**INFORME FINAL DEL PROYECTO**

**DE SEGURIDAD INFORMATICA**

Instalación, escaneo y mitigación de vulnerabilidades con

Nessus, Nmap, OpenVas en Linux, como parte de la Certificación

en Ciberseguridad IRSI – SISAP. Última revisión: junio de 2025.

Contenido

[1.INTRODUCCIÓN 3](#_Toc202021530)

[2. OBJETIVOS DEL PROYECTO 3](#_Toc202021531)

[Objetivo General 3](#_Toc202021532)

[Objetivos Específicos 3](#_Toc202021533)

[3. FUNDAMENTO NORMATIVO 3](#_Toc202021534)

[4. INSTALACIÓN Y CONFIGURACIÓN DE ENTORNOS 3](#_Toc202021535)

[4.1 Instalación del sistema operativo objetivo (Debian 12) 3](#_Toc202021536)

[4.2 Instalación y configuración de Nessus en Kali 8](#_Toc202021537)

[4.3 Escaneo de vulnerabilidades 8](#_Toc202021538)

[Nmap: 8](#_Toc202021539)

[Nessus: 11](#_Toc202021540)

[5.Análisis Consolidado de Vulnerabilidades – Nmap y Nessus 17](#_Toc202021541)

[6.Recomendaciones Generales de Mitigación 18](#_Toc202021542)

[7.Resumen del laboratorio de mitigación de vulnerabilidades 21](#_Toc202021543)

[8. CONCLUSIONES 21](#_Toc202021544)

[9. GLOSARIO TÉCNICO 21](#_Toc202021545)

[10. FUENTES DE CONSULTA 22](#_Toc202021546)

# 1.INTRODUCCIÓN

Este informe describe el desarrollo de un laboratorio práctico en ciberseguridad, enfocado en el reconocimiento, análisis y mitigación de vulnerabilidades presentes en sistemas operativos Linux. Para esta practica se utilizaron herramientas de escaneo como Nessus, Nmap, desplegados en un entorno virtual con Kali Linux y Debian 12, con configuraciones deliberadamente vulnerables.

El objetivo fue simular un entorno real de auditoría técnica, fortaleciendo habilidades para la gestión de riesgos, priorización de vulnerabilidades y aplicación de controles técnicos, en alineación con buenas prácticas y estándares como la norma internacional ISO/IEC 27001:2022.

# 2. OBJETIVOS DEL PROYECTO

Objetivo General**:** Desarrollar habilidades técnicas para la identificación, análisis y mitigación de vulnerabilidades en sistemas Linux, utilizando herramientas profesionales y marcos normativos reconocidos en el ámbito de la ciberseguridad.

Objetivos Específicos**:**

* Instalar y configurar entornos virtualizados con Kali Linux (sistema atacante) y Debian (sistema objetivo).
* Ejecutar escaneos de red con Nmap.
* Detectar vulnerabilidades con Nessus.
* Aplicar mitigaciones técnicas en el sistema objetivo.
* Documentar los resultados conforme a estándares técnicos.

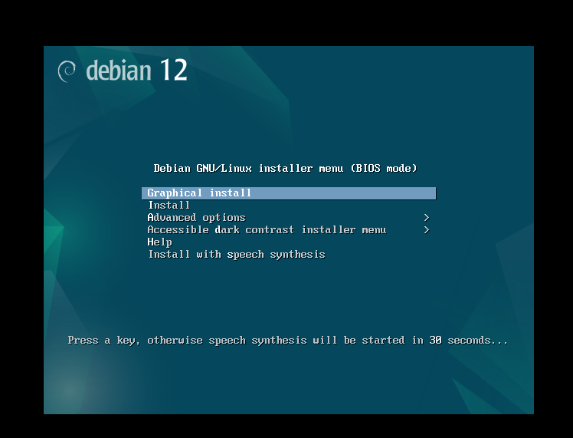
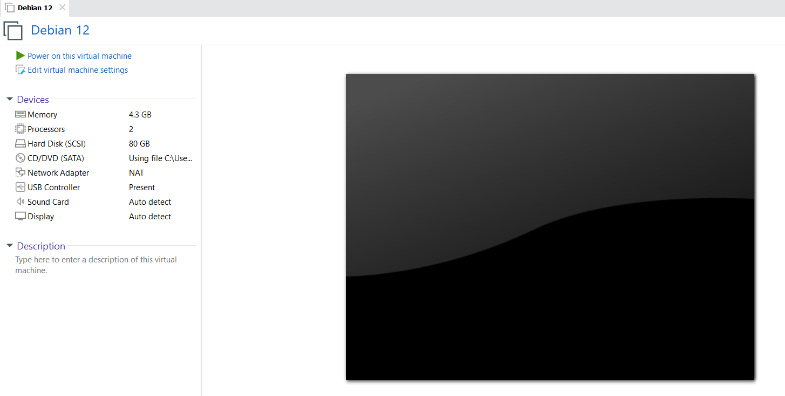
# 3. FUNDAMENTO NORMATIVO

