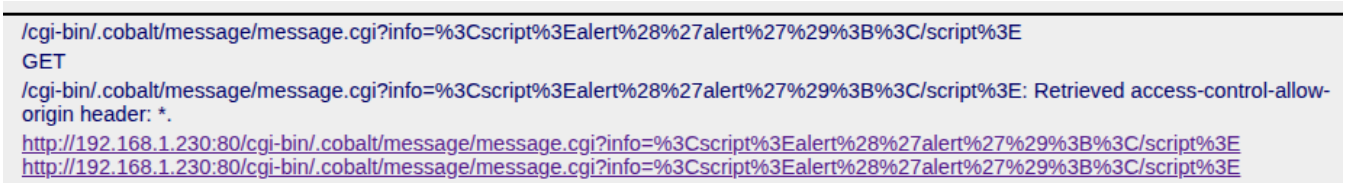
Mitigación de Directory Traversal en Cisco ACS

URI	/..\..\..\..\..\..\temp\temp.class
HTTP Method	GET
Description	/..\..\..\..\..\..\temp\temp.class: Cisco ACS 2.6.x and 3.0.1 (build 40) allows authenticated remote users to retrieve any file from the system. Upgrade to the latest version.
Test Links	https://192.168.1.2:8888/..\..\..\..\..\..\temp\temp.class https://192.168.1.2:8888/..\..\..\..\..\..\temp\temp.class
References	

Para prevenir la vulnerabilidad que permite a usuarios autenticados recuperar archivos arbitrarios del sistema mediante rutas como /temp/temp.class, se recomienda seguir las siguientes medidas:

- **Actualizar Cisco ACS** a la versión más reciente disponible, que contenga el parche que corrige la vulnerabilidad. Versiones antiguas como 2.6.x y 3.0.1 (build 40) están afectadas [Cisco Secure ACS RMI Arbitrary File Read Vulnerability].
- **Validar todas las entradas de las rutas** (path inputs) usadas por la aplicación, para impedir secuencias de escape como $o\gamma$ o uso de nombres que permitan escapar del directorio raíz. Filtrar o normalizar los paths antes de acceder al sistema de archivos. [Cisco Secure ACS RMI Arbitrary File Read Vulnerability].
- **Configurar adecuadamente los permisos del sistema de archivos** de modo que los usuarios autenticados sólo tengan acceso de lectura/escritura a los ficheros estrictamente necesarios, y que los directorios sensibles (fuera del web root) no sean accesibles por la interfaz web. [Cisco Secure ACS RMI Arbitrary File Read Vulnerability]
- **Limitar el acceso al módulo o interfaz vulnerable a sólo usuarios de confianza**, mediante reglas de firewall, listas de control de acceso (ACL) u otras barreras de red, mientras no se aplique la actualización. [Cisco Secure ACS RMI Arbitrary File Read Vulnerability].

Mitigación de cabecera Access-Control-Allow-Origin con wildcard y posible XSS



Para prevenir la vulnerabilidad que permite la cabecera Access-Control-Allow-Origin: junto con inyección de contenido (como en el ejemplo info= %3Cscript %3Ealert(ˣˣˣˣˣˣ) %3C/script %3E), se recomienda seguir las siguientes medidas:

- **Restringir el valor de la cabecera Access-Control-Allow-Origin** a orígenes específicos de confianza en lugar de usar el comodín *. Solo dominios seguros y verificados deberían poder hacer solicitudes cross-origin [WSTG - Latest | OWASP Foundation, CORS Errors Demystified: How to Fix Cross-Origin Issues.]
- **Evitar permitir credenciales** (cookies, sesiones, autenticaciones) cuando se usa Access-Control-Allow*. Si se requieren credenciales, la política CORS no debe usar * sino un origen concreto [WSTG - Latest | OWASP Foundation, CORS Errors Demystified: How to Fix Cross-Origin Issues].
- **Validar y filtrar toda entrada** que luego se refleje en la respuesta para evitar inyección de scripts o contenido no deseado. En particular, cuando se permite un parámetro como info que luego se envía al cliente, asegurarse de escaparlo adecuadamente. [CORS Errors Demystified: How to Fix Cross-Origin Issues]

Mitigación de uso de cabecera Content-Type: text/plain con contenido inseguro

/cgi-bin/.cobalt/message/message.cgi?info=%3Cscript%3Ealert%28%27alert%27%29%3B%3C/script%3E
GET
/cgi-bin/.cobalt/message/message.cgi?info=%3Cscript%3Ealert%28%27alert%27%29%3B%3C/script%3E: Uncommon header ' content-type' found, with contents: text/plain.
http://192.168.1.230:80/cgi-bin/.cobalt/message/message.cgi?info=%3Cscript%3Ealert%28%27alert%27%29%3B%3C/script%3E
http://192.168.1.230:80/cgi-bin/.cobalt/message/message.cgi?info=%3Cscript%3Ealert%28%27alert%27%29%3B%3C/script%3E

Para prevenir la vulnerabilidad asociada al uso de la cabecera Content-Type: text/plain cuando se devuelve contenido que puede incluir código (por ejemplo un script), se recomienda seguir las siguientes medidas:

- **Establecer el Content-Type correcto acorde al contenido devuelto**: Si la respuesta incluye HTML o fragmentos de código que el navegador pueda interpretar, usar text/html; charset=UTF-8 en vez de text/plain. [ZAP – X-Content-Type-Options Header Missing]
- **Añadir la cabecera de respuesta X-Content-Type-Options:** nosniff para evitar que navegadores realicen "MIME sniffing" (interpretación automática del contenido), lo que puede permitir ejecución de scripts aunque el Content-Type sea text/plain. [ZAP – X-Content-Type-Options Header Missing]

Mitigación de exposición de la utilidad de configuración BIG-IP vía bigconf.cgi

Para prevenir la vulnerabilidad que permite a usuarios acceder a la utilidad de configuración (o archivos relacionados) mediante el script /cgi-bin/bigconf.cgi, se recomienda seguir las siguientes medidas:

- **Actualizar BIG-IP a la versión más reciente** que ya no incluya la vulnerabilidad de bigconf.cgi o que tenga parches publicados para ese script [web application - How can I test that I have correctly disabled unnecessary HTTP methods? - Information Security Stack Exchange].
- **Restringir el acceso al recurso bigconf.cgi** de modo que exclusivamente usuarios con los permisos adecuados (por ejemplo administradores) lo puedan ejecutar, mediante autenticación sólida y autorizaciones [web application - How can I test that I have correctly disabled unnecessary HTTP methods? - Information Security Stack Exchange].
- **Deshabilitar o eliminar bigconf.cgi** si no es necesario para la operación normal del sistema; si la funcionalidad que ofrece no es usada, eliminarla reduce la superficie de ataque[web application - How can I test that I have correctly disabled unnecessary HTTP methods? - Information Security Stack Exchange].

Mitigación de ejecución arbitraria de comandos mediante webdist.cgi

Para prevenir la vulnerabilidad que permite a atacantes remotos ejecutar comandos arbitrarios mediante el parámetro distloc de webdist.cgi, se recomienda seguir las siguientes medidas:

- **Eliminar o deshabilitar el script webdist.cgi** si no es estrictamente necesario para la operación del sistema. [web application - How can I test that I have correctly disabled unnecessary HTTP methods? - Information Security Stack Exchange]
- **Restringir el acceso a webdist.cgi** a redes confiables mediante firewall, reglas de red, listas de control de acceso, etc. No exponerlo públicamente si no es imprescindible. [web application - How can I test that I have correctly disabled unnecessary HTTP methods? - Information Security Stack Exchange]

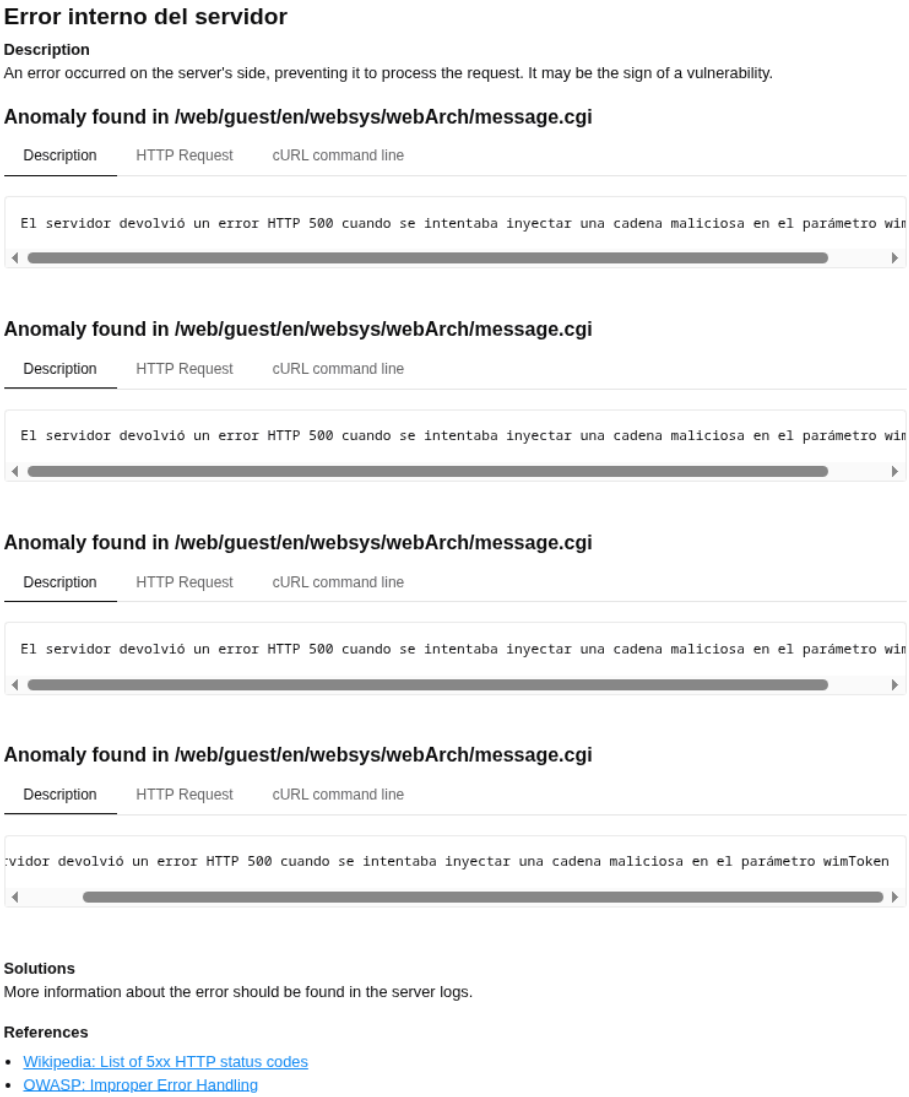
Mitigación de divulgación de archivos arbitrarios mediante pfdispaly.cgi

```
/cgi-bin/pfdisplay.cgi?../../../../../../etc/passwd
GET
/cgi-bin/pfdisplay.cgi?../../../../../../etc/passwd: Comes with IRIX 6.2-6.4; allows to run arbitrary commands.
```

Para prevenir la vulnerabilidad que permite a usuarios remotos leer cualquier archivo del sistema usando rutas como ../../../../etc/passwd vía pfdispaly.cgi, se recomienda seguir las siguientes medidas:

- **Validar la entrada del parámetro** que recibe la ruta en el script para eliminar o rechazar secuencias de escape. [Cisco Secure ACS RMI Arbitrary File Read Vulnerability]
- **Ajustar los permisos de archivos y directorios sensibles** de modo que el usuario que corre el servidor web (o CGI) no tenga acceso de lectura (o permisos innecesarios) sobre los ficheros fuera del directorio web root. [Cisco Secure ACS RMI Arbitrary File Read Vulnerability]
- **Restringir el acceso al script pfdispaly.cgi** solo a usuarios autenticados y autorizados, o mediante restricciones de red (firewall, listas blancas de IP) si el uso público no es necesario. [Cisco Secure ACS RMI Arbitrary File Read Vulnerability]

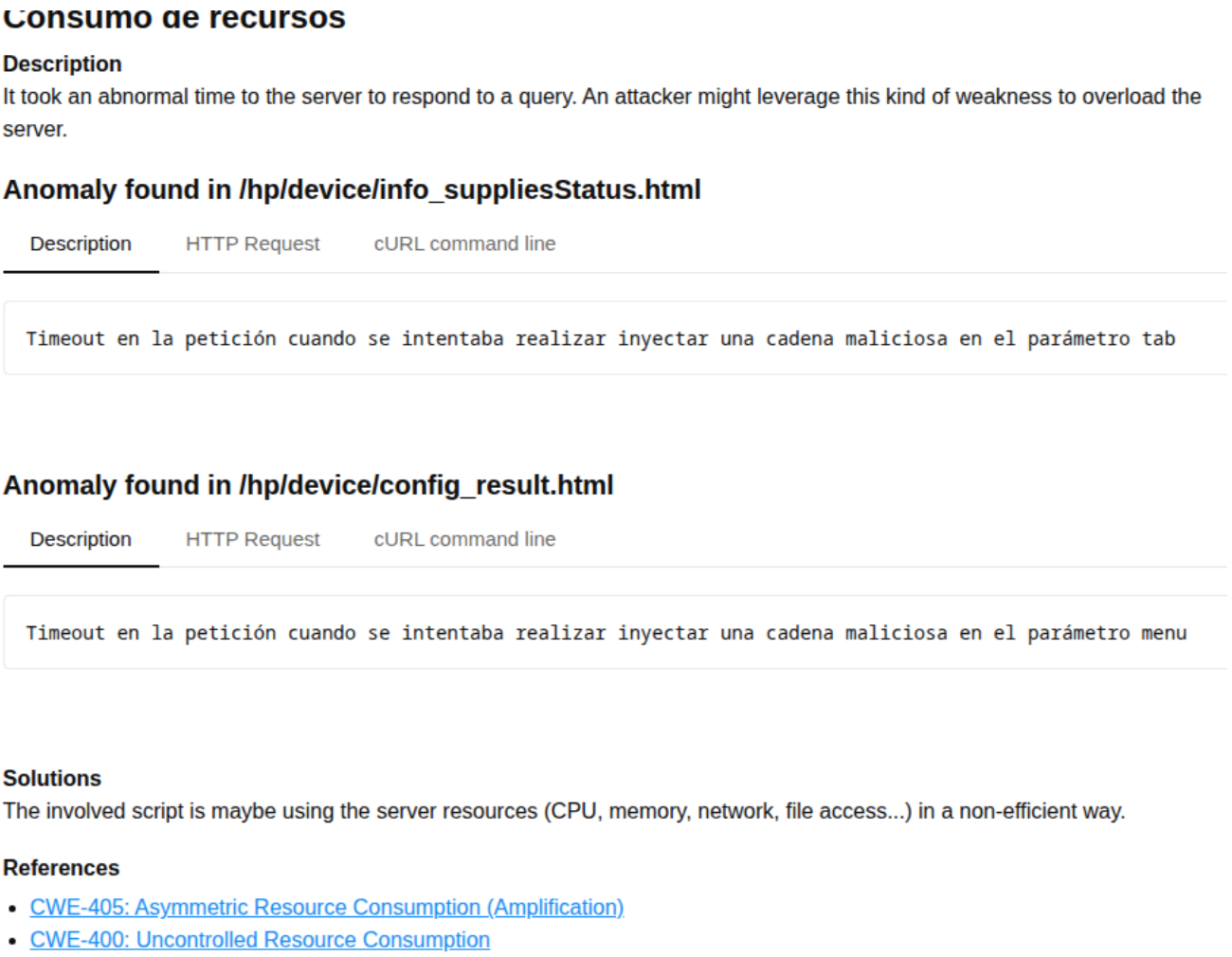
Mitigación de consumo de recursos sin control



Para proteger una pagina web contra la ausencia de protección contra el consumo de recursos, se recomiendan las siguientes medidas:

- **Imponer límites de tiempo (timeouts) y uso máximo de CPU/memoria por petición.** [API4:2019 Lack of Resources & Rate Limiting - OWASP API Security Top 10]
- **Validar y sanear parámetros** (longitud máxima, formato esperado). [API4:2019 Lack of Resources & Rate Limiting - OWASP API Security Top 10]
- **Controlar la complejidad de operaciones internas** (evitar loops costosos, recursividad excesiva). [API4:2019 Lack of Resources & Rate Limiting - OWASP API Security Top 10]

Mitigación de errores internos (HTTP 500) inducidos por inyección



Consumo de recursos

Description

It took an abnormal time to the server to respond to a query. An attacker might leverage this kind of weakness to overload the server.

Anomaly found in /hp/device/info_suppliesStatus.html

Description | HTTP Request | cURL command line

```
Timeout en la petición cuando se intentaba realizar inyectar una cadena maliciosa en el parámetro tab
```

Anomaly found in /hp/device/config_result.html

Description | HTTP Request | cURL command line

```
Timeout en la petición cuando se intentaba realizar inyectar una cadena maliciosa en el parámetro menu
```

Solutions

The involved script is maybe using the server resources (CPU, memory, network, file access...) in a non-efficient way.

References

- CWE-405: Asymmetric Resource Consumption (Amplification)
- CWE-400: Uncontrolled Resource Consumption

Para mitigar este tipo de vulnerabilidad, se sugieren las siguientes acciones:

- **Validar y sanear el parámetro wimToken** (longitud, caracteres permitidos, formato) antes de procesarlo internamente. [Guía del Desarrollador OWASP | Fundamentos de Seguridad | OWASP Foundation]
- **Implementar manejo de excepciones seguro:** capturar errores internos y retornar respuestas genéricas al cliente sin exponer detalles internos. [Guía del Desarrollador OWASP | Fundamentos de Seguridad | OWASP Foundation]
- **Registrar los errores internamente (logs)** para diagnóstico, pero no mostrar trazas al cliente. [Guía del Desarrollador OWASP | Fundamentos de Seguridad | OWASP Foundation]

Mitigación de inyección SQL a ciegas

Inyección SQL a ciegas

Description

Blind SQL injection is a technique that exploits a vulnerability occurring in the database of an application. This kind of vulnerability is harder to detect than basic SQL injections because no error message will be displayed on the webpage.

Vulnerability found in /web/guest/en/websys/webArch/login.cgi

Description | HTTP Request | cURL command line

```
Inyección SQL ciega mediante inyección en el parámetro userid_work
```

Solutions

To protect against SQL injection, user input must not directly be embedded in SQL statements. Instead, user input must be escaped or filtered or parameterized statements must be used.

References

- OWASP: Blind SQL Injection
- Wikipedia: SQL injection
- CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Para evitar este tipo de vulnerabilidad, se recomiendan las siguientes medidas:

- **Emplear consultas preparadas** (parameterized queries) o sentencias parametrizadas en lugar de concatenar directamente cadenas de usuario. [Guía del Desarrollador OWASP | Fundamentos de Seguridad | OWASP Foundation]
- **Validar y filtrar las entradas del usuario** (lista blanca, longitud máxima, caracteres permitidos) antes de pasarlas a la capa de datos. [Guía del Desarrollador OWASP | Fundamentos de Seguridad | OWASP Foundation]
- **Limitar el privilegio de acceso a la base de datos** (mínimos permisos necesarios) para que una explotación no decante en una toma total del sistema. [Riesgos de seguridad de aplicaciones web | Fluid Attacks]
- **Implementar mecanismos de detección de patrones sospechosos** en tiempos de respuesta (delay-based) e introducir límites de tasa (rate limiting) para dificultar la extracción por fuerza bruta. [Guía del Desarrollador OWASP | Fundamentos de Seguridad | OWASP Foundation]
- **Usar firewall de aplicación web** (WAF) para filtrar peticiones con payloads SQL comunes y bloquear solicitudes maliciosas antes de llegar al servidor. [Riesgos de seguridad de aplicaciones web | Fluid Attacks]