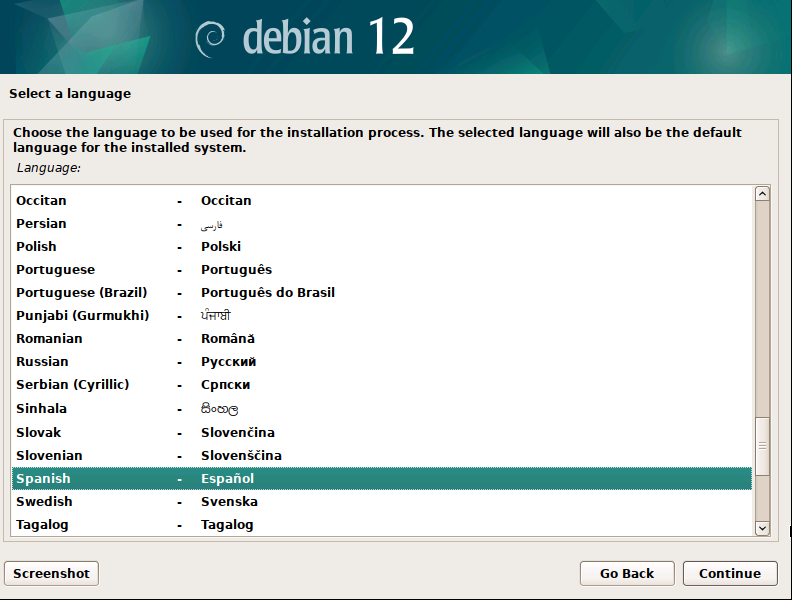
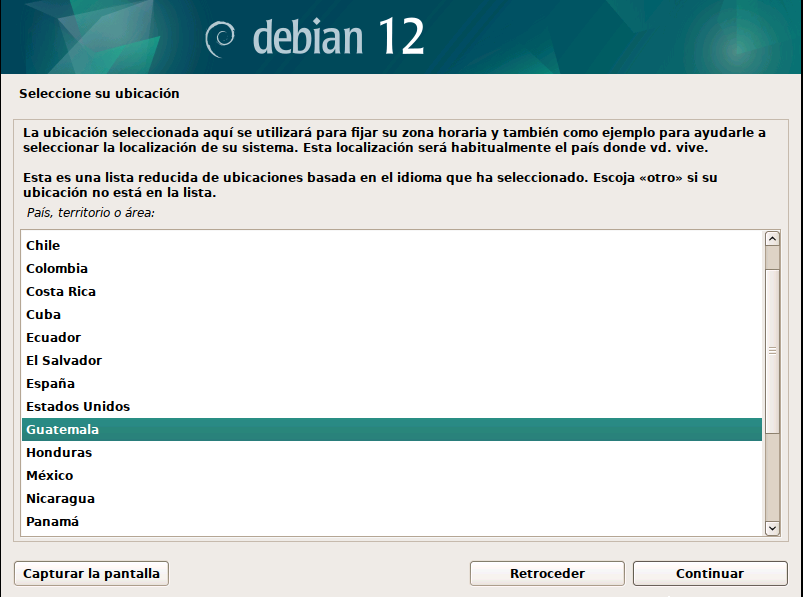
Para la ejecución del presente laboratorio, se tomo como referencia los lineamientos de la norma ISO/IEC 27001:2022, en especial los controles relacionados con la gestión de vulnerabilidades técnicas (A.12.6.1), protección contra software malicioso (A.12.2.1), y gestión de la configuración de seguridad (A.9.4.1). También se consideraron prácticas recomendadas de hardening del CIS Benchmark para Debian Linux y guías de seguridad publicadas por Tenable y Nmap.org.