4.Resumen de las vulnerabilidades

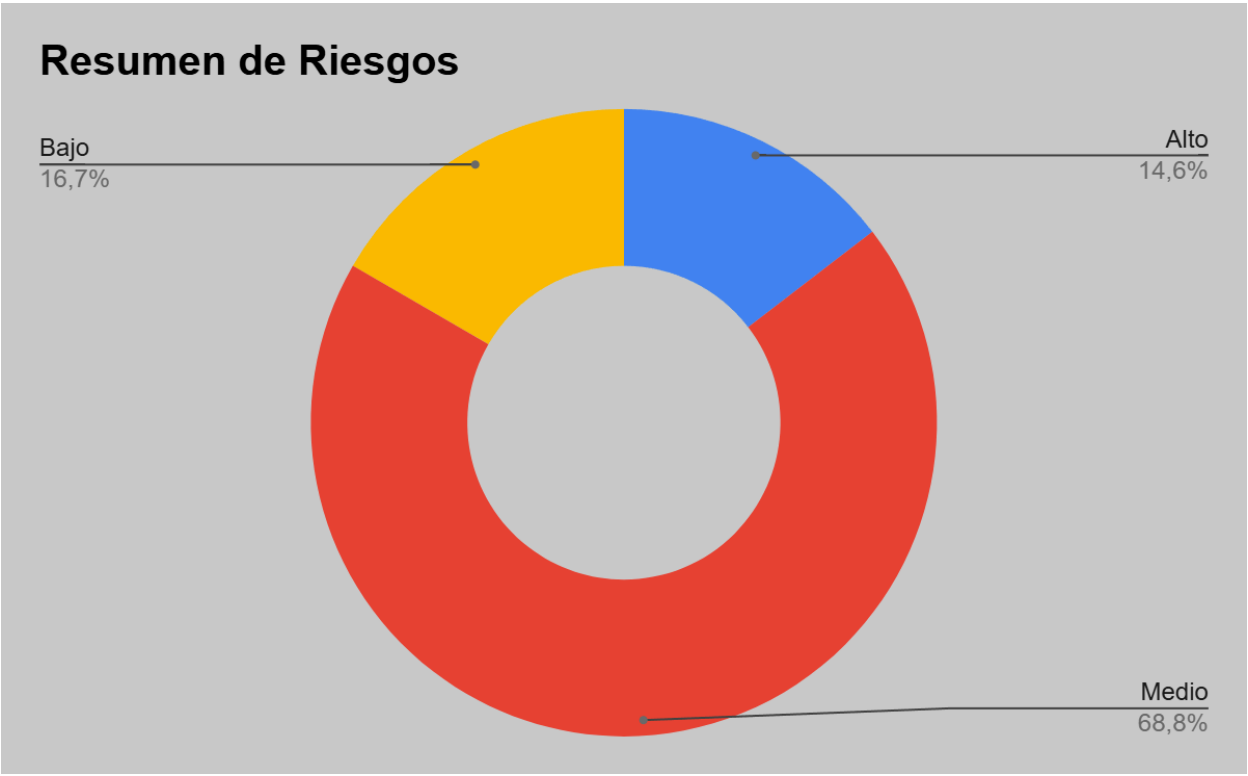
Tras analizar en detalle un amplio conjunto de vulnerabilidades detectadas en el sistema, a continuación se presenta un resumen concentrado de los principales riesgos identificados, agrupados por su nivel de criticidad y con énfasis en su impacto potencial.

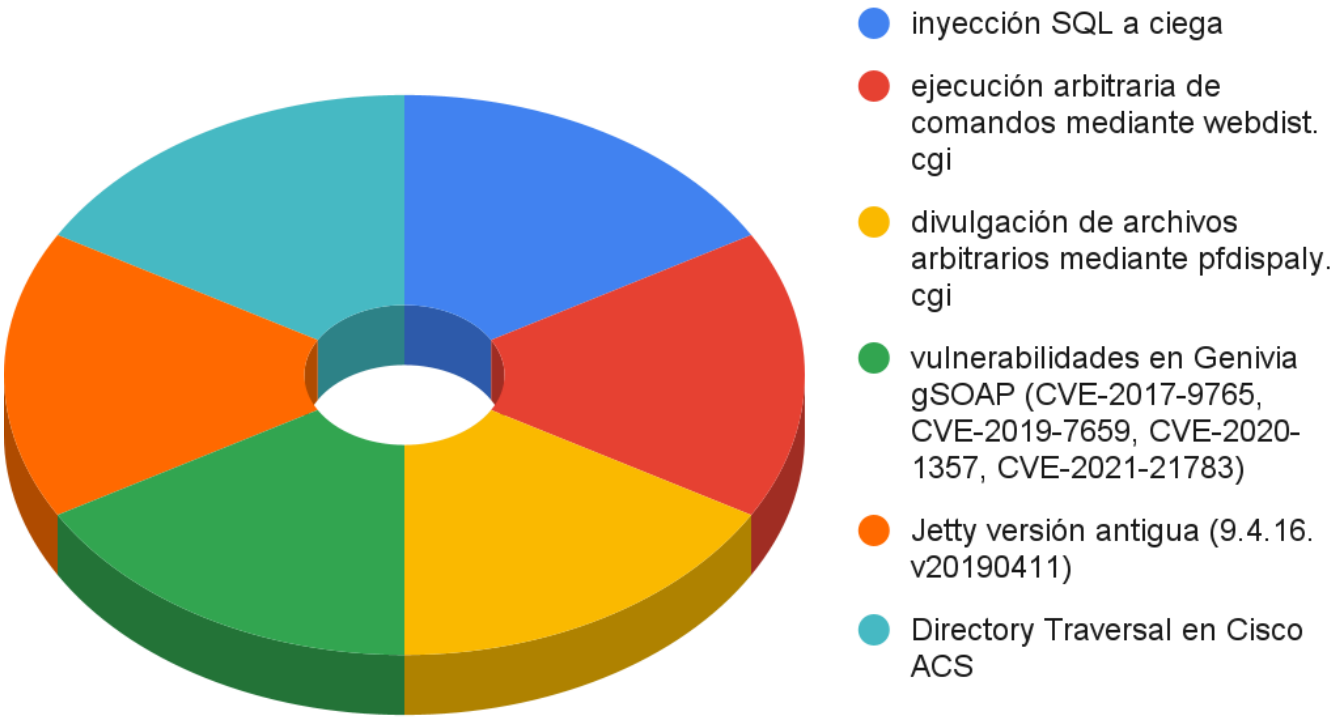
Resumen de amenazas

- **Alto riesgo**: vulnerabilidades que permiten ejecución de comandos remotos, control total, divulgación de archivos sensibles o explotación de servicios críticos (por ejemplo, inyección de comandos, SQL ciega, CGI vulnerables, gSOAP, versiones antiguas de Jetty, transversal de directorios).

- **Riesgo medio**: debilidades de configuración, protocolos o exposición informativa que pueden facilitar ataques (como ataques tipo Slowloris, vulnerabilidades SSL/SSLv3, LFI, redirecciones inseguras, CSRF, cabeceras de seguridad débiles o ausentes, cookies inseguras, exposición de versiones o IPs, consumo de recursos sin control, etc.).

- **Bajo riesgo**: fugas de información menores pero útiles para un atacante (como revelar la versión del servidor o framework, mostrar errores detallados, exponer IP privada, omitir cabeceras de seguridad como X-Frame-Options o X-Content-Type-Options).




inyección SQL a ciega
ejecución arbitraria de comandos mediante webdist.cgi
divulgación de archivos arbitrarios mediante pfdispaly.cgi
vulnerabilidades en Genivia gSOAP (CVE-2017-9765, CVE-2019-7659, CVE-2020-1357, CVE-2021-21783)
Jetty versión antigua (9.4.16.v20190411)
Directory Traversal en Cisco ACS

Riesgos Medios

Slowloris	POODLE (CVE-2014-3566)
LFI en phpMyAdmin grab_globals.lib.php (CVE-2005-3299)	desmb-vuln-cve2009-3103
Diffie-Hellman	desmb-vuln-cve2009-3103
Redirección Externa	HTTP verb tampering
Cabecera CSP ausente o débil	Ausencia de protección CSRF
Cookie sin el atributo Samesite	Exposición de listado de directorios
Errores internos (HTTP 500) inducidos por inyección	Ocultar la versión en la cabecera Server
Consumo de recursos sin control	Gran redirección detectada
Autenticación débil	Revelación de información por UPnP
Librería JS vulnerable	Credenciales capturadas
Devolver falsos positivos por método JUNK	Falta encabezado AnticlickJacking
Cabecera Strict-Transport-Security no definida para TLS	Vulnerabilidad de métodos permitidos
Error de que el nombre del host no coincide con el certificado (CWE-297)	Cookie sin HttpOnly/Secure
Exposición de la utilidad de configuración BIG-IP vía bigconf.cgi	

Riesgos bajos

Ocultar la versión en la cabecera Server	X-Powered-By (divulga información)
Falta encabezado X-Content-Type-Options	Mostrar errores de aplicación
Revelación de información del servidor	Revelación de IP privada
Falta configuración de la cabecera X-Frame-Options	

4. Contraseñas encontradas

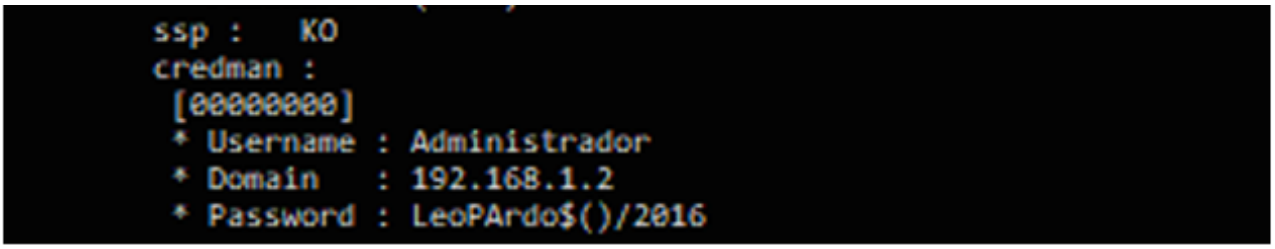


Figura 46: Salida de *sekurlsa::tickets /export* mostrando contraseña en memoria

Descripción del hallazgo: Se expuso una contraseña en texto claro asociada al host 192.168.1.2 dentro de la red local, permitiendo autenticación no autorizada y posible movimiento lateral.

Riesgo: Alto.

Para mitigar este tipo de vulnerabilidad, se sugieren las siguientes acciones:

- Rotar de inmediato la contraseña filtrada y cualquier otra donde se haya reutilizado.
- Restringir temporalmente el acceso al servicio (ACLs/firewall) hasta completar el hardening.
- Eliminar credenciales por defecto, aplicar política robusta de contraseñas y activar MFA cuando sea posible..
- Limitar el acceso administrativo a rangos/"jump hosts" de administración (mínimo privilegio).
- Deshabilitar servicios innecesarios, limitar intentos de login y activar bloqueo de cuentas.
- Actualizar firmware/paquetes del dispositivo y del servicio afectado.
- Comprobar cierre: que la credencial ya no funcione, que no haya contraseñas en claro en el tráfico, y que las nuevas reglas estén activas.