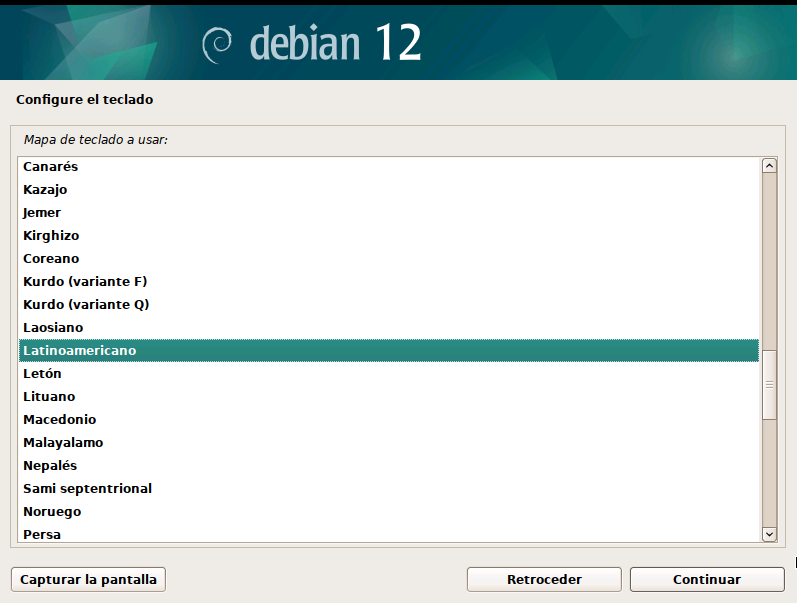
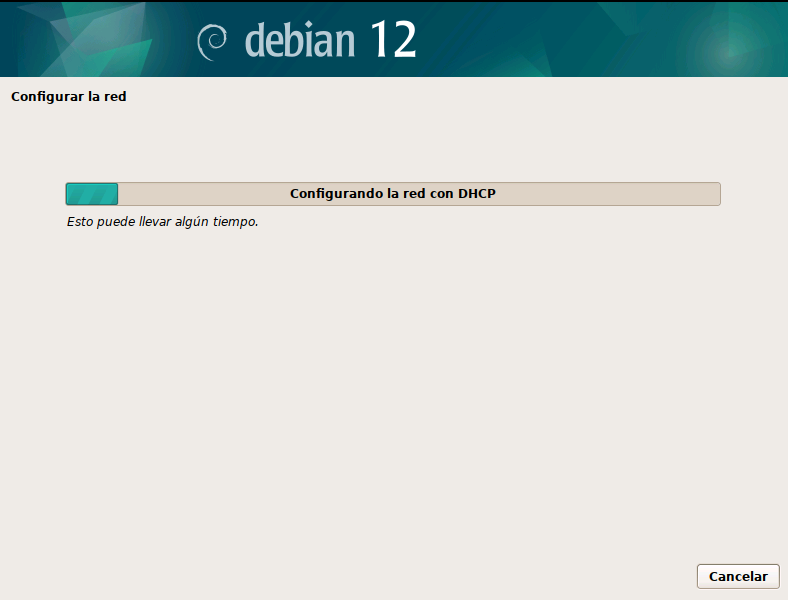
# 4. INSTALACIÓN Y CONFIGURACIÓN DE ENTORNOS

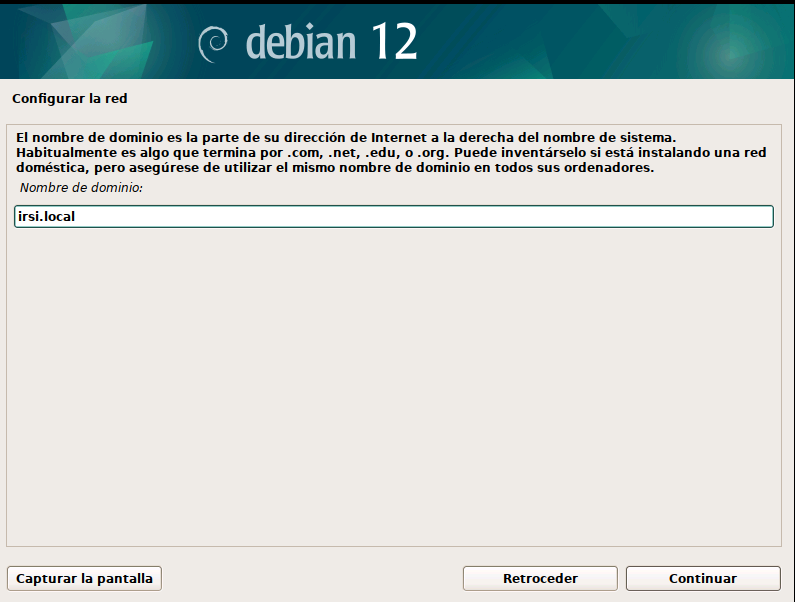
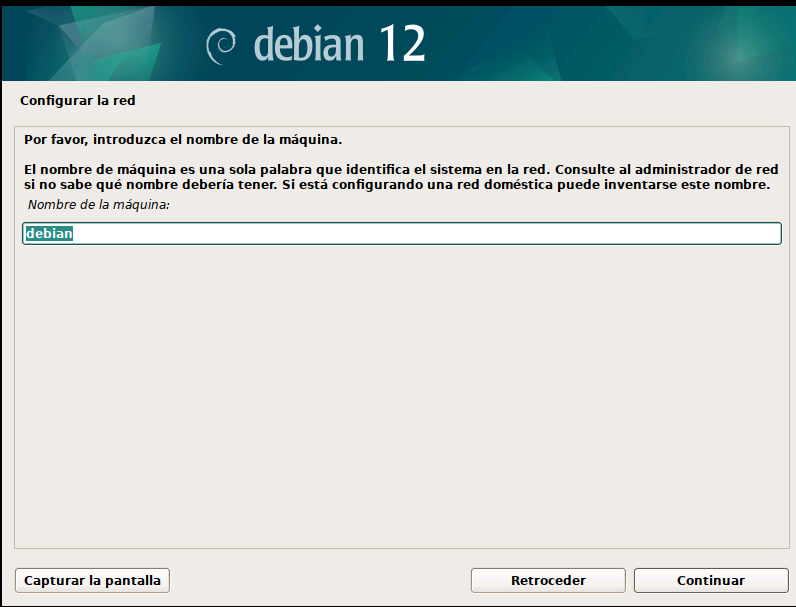
## 4.1 Instalación del sistema operativo objetivo (Debian 12)