5. Dispositivos vulnerados

Impresoras y Escáneres

Brother HL-L5100DN

-Se hallaron las vulnerabilidades:

- Redirección externa
- Ausencia de token CSRF
- Cabecera CSP no configurada
- Exposición de listado de directorios
- Cookie sin el atributo Samesite
- Revelación de información del servidor
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Revelación de Ip privada
- Gran Redirección detectada

Brother HL-L5210-DN

Se hallaron las vulnerabilidades:

- Slowloris
- Ausencia de token CSRF
- Cabecera CSP no configurada
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Métodos Permitidos

Brother HL-L5200DW

Se hallaron las vulnerabilidades:

- Ausencia de token CSRF
- Cabecera CSP no configurada
- Falta de encabezado X-CONTENT-TYPE-OPTIONS

Brother HL-L5100DN (Modelo 2)

Se hallaron las vulnerabilidades:

- Revelación de Ip privada
- Gran Redirección detectada
- Cabecera CSP no configurada
- Falta de encabezado X-CONTENT-TYPE-OPTIONS

Ricoh MP C401SR (Modelo 2)

Se hallaron las vulnerabilidades:

- Poodle
- Desmb-vuln-cve2009-3103
- Inyección Remota de Comandos del Sistema Operativo
- Cabecera CSP no configurada
- Falta de encabezado Anticlickjacking
- Librería JS Vulnerable
- Revelación de información del servidor
- Mostrar errores de aplicación
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Cookie establecida sin el atributo HttpOnly y Secure
- Inyección Sql a ciegas
- Errores internos (HTTP 500) inducidos por inyección

Ricoh MP C401SR

Se hallaron las vulnerabilidades:

- Poodle
- Inyección Remota de Comandos del Sistema Operativo
- Cabecera CSP no configurada
- Falta de encabezado Anticlickjacking
- Librería JS Vulnerable
- Revelación de información del servidor
- Mostrar errores de aplicación
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Cookie establecida sin el atributo HttpOnly y Secure
- Inyección Sql a ciegas
- Errores internos (HTTP 500) inducidos por inyección

Escáner PFU

Se hallaron las vulnerabilidades:

- Slowloris
- Cabecera CSP no configurada
- Falta de encabezado Anticlickjacking
- Métodos Permitidos

Escáner PFU (Modelo 2)

Se hallaron las vulnerabilidades:

- Slowloris
- Cabecera CSP no configurada
- Exposición de listado de directorios
- Cookie sin el atributo Samesite
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Revelación de Ip privada

HPLaserJet M404DN

Se hallaron las vulnerabilidades:

- Slowloris
- Genivia SOAP
- LFI en phpMyAdmin grab_globals.lib.php
- Credenciales capturadas
- Cabecera CSP no configurada
- Ausencia de token CSRF
- Autenticación débil
- Librería JS Vulnerable
- Revelación de Ip privada
- Consumo de recursos sin control

Cámaras

Cámara GrandStream Network

Se hallaron las vulnerabilidades:

- Slowloris
- Cabecera CSP no configurada
- Falta de encabezado Anticlickjacking
- Revelación de información del servidor
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Revelación de Ip privada

Dispositivos de red

TpLink

Se hallaron las vulnerabilidades:

- Desmb-vuln-cve2009-3103

Millenial Net

Se hallaron las vulnerabilidades:

- Slowloris
- Cabecera CSP no configurada
- Falta de encabezado Anticlickjacking
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Librería JS Vulnerable
- Directory Traversal en Cisco ACS
- Ejecución arbitraria de comandos mediante webdist.cgi
- Exposición de la utilidad de configuración BIG-IP vía bigconf.cgi
- Cabecera Access-Control-Allow-Origin con wildcard y posible XSS
- Devolver falsos positivos por medio de método JUNK
- Uso de cabecera Content-Type: text/plain con contenido inseguro

Ordenadores

Hewlett Packard y HonHai Precision

No se hallaron las vulnerabilidades.

Hewlett Packard iLO

Se hallaron las vulnerabilidades:

- Cabecera CSP no configurada
- Mostrar errores de aplicación
- Cookie sin el atributo Samesite
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Librería JS Vulnerable
- Strict-Transport-Security (HSTS) no esté definida para TLS
- Revelación de Ip privada

Servicios IIS

IIS Asustek Computer, IIS Dell, IIS Hewlett Packard (Modelo 2) y IIS Hewlett Packard (Modelo 3)

No se hallaron las vulnerabilidades.

IIS Dell (Modelo 2)

Se hallaron las vulnerabilidades:

- Desmb-vuln-cve2009-3103
- Falta de encabezado Anticlickjacking
- Falta de encabezado X-CONTENT-TYPE-OPTIONS

IIS Hewlett Packard

Se hallaron las vulnerabilidades:

- Desmb-vuln-cve2009-3103
- Revelación de información del servidor
- Falta de encabezado Anticlickjacking
- Falta de encabezado X-CONTENT-TYPE-OPTIONS

IIS Hewlett Packard (Modelo 4)

Se hallaron las vulnerabilidades:

- Slowloris
- HTTP verb tampering
- Cabecera CSP no configurada
- Credenciales capturadas
- Falta de encabezado Anticlickjacking
- Autenticación débil
- Revelación de información del servidor
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Métodos permitidos

IIS Hewlett Packard (Modelo 5)

Se hallaron las vulnerabilidades:

- Cabecera CSP no configurada
- Falta de encabezado Anticlickjacking
- Autenticación débil
- Revelación de información del servidor
- Falta de encabezado X-CONTENT-TYPE-OPTIONS

Microsoft SQL Server (Base de Datos)

Se hallaron las vulnerabilidades:

- Poodle
- Diffie-Hellman

Servidor Hewlett Packard

Se hallaron las vulnerabilidades:

- Credenciales capturadas
- Cabecera CSP no configurada
- Falta de encabezado Anticlickjacking
- Ausencia de token CSRF
- Autenticación débil
- Librería JS Vulnerable
- divulga información mediante un campo(s) de encabezado de respuesta HTTP X-Powered-By
- Revelación de Ip privada
- Strict-Transport-Security (HSTS) no esté definida para TLS
- Métodos permitidos
- Error de que el nombre del host no coincide con los nombres del certificado (CWE-297: Validación incorrecta de certificado con desajuste de host)

6.Página Web

Se hallaron las vulnerabilidades:

- Inyección SQL a ciegas
- Ausencia de token CSRF
- Cabecera CSP no configurada
- Falta de encabezado Anticlickjacking
- Strict-Transport-Security (HSTS) no esté definida para TLS
- Falta de encabezado X-CONTENT-TYPE-OPTIONS
- Métodos permitidos

Se hallaron las vulnerabilidades de tipo wordpress en la página web, que son las siguientes:

Elementor<3.24.0—Authenticated(Contributor+)StoredXSSenparámetro URL(CVE-2024-5416)

Descripción: Un usuario con rol *contributor* puede inyectar contenido malicioso persistente (Cross-Site Scripting almacenado) a través de un parámetro URL en determinados widgets.

Impacto: ejecución de JavaScript en el navegador de administradores u otros usuarios que visualicen la página; robo de cookies/sesiones, redirecciones maliciosas, defacement.

Mitigación:

- Actualizar Elementor a versión ≥ 3.24.0.
- Revisar y limitar privilegios del rol *contributor* (evitar que editen widgets que renderizan HTML sin filtrado).
- Sanear y escapar todas las entradas que se muestren en vistas públicas o de administración.
- Auditar entradas recientes creadas por usuarios *contributor*.

Elementor 3.24.6 — Exposición de información (CVE-2024-6757)

Descripción: La función encargada de obtener el texto alternativo de imágenes podía exponer metadatos, rutas u otros datos sensibles.

Impacto: divulgación de información útil para la enumeración y la planificación de ataques posteriores.

Mitigación:

- Actualizar a versión ≥ 3.24.6.
- Revisar la información mostrada en metadatos de imágenes y limitar los datos expuestos.
- Aplicar control de acceso donde proceda y sanitizar las salidas.

Elementor < 3.25.8 — Contributor+ Stored XSS (CVE-2024-8236)

Descripción: Vulnerabilidad de XSS persistente explotable por usuarios con rol *contributor*.

Impacto: ejecución de scripts en contexto de usuarios que visualicen el contenido afectado.

Mitigación:

- Actualizar a versión ≥ 3.25.8.
- Validar y sanear las entradas de usuario en los widgets/configuraciones susceptibles.
- Auditar contenido previamente creado por roles contribuyentes.

Elementor<3.25.10—Contributor+Stored XSS vía Typography Settings(CVE 2024-10453)

Descripción: En los ajustes tipográficos permitía introducir código malicioso que se almacenaba y luego se ejecutaba al renderizar la página.

Impacto: XSS persistente mediante configuraciones visuales; riesgo para usuarios y administradores.

Mitigación:

- Actualizar a versión ≥ 3.25.10.
- Validar y filtrar los valores recibidos en las configuraciones tipográficas.
- Auditar ajustes tipográficos ya existentes para detectar posibles payloads.

Elementor < 3.27.5 — Contributor+ Stored XSS (CVE-2024-13445)

Descripción: Nueva instancia de XSS persistente explotable por roles *contributor* en widgets o parámetros internos.

Impacto: ejecución de scripts cuando usuarios visualicen contenido comprometido.

Mitigación:

- Actualizar a versión ≥ 3.27.5.
- Aplicar saneamiento y escape de todas las entradas.
- Auditar contenido generado por roles de bajo privilegio.

Elementor < 3.25.11 — Contributor+ Stored XSS (CVE-2024-54444)

Descripción: Variante adicional de XSS persistente corregida en la versión indicada.

Impacto: ejecución de scripts maliciosos al mostrar contenido afectado.

Mitigación:

- Actualizar a versión ≥ 3.25.11.
- Sanear y escapar todas las entradas de usuario.
- Auditar contenido existente para detectar inyecciones previas.

Elementor<3.29.1—Contributor+StoredXSS(CVE-2024-50555 y CVE-2025 3075)

Descripción: Dos vulnerabilidades tipo XSS persistente corregidas en la versión 3.29.1, explotables por usuarios con rol *contributor*.

Impacto: ejecución de scripts cuando se visualiza contenido infectado; riesgos de manipulación de sesión, redirecciones maliciosas, defacement.

Mitigación:

- Actualizar Elementor a versión ≥ 3.29.1.
- Auditar contenido aportado por usuarios *contributor*.
- Garantizar saneamiento y escape adecuados para todas las rutas de entrada.

Elementor < 3.30.3 — Contributor+ Stored XSS vía Text Path Widget (CVE 2025-4566)

Descripción: El widget *Text Path* permitía la inyección y almacenamiento de payloads XSS, los cuales se ejecutaban al renderizar la página.
Impacto: ejecución de código malicioso en el contexto de los usuarios que visitan páginas afectadas.
Mitigación:

- Actualizar a versión ≥ 3.30.3.
- Restringir el uso del widget *Text Path* si no es imprescindible.
- Validar y sanear todos los parámetros del widget antes de su renderización.
- Auditar páginas que usen este widget para detectar inyecciones anteriores.

Elementor < 3.30.3 — Admin+ Arbitrary File Read vía Image Import (CVE 2025-8081)

Descripción: La funcionalidad de importación de imágenes podía ser usada por un actor con privilegios administrativos para leer ficheros arbitrarios del servidor (por ejemplo `.env`, backups o archivos de configuración).
Impacto: divulgación de archivos sensibles del servidor, lo que puede derivar en el compromiso completo del sitio o del servidor.
Mitigación:

- Actualizar a versión ≥ 3.30.3.
- Revisar permisos de ficheros y del usuario que ejecuta el proceso web, limitando lecturas solo a rutas seguras.
- Evitar otorgar privilegios de admin a cuentas no necesarias.
- Auditar logs de importación de imágenes y accesos a rutas sensibles.

Ocean Extra < 2.4.7 — Contributor+ Stored XSS vía Shortcode (CVE-2025 3457)

Descripción: Shortcodes gestionados por el plugin permitían que un usuario con rol *contributor* guardara contenido malicioso que luego se interpretaba como script al visualizar la página.

Impacto: XSS persistente al renderizar páginas que usan esos shortcodes; riesgo de robo de sesión, relevancia para defacement u otras manipulaciones.

Mitigación:

- Actualizar Ocean Extra a versión ≥ 2.4.7.
- Deshabilitar shortcodes inseguros o no utilizados.
- Validar y sanear todos los parámetros aceptados por los shortcodes.
- Auditar contenido existente que use esos shortcodes para detectar inyecciones previas.

Ocean Extra < 2.4.7 — Contributor+ Stored XSS vía ocean_gallery_id (CVE 2025-3458)

Descripción: El parámetro `ocean_gallery_id` no validaba ni escapaba correctamente su contenido, permitiendo la inyección persistente de código malicioso.

Impacto: ejecución de scripts al renderizar la página que use ese parámetro.

Mitigación:

- Actualizar a versión ≥ 2.4.7.
- Sanitizar los valores recibidos para `ocean_gallery_id`.
- Revisar contenido existente que use dicho parámetro para detectar inyecciones.

Ocean Extra < 2.4.7 — Ejecución arbitraria de shortcodes sin autenticación (CVE-2025-3472)

Descripción: Permite la ejecución de shortcodes arbitrarios sin autenticación previa: un atacante remoto podría invocar funciones del plugin usando parámetros públicos que no se validan adecuadamente.
Impacto: ejecución no autorizada de funcionalidades del plugin, con posibilidad de divulgación, modificación o ejecución de operaciones no previstas.
Mitigación:

- Actualizar a versión ≥ 2.4.7.
- Si no hay parche disponible, desactivar Ocean Extra hasta aplicar la corrección.
- Revisar todas las páginas que usen shortcodes y eliminar o restringir los que no sean imprescindibles.
- Implementar controles de validación antes de ejecutar do-shortcode o funciones similares.

Ocean Extra < 2.4.9 — Authenticated (Contributor+) Stored XSS (CVE-2025 49068)

Descripción: Variante de XSS persistente explotable por usuarios con rol *contributor*, corregida en la versión 2.4.9.
Impacto: ejecución de scripts maliciosos al visualizar contenido afectado.
Mitigación:

- Actualizar a versión ≥ 2.4.9.
- Validar y sanear todas las entradas de usuario asociadas.
- Auditar contenido existente para detectar inyecciones.

Ocean Extra < 2.5.0 — Contributor+ Stored XSS (CVE-2025-9499)

Descripción: Vulnerabilidad de XSS persistente presente en versiones anteriores a 2.5.0, explotable por usuarios con rol *contributor*.
Impacto: ejecución de scripts maliciosos al mostrar contenido afectado, con riesgos de robo de sesión, manipulación del DOM y otros efectos.
Mitigación:

- Actualizar a versión ≥ 2.5.0.
- Auditar las entradas o shortcodes usados por usuarios para detectar inyecciones.
- Validar y sanear todos los parámetros del plugin que puedan admitir inyección.

La siguiente tabla resume las vulnerabilidades detectadas en los plugins **Elementor** y **Ocean Extra** (WordPress), clasificadas por **criticidad** y ordenadas de mayor a menor riesgo. La priorización combina la **severidad técnica** (p. ej., XSS persistente frente a simple divulgación), el **rol mínimo requerido** para explotar (no autenticado, contributor, admin), la **exposición** (si afecta a vistas públicas) y el **impacto operativo** (robo de sesión, ejecución de JavaScript, fuga de datos o lectura de ficheros). Interpreta así las columnas: *Versiones afectadas* delimita el rango vulnerable; *Vulnerabilidad* describe el vector; *Rol requerido* indica la barrera de entrada; *Impacto* sintetiza el efecto práctico. Los ítems **Alto** deben abordarse de forma inmediata (parcheo, mitigaciones temporales); los **Medio** pueden planificarse en el siguiente ciclo, sin perder seguimiento.

Riesgo	Plugin	Versiones afectadas	Vulnerabilidad	Rol requerido	Impacto (muy resumido)
Alto	Ocean Extra	< 2.4.7	Ejecución de *shortcodes* sin autenticación	No autenticado	Lógica del sitio expuesta; posible XSS/fuga de datos.
Alto	Elementor	< 3.30.3	Stored XSS en Text Path Widget	Contributor+	JS persistente; robo de sesión y acciones como admin.
Alto	Elementor	< 3.25.10	Stored XSS en Typography Settings	Contributor+	JS persistente; secuestro de sesión/defacement.
Alto	Elementor	< 3.24.0	Stored XSS en parámetro de URL	Contributor+	JS persistente desde URL; toma de cuenta.
Alto	Elementor	< 3.25.8	Stored XSS autenticado	Contributor+	JS persistente; robo de sesión y movimientos laterales.
Alto	Elementor	< 3.27.5	Stored XSS autenticado	Contributor+	JS persistente; acciones no autorizadas.
Alto	Elementor	3.29.1	Stored XSS (múltiples rutas)	Contributor+	Varios vectores de JS persistente; elevación de privilegios.
Alto	Ocean Extra	< 2.4.7	Stored XSS vía `ocean_gallery_id`	Contributor+	JS persistente al mostrar galerías.
Alto	Ocean Extra	< 2.4.7	Stored XSS vía *shortcode*	Contributor+	JS persistente inyectado por *shortcode*.
Alto	Ocean Extra	< 2.4.9	Stored XSS autenticado	Contributor+	JS persistente; robo de sesión.
Alto	Ocean Extra	< 2.5.0	Stored XSS autenticado	Contributor+	JS persistente; redirecciones y exfiltración.
Medio	Elementor	3.24.6	Divulgación de información	—	Fuga de rutas/metadatos; facilita otros ataques.

Medio	Elementor	< 3.30.3	Lectura arbitraria de archivos (Image Import)	Admin+	Exposición de ficheros sensibles; requiere admin.

7.Conclusión general de la auditoría de pentesting

La auditoría de seguridad realizada ha permitido identificar un amplio conjunto de vulnerabilidades distribuidas entre los distintos dispositivos, servicios y aplicaciones de la organización. Los hallazgos abarcan desde configuraciones débiles y cabeceras HTTP ausentes hasta vulnerabilidades críticas que podrían comprometer la confidencialidad, integridad y disponibilidad de los sistemas.

De forma general, las **principales áreas de atención prioritaria** son las siguientes:

1. **Vulnerabilidades críticas de ejecución remota y exposición de credenciales.**
   Se detectaron casos de *inyección de comandos del sistema operativo*, *inyecciones SQL a ciegas* y *credenciales expuestas en texto claro (192.168.1.2)*. Estas vulnerabilidades suponen un riesgo directo de **toma de control total del sistema** y deben ser tratadas de manera inmediata mediante:
   - Actualización y parcheo urgente de los servicios afectados.
   - Eliminación o rotación de contraseñas comprometidas.
   - Implementación de autenticación robusta (HTTPS, MFA, tokens seguros).
2. **Fortalecimiento de las configuraciones de seguridad en dispositivos y servidores.**
   La mayoría de impresoras, cámaras y servidores presentan **cabeceras CSP, HSTS, X-Frame-Options o X-Content-Type-Options ausentes**, así como configuraciones inseguras de cookies. Estas deficiencias facilitan ataques como XSS, CSRF o clickjacking.
   Se recomienda establecer una política homogénea de endurecimiento (hardening) aplicando las cabeceras de seguridad, restringiendo métodos HTTP y deshabilitando servicios o scripts obsoletos.
3. **Actualización de software y bibliotecas vulnerables.**
   Se identificaron versiones desactualizadas de **OpenSSH, phpMyAdmin, gSOAP, jQuery y plugins de WordPress (Elementor y Ocean Extra)** con múltiples CVE conocidos.
   La prioridad debe centrarse en **mantener un proceso continuo de gestión de parches** (tanto del núcleo como de los componentes complementarios) y en **verificar que las versiones desplegadas son seguras**.
4. **Revisión de la arquitectura web y control de cabeceras.**
   El análisis web muestra vulnerabilidades recurrentes por ausencia de validación de entradas, cabeceras mal configuradas, uso inseguro de redirecciones y exposición de información sensible (servidor, IP privada, versiones).
   Es imprescindible **definir una política unificada de cabeceras HTTP de seguridad**, implementar validaciones del lado del servidor y utilizar *WAF* o *reverse proxies* para filtrar tráfico sospechoso.
5. **Gestión de la red y segmentación de servicios.**
   Algunos servicios (como SMBv2 o UPnP) exponen información que puede ser explotada lateralmente. Se aconseja **aislar dispositivos en VLAN específicas**, deshabilitar protocolos innecesarios y restringir los accesos administrativos mediante ACLs o firewalls.

Aspectos secundarios (prioridad media/baja)

- **Fugas informativas menores** (versiones en cabecera Server, X-Powered-By, errores 500 o mensajes de depuración) no representan un riesgo inmediato, pero deben corregirse progresivamente.
- **Cabeceras no críticas** (por ejemplo, SameSite, nosniff o ServerTokens) pueden implementarse en una segunda fase, tras abordar las vulnerabilidades de ejecución y autenticación.
- **Pruebas periódicas de revalidación (retesting)** deben programarse tras aplicar mitigaciones, para garantizar que no se reintroducen fallos durante las actualizaciones.

Síntesis final

La infraestructura auditada presenta un perfil de riesgo **medio-alto**, principalmente por la coexistencia de **vulnerabilidades críticas explotables** y **una base común de configuraciones débiles**.
El esfuerzo de remediación debe priorizar:

1. Eliminación de vulnerabilidades críticas (inyección, credenciales expuestas, ejecución remota).
2. Endurecimiento general de la configuración de servicios y cabeceras HTTP.
3. Actualización y mantenimiento continuo del software y librerías.

Una vez estabilizado el entorno, las fases siguientes deben centrarse en **automatizar el control de versiones, la gestión de parches y la monitorización continua de seguridad**, permitiendo mantener la postura defensiva de la organización frente a futuras amenazas.

POR EL CLIENTE	**PENTESTINGS SÁNCHEZ-ROSSO ALMOGUERA**
Fdo:Jose Francisco Roldán Fecha: 25 de noviembre de 2025	Fdo:Jose Sánchez-Rosso Almoguera Fecha: 25 de noviembre de 2025

Bibliografía

[1] Innovación Digital 360. *Nikto: escaneo de vulnerabilidades empresariales*. https://www.innovaciondigital360.com/cyber-security/nikto-el-escaner-de-vulnerabilidades-para-aplicaciones-web-asi-funciona/. Consultado el 20 de mayo de 2025.

[2] Aircrack-ng. *es:airodump-ng*. https://www.aircrack-ng.org/doku.php?id=es:airodump-ng. Consultado el 20 de mayo de 2025.