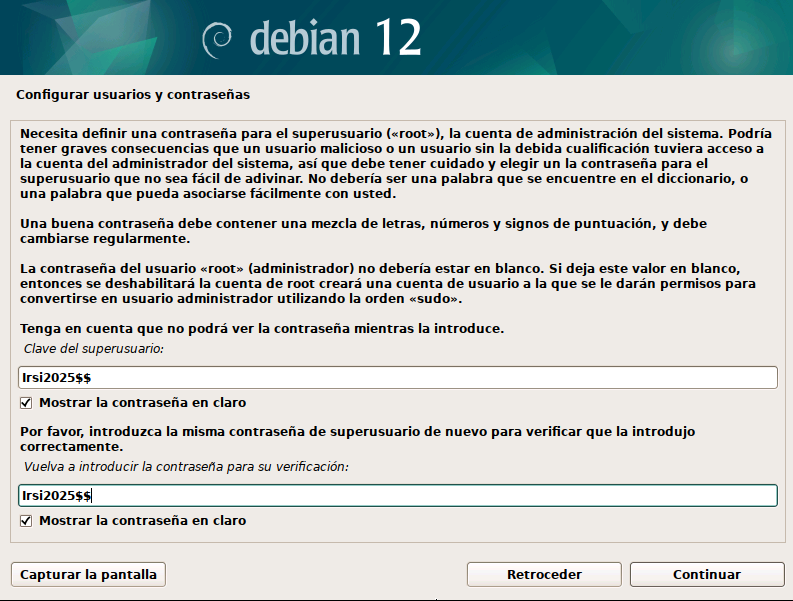
Se instalo Debian 12 (64 bits) en VMware Workstation 17 Pro con una interfaz de red en modo host-only, sin firewall ni actualizaciones automáticas. El sistema fue deliberadamente expuesto con servicios comunes como SSH, Apache y FTP, tal como se observa en las siguientes capturas:

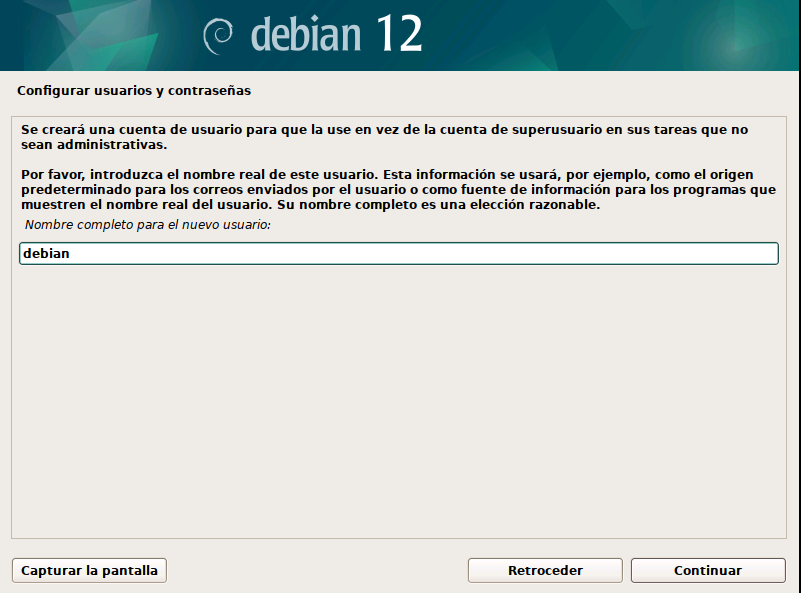


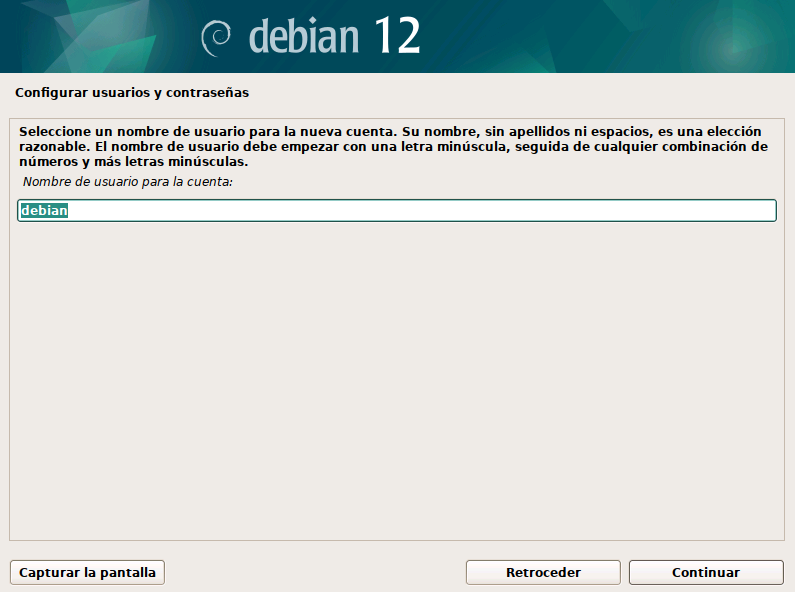
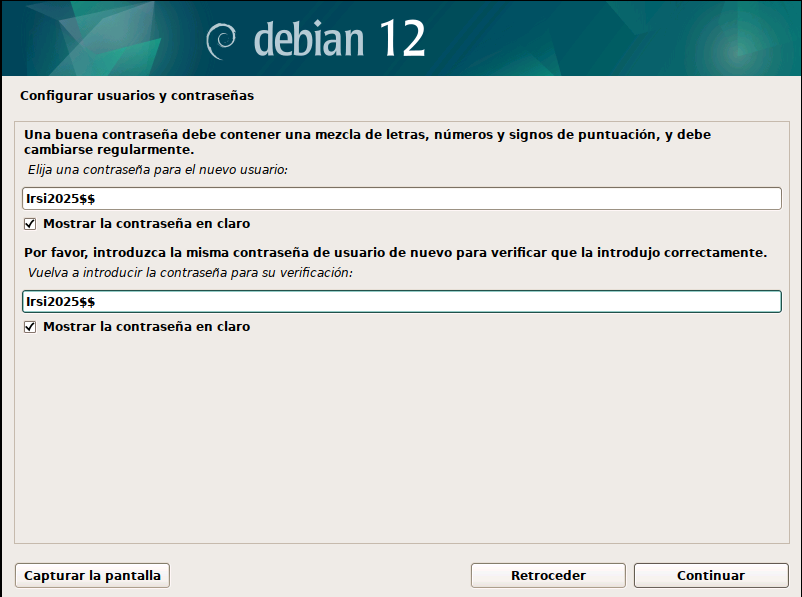