[3] Aircrack-ng. *es:newbie_guide*. https://www.aircrack-ng.org/doku.php?id=es:newbie_guide. Consultado el 20 de mayo de 2025.

[4] Aircrack-ng. *es:newbie_guide*. https://www.aircrack-ng.org/doku.php?id=es:aireplay-ng. Consultado el 20 de mayo de 2025.

[5] Akamai. *Modelo de seguridad Zero Trust*. Consultado el 8 de mayo de 2025. URL: https://www.akamai.com/es/glossary/what-is-zero-trust.

[6] *Alerta "Cookie sin atributo SameSite" - ZAP*. https://www.zaproxy.org/docs/alerts/10054/. Consultado el 24 de junio de 2025.

[7] *Alerta Seguro por Diseño: eliminar vulnerabilidades de inyección de comandos del sistema operativo (CISA / FBI)*. Consultado el 7 de julio de 2025. CISA. URL: https://www.cisa.gov/resources-tools/resources/secure-design-alert-eliminating-os-command-injection-vulnerabilities.

[8] Chema Alonso. *Lynis: Auditar y fortificar servidores GNU/Linux*. https://www.elladodelmal.com/2016/04/lynis-auditar-y-fortificar-servidores.html. Consultado el 20 de mayo de 2025.

[9] Chema Alonso. *Pentesting en Active Directory: Pass-the-ticket & Mimikatz*. Consultado el 19 de mayo de 2025. URL: https://www.elladodelmal.com/2021/03/pentesting-en-active-directory-pass.html.

[10] Amazon Web Services. *¿Qué es DevSecOps?* Consultado el 7 de mayo de 2025. URL: https://aws.amazon.com/es/what-is/devsecops/.

[11] Amazon Web Services. *¿Qué es el Aprendizaje mediante refuerzo?* Consultado el 8 de mayo de 2025. URL: https://aws.amazon.com/es/what-is/reinforcement-learning/.

[12] Amazon Web Services. *Guía de respuestas ante incidentes de seguridad de AWS*. Consultado el 8 de mayo de 2025. URL: https://docs.aws.amazon.com/es_es/whitepapers/latest/aws-security-incident-response-guide/aws-security-incident-response-guide.pdf.

[13] AppCheck. *Automated Penetration Testing*. Consultado el 7 de mayo de 2025. URL: https://appcheck-ng.com/automated-penetration-testing/.

[14] Fluid Attacks. *Cumplimiento que pide pentesting*. Consultado el 20 de mayo de 2025. URL: https://fluidattacks.com/es/blog/cumplimiento-pruebas-de-penetracion.

[15] Fluid Attacks. *Riesgos de seguridad para aplicaciones web: ataques y contramedidas*. https://fluidattacks.com/es/blog/riesgos-seguridad-aplicaciones-web. Consultado el 25 de septiembre de 2025.

[16] *Autenticación básica sobre HTTP - vulnerabilidad (Acunetix)*. Consultado el 3 de julio de 2025. Acunetix. URL: https://www.acunetix.com/vulnerabilities/web/basic-authentication-over-http/.

[17] Rahul Awati. *¿Qué es la plataforma de escaneo de vulnerabilidades Nessus?* Consultado el 2 de mayo de 2025. URL: https://www-techtarget-com.translate.goog/searchnetworking/definition/Nessus?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq.

[18] PixelFreeStudio Blog. *Buenas prácticas CORS: evitar usar comodines (*) en producción*. Consultado el 17 de agosto de 2025. URL: https://blog.pixelfreestudio.com/cors-errors-demystified-how-to-fix-cross-origin-issues-2/.

[19] Qualys Blog. *La importancia de una correcta implementación de HTTP Strict Transport Security (HSTS)*. Consultado el 16 de agosto de 2025. URL: https://blog.qualys.com/vulnerabilities-threat-research/2016/03/28/the-importance-of-a-proper-http-strict-transport-security-implementation-on-your-web-server.

[20] Campus Ciberseguridad. *Burp Suite una herramienta de pentesting*. Consultado el 16 de mayo de 2025. URL: https://www.campusciberseguridad.com/blog/burp-suite-una-herramienta-de-pentesting.

[21] Check Point Research. *Campaña de phishing renovada de APT29 contra diplomáticos europeos*. Consultado el 15 de mayo de 2025. URL: https://research.checkpoint.com/2025/apt29-phishing-campaign/.

[22] Check Point Research. *El malware más buscado en octubre de 2024: aumento de infostealers como Lumma Stealer*. Consultado el 15 de mayo de 2025. URL: https://blog.checkpoint.com/security/october-2024s-most-wanted-malware-infostealers-surge-as-cyber-criminals-leverage-innovative-attack-vectors/.

[23] Check Point Research. *El malware móvil más buscado en noviembre de 2024: Androxgh0st lidera el grupo*. Consultado el 15 de mayo de 2025. URL: https://blog.checkpoint.com/research/november-2024s-most-wanted-malware-androxgh0st-leads-the-pack-targeting-iot-devices-and-critical-infrastructure/.

[24] Check Point Research. *Informe de Seguridad de IA 2025: Comprendiendo las amenazas y construyendo defensas más inteligentes*. Consultado el 15 de mayo de 2025. URL: https://blog.checkpoint.com/research/ai-security-report-2025-understanding-threats-and-building-smarter-defenses/.

[25] Check Point Software Technologies. *Informe de Ciberseguridad 2025*. Consultado el 15 de mayo de 2025. URL: https://www.checkpoint.com/security-report/.

[26] Check Point Software Technologies. *Mapa de amenazas cibernéticas en vivo*. Consultado el 16 de mayo de 2025. URL: https://threatmap.checkpoint.com.

[27] CISA. *SSL 3.0 Protocolo Vulnerabilidad y POODLE Attack*. https://www.cisa.gov/news-events/alerts/2014/10/17/ssl-30-protocol-vulnerability-and-poodle-attack. Consultado el 15 de julio de 2025.

[28] *CISA incluye vulnerabilidad jQuery (XSS) en su catálogo de vulnerabilidades explotadas*. Consultado el 5 de julio de 2025. The Hacker News. URL: https://thehackernews.com/2025/01/cisa-adds-five-year-old-jquery-xss-flaw.html.

[29] Inc. Cisco Systems. *Vulnerabilidad de lectura arbitraria de archivos en Cisco Secure ACS vía RMI*. Consultado el 21 de agosto de 2025. 2014. URL: https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/Cisco-SA-20140116-CVE-2014-0667.

[30] CISOfy. *Does Lynis need root permissions?* https://cisofy.com/faq/does-lynis-need-root-permissions/. Consultado el 20 de mayo de 2025.

[31] *Clickjacking - MDN Web Docs*. Consultado el 15 de julio de 2025. MDN Web Docs. URL: https://developer.mozilla.org/docs/Web/Security/Attacks/Clickjacking.

[32] Cloudflare. *¿Qué es la seguridad del IoT?* Consultado el 7 de mayo de 2025. URL: https://www.cloudflare.com/es-es/learning/security/glossary/iot-security/.

[33] *Configurar tu servidor para no revelar su identidad - Acunetix*. Consultado el 25 de junio de 2025. Acunetix. URL: https://www.acunetix.com/blog/articles/configure-web-server-disclose-identity/.

[34] *Cookie sin atributo SameSite detectado - Tenable*. https://www.tenable.com/plugins/was/115540. Consultado el 24 de junio de 2025.

[35] *CVE-2023-38408 — Vulnerabilidad en ssh-agent de OpenSSH*. Consultado el 8 de octubre de 2025.

[36] *CWE-1275: Cookie sensible con atributo SameSite inapropiado*. https://cwe.mitre.org/data/definitions/1275.html. Consultado el 24 de junio de 2025.

[37] *CWE-1287: Validación inapropiada del tipo de entrada especificado*. https://cwe.mitre.org/data/definitions/1287.html. Consultado el 20 de junio de 2025.

[38] *CWE-352: Falsificación de solicitud entre sitios (CSRF)*. https://cwe.mitre.org/data/definitions/352.html. Consultado el 20 de junio de 2025.

[39] *CWE-601: Redirección de URL no confiable (Redirección abierta)*. https://cwe.mitre.org/data/definitions/601.html. Consultado el 20 de junio de 2025.

[40] Snyk Vulnerability Database. *Exposición de información en Jetty — actualización recomendada*. Consultado el 20 de agosto de 2025. URL: https://security.snyk.io/package/maven/org.eclipse.jetty:jetty-http/9.4.44.v20210927.

[41] *Defensa contra Clickjacking - hoja de trucos OWASP*. Consultado el 15 de julio de 2025. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html.

[42] *Deshabilitando el listado de directorios en tu servidor web - Acunetix*. https://www.acunetix.com/blog/articles/disabling-directory-listing-web-server/. Consultado el 24 de junio de 2025.

[43] DNSimple. *Qué es SSL SAN: uso de Subject Alternative Names frente a Common Name*. Consultado el 18 de agosto de 2025. URL: https://support.dnsimple.com/articles/what-is-ssl-san/.

[44] MDN Web Docs. *Configuración segura de cookies: atributos Secure, HttpOnly, prefijos, SameSite*. Consultado el 15 de agosto de 2025. URL: https://developer.mozilla.org/en-US/docs/Web/Security/Practical_implementation_guides/Cookies.

[45] MDN Web Docs. *Guía de política de seguridad de contenido (CSP) - MDN Web Docs*. https://developer.mozilla.org/docs/Web/HTTP/Guides/CSP. Consultado el 24 de junio de 2025.

[46] DragonJAR. *Wapiti: Escáner de Vulnerabilidades en Aplicaciones Web*. https://www.dragonjar.org/wapiti-escaner-vulnerabilidades-aplicaciones-web.xhtml. Consultado el 20 de mayo de 2025.

[47] Elderecho.com. *Pruebas de intrusión o pentesting, clave en el cumplimiento de normativas como GDPR, PCI-DSS, ENS, NIS2 e ISO*. Consultado el 20 de mayo de 2025. URL: https://elderecho.com/pruebas-de-intrusion-o-pentesting-clave-en-el-cumplimiento-de-normativas-como-gdpr-pci-dss-ens-nis2-e-iso-.

[48] *Encabezado X-Content-Type-Options faltante - alerta ZAP*. Consultado el 24 de junio de 2025. OWASP ZAP. URL: https://www.zaproxy.org/docs/alerts/10021/.

[49] *Encabezados de seguridad HTTP - Invicti*. Consultado el 25 de junio de 2025. Invicti. URL: https://www.invicti.com/blog/web-security/http-security-headers/.

[50] ESED. *¿Qué normativas de ciberseguridad debe cumplir tu empresa?* Consultado el 20 de mayo de 2025. URL: https://www.esedsl.com/blog/que-normativas-de-ciberseguridad-debe-cumplir-tu-empresa.

[51] EsGeeks. *Guía de 3 hacks para Windows con el uso de Mimikatz*. Consultado el 19 de mayo de 2025. URL: https://esgeeks.com/mimikatz-guia-uso/.

[52] EsGeeks. *LEGION: Herramienta de Enumeración Automática*. https://esgeeks.com/legion-herramienta-enumeracion-automatica/. Consultado el 20 de mayo de 2025.

[53] Gobierno de España. *Real Decreto 311/2022, de 3 de mayo, por el que se regula el Esquema Nacional de Seguridad*. https://www.boe.es/buscar/act.php?id=BOE-A-2022-7191. Publicado en el Boletín Oficial del Estado, núm. 106, de 4 de mayo de 2022. Consultado el 20 de mayo de 2025.

[54] OWASP (versión en español). *Guía del desarrollador – Fundamentos de Seguridad*. https://owasp.org/www-project-developer-guide/release-es/fundamentos/fundamentos_seguridad/. Consultado el 25 de septiembre de 2025.

[55] EtherGroup. *Nikto, escáner de vulnerabilidades para aplicaciones web*. https://blog.ethergroup.mx/posts/Nikto/. Consultado el 20 de mayo de 2025.

[56] Dan Farmer and Wietse Venema. *Security Administrator Tool for Analyzing Networks (SATAN)*. Consultado el 2 de mayo de 2025. URL: https://es.wikipedia.org/wiki/Security_Administrator_Tool_for_Analyzing_Networks (visited on 05/20/2025).

[57] Firma-e. *Test de Intrusión Externo e Interno*. Consultado el 1 de mayo de 2025. URL: https://www.firma-e.com/test-de-intrusion/.

[58] GlobalSign. *Qué es un error de desajuste de nombre común (Common Name Mismatch) y cómo solucionarlo*. Consultado el 18 de agosto de 2025. URL: https://www.globalsign.com/es/blog/what-is-common-name-mismatch-error.

[59] *Hoja de trucos de defensa contra la inyección de comandos del sistema operativo - OWASP*. Consultado el 7 de julio de 2025. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html.

[60] *Hoja de trucos de encabezados HTTP seguros - OWASP*. Consultado el 25 de junio de 2025. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html.

[61] *Hoja de trucos de prevención de inyección - OWASP*. Consultado el 7 de julio de 2025. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html.

[62] *Hoja de trucos de redirecciones y reenvíos no validados - OWASP*. Consultado el 26 de junio de 2025. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html.

[63] Hostinger. *Qué es el error "405 Method Not Allowed" y cómo solucionarlo*. Consultado el 21 de agosto de 2025. URL: https://www.hostinger.com/uk/tutorials/error-405-method-not-allowed.

[64] HowtoForge. *Auditoría de seguridad de Linux con Lynis*. https://howtoforge.es/auditoria-de-seguridad-de-linux-con-lynis/. Consultado el 20 de mayo de 2025.

[65] IBM. *Recomendaciones para protegerse contra ataques DDoS Slowloris*. https://www.ibm.com/docs/en/configurepricequote/10.0.0?topic=security-recommendations-protect-against-slowloris-ddos-attack. Consultado el 15 de julio de 2025.

[66] InfoSec Institute. *HTTP Verb Tampering: Bypassing Web Autenticación y autorización*. https://www.infosecinstitute.com/resources/application-security/http-verb-tempering-bypassing-web-authentication-and-authorization/. Consultado el 17 de julio de 2025.

[67] Internet Security Auditors. *Test de Intrusión en entornos Cloud*. Consultado el 7 de mayo de 2025. URL: https://www.isecauditors.com/test-intrusion-entornos-cloud.

[68] Internet Security Auditors. *Test de Intrusión Interno*. Consultado el 1 de mayo de 2025. URL: https://www.isecauditors.com/test-intrusion-interno.

[69] IT Digital Security. *4 herramientas gratuitas para el análisis de vulnerabilidades de apps y proyectos TIC*. Consultado el 16 de mayo de 2025. URL: https://www.itdigitalsecurity.es/vulnerabilidades/2024/02/4-herramientas-gratuitas-para-el-analisis-de-vulnerabilidades-de-apps-y-proyectos-tic.

[70] Kinsta. *Cómo solucionar advertencias de contenido mixto (mixed content)*. Consultado el 16 de agosto de 2025. URL: https://kinsta.com/blog/mixed-content-warnings/.

[71] *Manejo seguro de errores - hoja de trucos de OWASP*. Consultado el 15 de julio de 2025. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/cheatsheets/Error_Handling_Cheat_Sheet.html.

[72] Microsoft. *Boletin de Seguridad de Microsoft MS09-050 – Vulnerabilidades en SMBv2 podrían permitir la ejecución remota de código*. https://learn.microsoft.com/es-es/security-updates/securitybulletins/2009/ms09-050. Consultado el 17 de julio de 2025.

[73] MITRE Corporation. *CVE-2024-10453 — Elementor < 3.25.10: XSS almacenado (Contributor+) vía ajustes tipográficos*. Consultado el 6 de octubre de 2025. MITRE. 2024. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-10453.

[74] MITRE Corporation. *CVE-2024-13445 — Elementor < 3.27.5: XSS almacenado (Contributor+)*. Consultado el 6 de octubre de 2025. MITRE. 2024. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-13445.

[75] MITRE Corporation. *CVE-2024-50555 — Elementor < 3.29.1: XSS almacenado (Contributor+)*. Consultado el 6 de octubre de 2025. MITRE. 2024. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-50555.

[76] MITRE Corporation. *CVE-2024-5416 — Elementor < 3.24.0: XSS almacenado autenticado (Contributor+) vía parámetro URL*. Consultado el 6 de octubre de 2025. MITRE. 2024. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-5416.

[77] MITRE Corporation. *CVE-2024-54444 — Elementor < 3.25.11: XSS almacenado (Contributor+)*. Consultado el 6 de octubre de 2025. MITRE. 2024. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-54444.

[78] MITRE Corporation. *CVE-2024-6757 — Elementor 3.24.6: Exposición de información mediante texto alternativo de imágenes*. Consultado el 6 de octubre de 2025. MITRE. 2024. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-6757.

[79] MITRE Corporation. *CVE-2024-8236 — Elementor < 3.25.8: XSS almacenado (Contributor+)*. Consultado el 6 de octubre de 2025. MITRE. 2024. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-8236.

[80] MITRE Corporation. *CVE-2025-3075 — Elementor < 3.29.1: XSS almacenado (Contributor+)*. Consultado el 6 de octubre de 2025. MITRE. 2025. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-3075.

[81] MITRE Corporation. *CVE-2025-3457 — Ocean Extra < 2.4.7: XSS almacenado (Contributor+) a través de Shortcodes*. Consultado el 6 de octubre de 2025. MITRE. 2025. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-3457.

[82] MITRE Corporation. *CVE-2025-3458 — Ocean Extra < 2.4.7: XSS almacenado (Contributor+) mediante parámetro `ocean_gallery_id`*. Consultado el 6 de octubre de 2025. MITRE. 2025. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-3458.

[83] MITRE Corporation. *CVE-2025-3472 — Ocean Extra < 2.4.7: Ejecución arbitraria de Shortcodes sin autenticación*. Consultado el 6 de octubre de 2025. MITRE. 2025. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-3472.

[84] MITRE Corporation. *CVE-2025-4566 — Elementor < 3.30.3: XSS almacenado (Contributor+) vía Text Path Widget*. Consultado el 6 de octubre de 2025. MITRE. 2025. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-4566.

[85] MITRE Corporation. *CVE-2025-49068 — Ocean Extra < 2.4.9: XSS almacenado autenticado (Contributor+)*. Consultado el 6 de octubre de 2025. MITRE. 2025. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-49068.

[86] MITRE Corporation. *CVE-2025-8081 — Elementor < 3.30.3: Lectura arbitraria de ficheros (Administrador+) vía importación de imágenes*. Consultado el 6 de octubre de 2025. MITRE. 2025. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-8081.

[87] MITRE Corporation. *CVE-2025-9499 — Ocean Extra < 2.5.0: XSS almacenado (Contributor+) en versiones anteriores*. Consultado el 6 de octubre de 2025. MITRE. 2025. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-9499.

[88] Sho Nakatani. *RapidPen: Fully Automated IP-to-Shell Penetration Testing with LLM-based Agents*. Consultado el 8 de mayo de 2025. URL: https://arxiv.org/abs/2502.16730.

[89] Inc. NGINX. *Mitigando DDoS Attacks con NGINX y NGINX Plus*. https://blog.nginx.org/blog/mitigating-ddos-attacks-with-nginx-and-nginx-plus. Consultado el 15 de julio de 2025.

[90] Nimbus Tech. *Las 11 herramientas de pentesting más utilizadas*. Consultado el 16 de mayo de 2025. URL: https://nimbustech.es/ciberseguridad/herramientas-pentesting/.

[91] NIST / NVD. *CVE-2019-7659 Detalle - NVD*. https://nvd.nist.gov/vuln/detail/CVE-2019-7659. Consultado el 15 de julio de 2025.

[92] *OAuth 2.0: Mejores prácticas de seguridad actuales*. Consultado el 3 de julio de 2025. IETF. URL: https://www.ietf.org/archive/id/draft-ietf-oauth-security-topics-29.html.

[93] *Objetivos de seguridad y modelo de amenazas de DOMPurify*. Consultado el 5 de julio de 2025. Cure53. URL: https://github.com/cure53/DOMPurify/wiki/Security-Goals-%26-Threat-Model.

[94] NWebsec (documentación oficial). *Suprimir encabezados de versión (X-AspNet-Version, X-Powered-By) en ASP.NET*. Consultado el 18 de agosto de 2025. URL: https://nwebsec.readthedocs.io/en/aspnet4/nwebsec/Suppressing-version-headers.html.

[95] *OpenSSH – Security*. Consultado el 8 de octubre de 2025.

[96] OpenWebinars. *Fases del pentesting: Pasos para asegurar tus sistemas*. Consultado el 8 de mayo de 2025. URL: https://openwebinars.net/blog/fases-del-pentesting-pasos-para-asegurar-tus-sistemas/.

[97] OWASP. *A03 Inyección - OWASP Top 10:2021*. Consultado el 7 de mayo de 2025. URL: https://owasp.org/Top10/es/A03_2021-Injection/.

[98] OWASP. *API4: Falta de recursos y limitación de velocidad*. https://owasp.org/API-Security/editions/2019/en/0xa4-lack-of-resources-and-rate-limiting/. Consultado el 25 de septiembre de 2025.

[99] OWASP. *Hoja de referencia HTTP Strict Transport Security (HSTS)*. Consultado el 16 de agosto de 2025. URL: https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html.

[100] OWASP. *Inyección SQL — OWASP*. Consultado el 8 de octubre de 2025. URL: %5Curl%7Bhttps://owasp.org/www-community/attacks/SQL_Injection%7D.