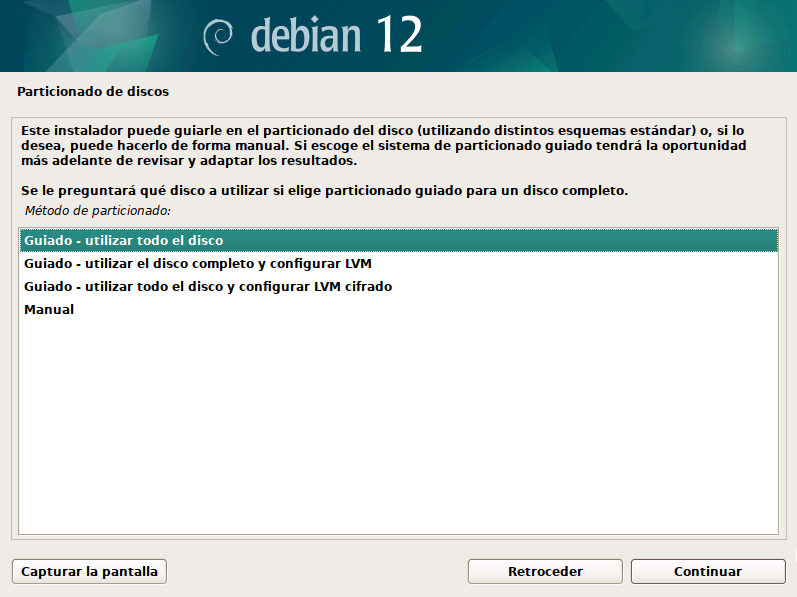


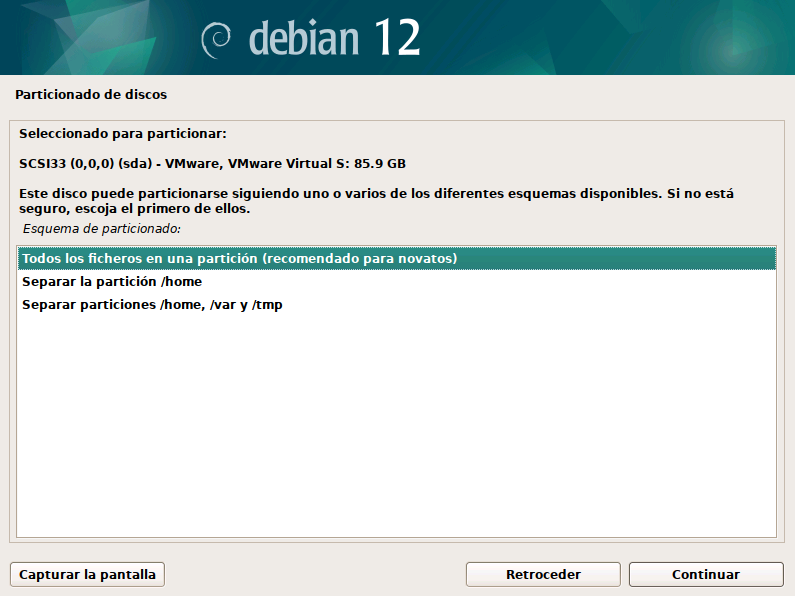


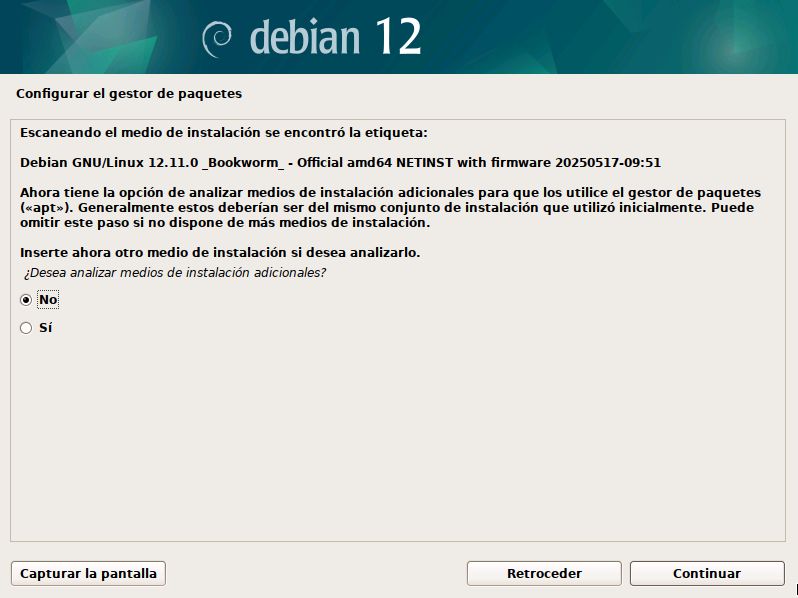
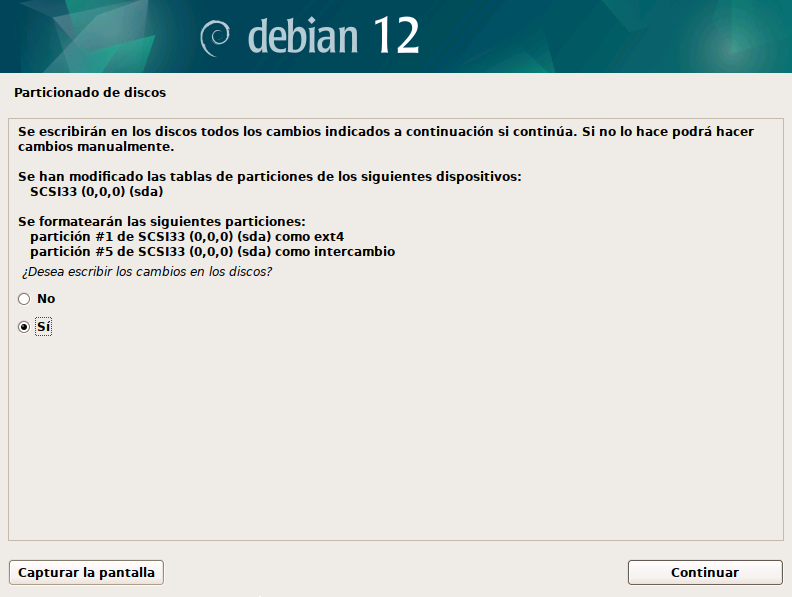


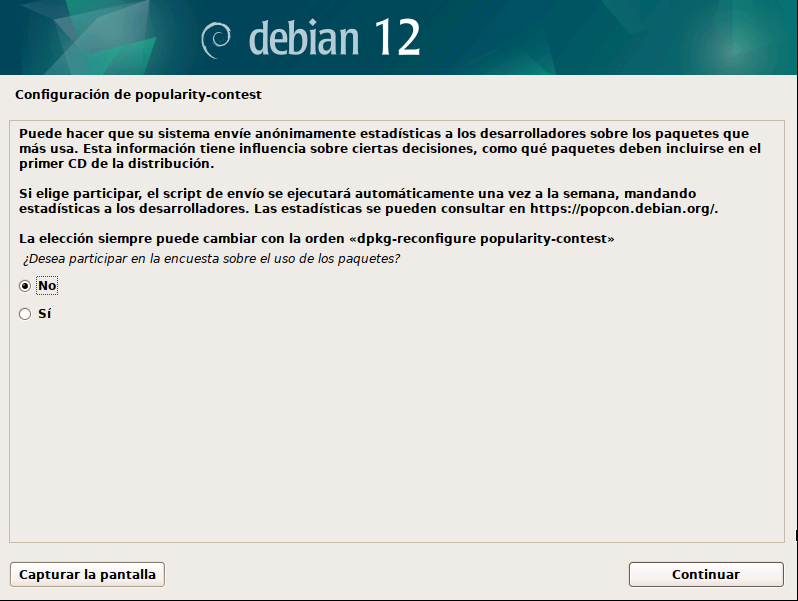
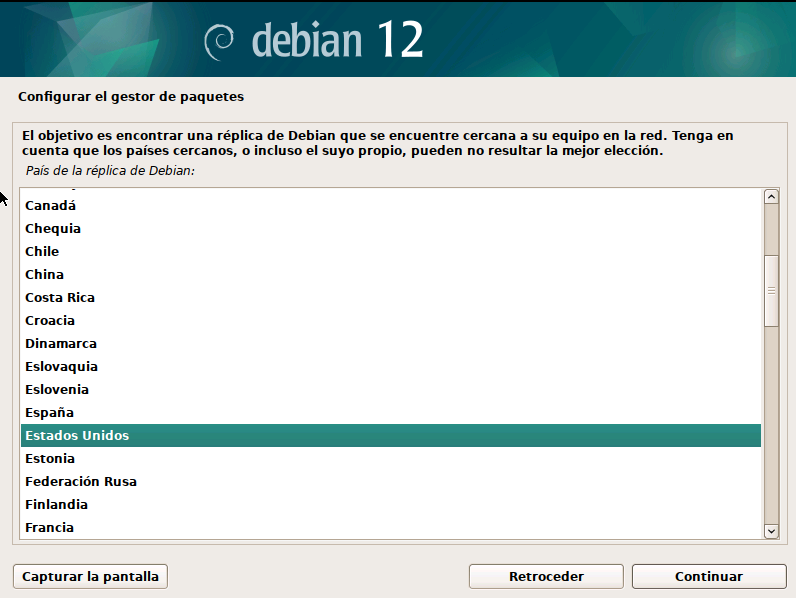




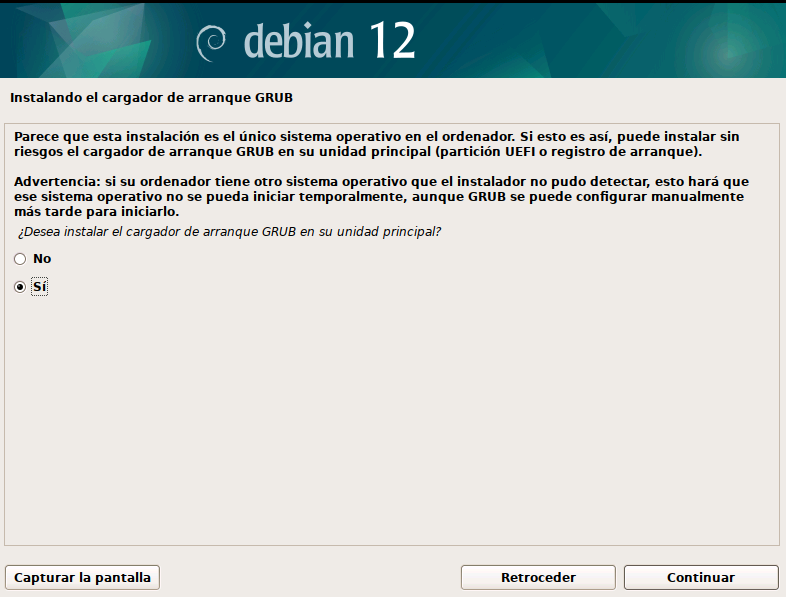


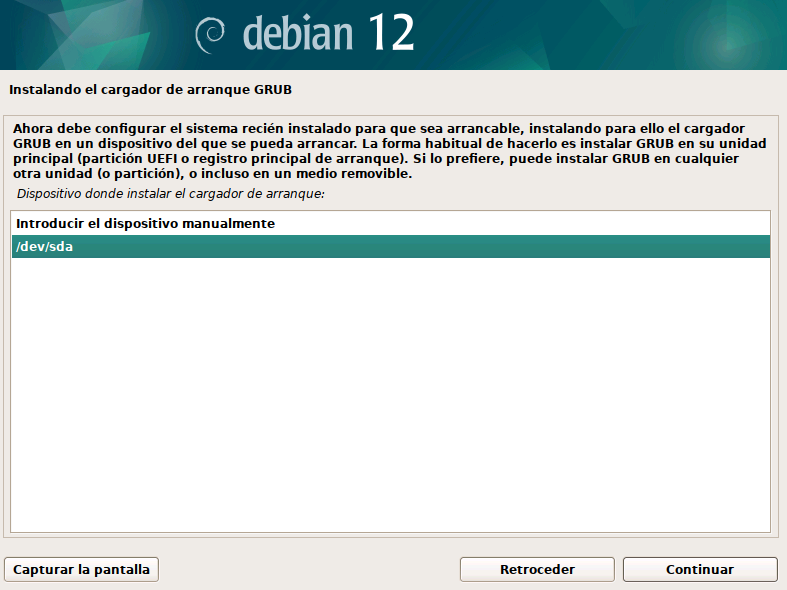
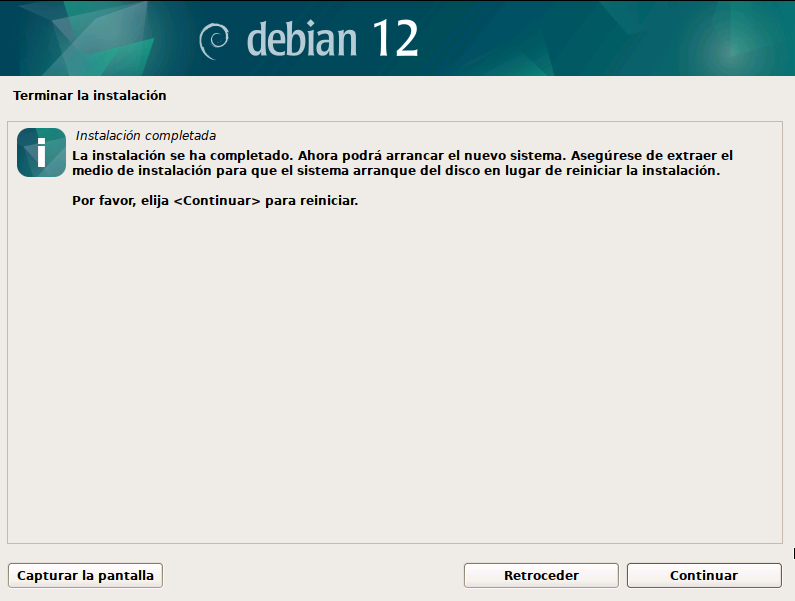


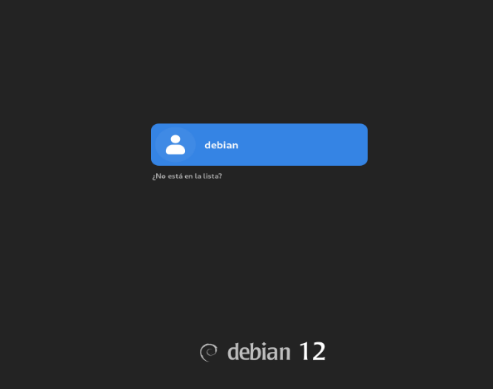
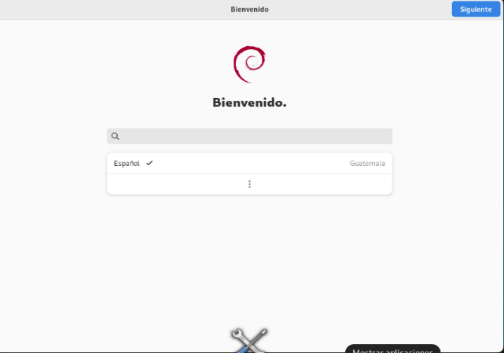


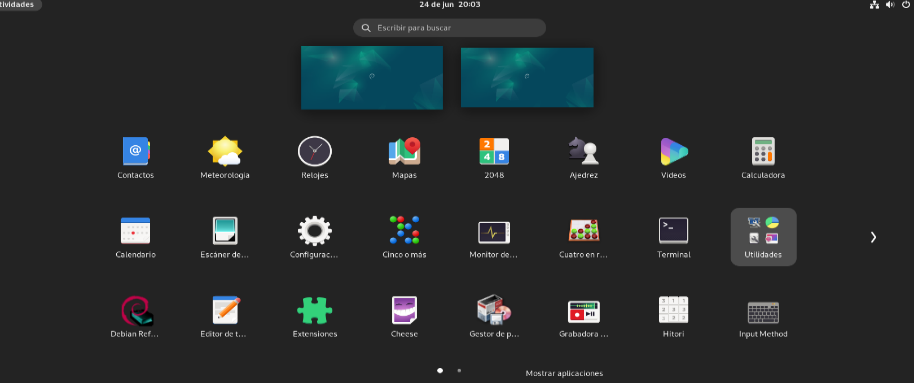




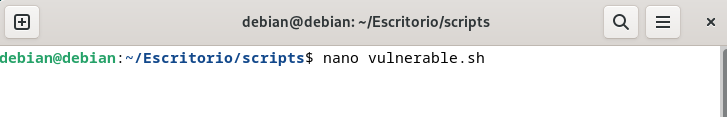


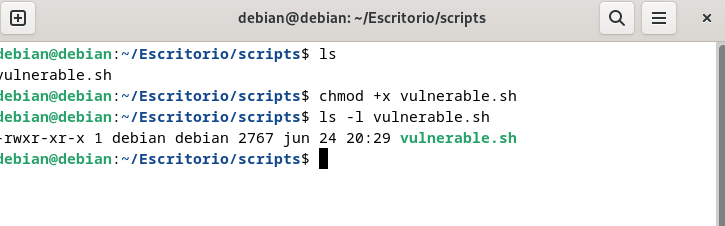