[101] OWASP. *Pruebas de Cross-Origin Resource Sharing inseguro (CORS) en aplicaciones web*. Consultado el 17 de agosto de 2025. URL: https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/11-Client-side_Testing/07-Testing_Cross_Origin_Resource_Sharing.

[102] OWASP. *Testeando HTTP Verb Tampering (WSTG-INPV-03)*. https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/07-Input_Validation_Testing/03-Testing_for_HTTP_Verb_Tampering. Consultado el 17 de julio de 2025.

[103] OWASP. *WSTG – Probando atributos de cookies (Secure, HttpOnly, SameSite, prefijos)*. Consultado el 15 de agosto de 2025. URL: https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.

[104] Serie Cheat Sheet de OWASP. *Hoja de trucos de prevención de Cross-Site Request Forgery (CSRF) - OWASP*. https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html. Consultado el 20 de junio de 2025.

[105] Serie Cheat Sheet de OWASP. *Hoja de trucos de validación de entrada - OWASP*. https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html. Consultado el 20 de junio de 2025.

[106] OWASP Foundation. *Guía de Pruebas de Seguridad de Aplicaciones Móviles (MASTG)*. Consultado el 15 de mayo de 2025. URL: https://devguide.owasp.org/es/06-verification/01-guides/02-mastg/.

[107] OWASP Foundation. *Guía de Pruebas de Seguridad Web (WSTG)*. Consultado el 15 de mayo de 2025. URL: https://devguide.owasp.org/es/06-verification/01-guides/01-wstg/.

[108] OWASP Foundation. *Mobile Top 10 2024 – Final Release*. Consultado el 15 de mayo de 2025. URL: https://owasp.org/www-project-mobile-top-10/.

[109] Palentino Blog. *Mejores herramientas para pruebas de Pentesting*. Consultado el 16 de mayo de 2025. URL: https://www.palentino.es/blog/mejores-herramientas-para-pruebas-de-pentesting/.

[110] PrimeIT. *Cómo los hackers cambiaron la historia*. Consultado el 2 de mayo de 2025. URL: https://www.primeit.es/como-los-hackers-cambiaron-la-historia.

[111] Eclipse Jetty Project. *Anuncio de Jetty: vulnerabilidades HTTP/2 y nuevas versiones seguras*. Consultado el 20 de agosto de 2025. URL: https://www.eclipse.org/lists/jetty-announce/msg00181.html.

[112] WeakDH Project. *Guia para la implementación de Diffie-Hellman en TLS (WeakDH)*. https://weakdh.org/sysadmin.html. Consultado el 17 de julio de 2025.

[113] Eric S. Raymond. *Breve historia de la cultura hacker*. Trans. by Abel R. Micó. Consultado el 2 de mayo de 2025, Publicado bajo la Open Publication License, versión 2.0. URL: https://biblioweb.sindominio.net/telematica/historia-cultura-hacker.html.

[114] *Revelación de IP privada - alerta en ZAP (Private IP Disclosure)*. Consultado el 27 de junio de 2025. OWASP ZAP. URL: https://www.zaproxy.org/docs/alerts/2/.

[115] David Santo Orcero. *Kali Linux*. Ra-Ma Editorial, 2018.

[116] David Santo Orcero. *Pentesting Con Kali*. 2024.

[117] X Sec. *WPScan Comandos*. https://xsec.sh/blog/wpscan-comandos/. Consultado el 20 de mayo de 2025.

[118] Secureframe. *SOC 2 Compliance: requisitos, proceso de auditoría y beneficios*. https://secureframe.com/es-es/blog/soc-2-compliance-guide. Consultado el 15 de mayo de 2025.

[119] Astra Security. *Cómo proteger el archivo wp-config en WordPress*. Consultado el 15 de agosto de 2025. URL: https://www.getastra.com/blog/wordpress-security-course/how-to-secure-wp-config-file/.

[120] Tarlogic Security. *Pruebas de seguridad de apps móviles: Proteger a las compañías y a sus clientes*. Consultado el 15 de mayo de 2025. URL: https://www.tarlogic.com/es/blog/pruebas-seguridad-apps-moviles/.

[121] SecurityScorecard. *Qué es UPnP y por qué representa un riesgo de seguridad*. Consultado el 16 de agosto de 2025. URL: https://securityscorecard.com/blog/what-is-upnp-and-why-is-it-a-security-risk/.

[122] Tenable / comunidad de seguridad. *Vulnerabilidad del script `bigconf.cgi` en F5 BIG/IP: ejecución / acceso arbitrario*. Consultado el 17 de agosto de 2025. 1999. URL: https://www.tenable.com/plugins/nessus/10027.

[123] OWASP Cheat Sheet Series. *Hoja de trucos de política de seguridad de contenido - inyección y uso de cabecera CSP (OWASP)*. https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html. Consultado el 24 de junio de 2025.

[124] OWASP Cheat Sheet Series. *Hoja de trucos de redirecciones y reenvíos no validados - OWASP*. https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html. Consultado el 20 de junio de 2025.

[125] *Server_tokens off en Nginx y eliminación total de la cabecera Server - discusión técnica*. Consultado el 25 de junio de 2025. Server Fault. URL: https://serverfault.com/questions/214242/can-i-hide-all-server-os-info.

[126] Justin Scott / comunidad ServerFault. *Cómo eliminar encabezados HTTP de ASP.NET / IIS (X-AspNet-Version, X-AspNetMvc-Version, etc.)* Consultado el 17 de agosto de 2025. URL: https://serverfault.com/questions/24885/how-to-remove-iis-asp-net-response-headers.

[127] Sonatype. *The OWASP ZAP HUD*. https://www.sonatype.com/blog/the-owasp-zap-hud. Consultado el 20 de mayo de 2025.

[128] *SSH agente*. Consultado el 8 de octubre de 2025.

[129] *SSH Mejores prácticas para usarlo de forma segura*. Consultado el 8 de octubre de 2025.

[130] Comunidad StackExchange. *Cómo probar que se han deshabilitado correctamente los métodos HTTP innecesarios*. Consultado el 21 de agosto de 2025. URL: https://security.stackexchange.com/questions/45694/how-can-i-test-that-i-have-correctly-disabled-unnecessary-http-methods.

[131] Jonathan Munshaw / Cisco Talos. *Vulnerabilidad Spotlight: Vulnerabilidades múltiples en Genivia gSOAP*. https://blog.talosintelligence.com/vuln-spotlight-genivia-gsoap-/. Consultado el 15 de julio de 2025.

[132] Tarlogic. *Metodología NIST: Sustento para los analistas de ciberseguridad*. https://www.tarlogic.com/es/blog/guias-nist-ciberseguridad/. Consultado el 20 de mayo de 2025.

[133] phpMyAdmin Team. *PMASA-2005-4: Inclusion de archivos locales en grab_globals.lib.php, arreglo en phpMyAdmin 2.6.4-pl2*. https://www.phpmyadmin.net/security/PMASA-2005-4/. Consultado el 15 de julio de 2025.

[134] TechTarget. *Vulnerabilidad de UPnP: cómo puede usarse indebidamente*. Consultado el 16 de agosto de 2025. URL: https://www.techtarget.com/searchsecurity/answer/UPnP-vulnerability-How-is-the-UPnP-protocol-being-misused.

[135] Tecnek Ciberseguridad. *Tipos de pruebas de intrusión*. Consultado el 20 de mayo de 2025. URL: https://www.tecnek.com/noticias-ciberseguridad/180-tipos-de-pruebas-de-intrusion.html.

[136] Microsoft / Broadcom / documentación técnica. *Cómo deshabilitar el método HTTP TRACE en IIS*. Consultado el 25 de julio de 2025. URL: https://knowledge.broadcom.com/external/article/389342/how-to-disable-trace-http-method-in-iis.html.

[137] *TLS para proteger Basic Auth - Security Stack Exchange*. Consultado el 3 de julio de 2025. Stack Exchange. URL: https://security.stackexchange.com/questions/138725/tls-to-secure-basic-http-auth.

[138] Steffen Ullrich. *¿Por qué no debería permitirse el método OPTIONS en un servidor HTTP?* Consultado el 25 de julio de 2025. URL: https://security.stackexchange.com/questions/138567/why-should-the-options-method-not-be-allowed-on-an-http-server.

[139] *UNE-EN ISO/IEC 27002:2023 Seguridad de la información, ciberseguridad y protección de la privacidad. Control de la seguridad de la información*. https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0071321. Consultado el 15 de mayo de 2025.

[140] *Verificación de tokens anti-CSRF - Zed Attack Proxy (ZAP)*. https://www.zaproxy.org/docs/alerts/20012/. Consultado el 20 de junio de 2025.

[141] Veselin. *Guía Completa: Cómo Realizar un Pentesting en Aplicaciones Móviles*. Consultado el 16 de mayo de 2025. URL: https://veselin.es/guia-completa-como-realizar-un-pentesting-en-aplicaciones-moviles/.

[142] Wallarm. *¿Qué es OWASP Zed Attack Proxy (ZAP)?* https://lab.wallarm.com/what/owasp-zap-proxy-de-ataque-zed/?lang=es. Consultado el 20 de mayo de 2025.

[143] Raspiblog / técnicas de administración web. *Protección en Apache con mod_qos y límites de tasa*. https://www.liquidweb.com/help-docs/security/using-mod_qos-and-mod_reqtimeout-to-mitigate-slowloris-attacks/. Consultado el 15 de julio de 2025.

[144] Wikipedia. *ISO/IEC*. Consultado el 2 de mayo de 2025. URL: https://es.wikipedia.org/wiki/ISO/IEC_27001.

[145] Wikipedia. *Metasploit*. Consultado el 16 de mayo de 2025. URL: https://es.wikipedia.org/wiki/Metasploit.

[146] Wikipedia. *Nmap*. Consultado el 16 de mayo de 2025. URL: https://es.wikipedia.org/wiki/Nmap.

[147] Wikipedia. *OWASP ZAP*. Consultado el 16 de mayo de 2025. URL: https://es.wikipedia.org/wiki/OWASP_ZAP.

[148] WordPress.org. *Endurecimiento de WordPress: proteger archivos sensibles*. Consultado el 15 de agosto de 2025. URL: https://developer.wordpress.org/advanced-administration/server/web-server/httpd/.

[149] WPScan. *WPScan: WordPress Security Scanner*. https://wpscan.com/. Consultado el 20 de mayo de 2025.

[150] José Luis Guillén Zafra. "Introducción al pentesting". Dirigido por Oriol Pujol Vila. Licencia Creative Commons. Trabajo de Fin de Grado. Barcelona, España: Universitat de Barcelona. URL: https://diposit.ub.edu/dspace/bitstream/2445/124085/2/memoria.pdf.

UNIVERSIDAD
DE MÁLAGA

uma.es

E.T.S. de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga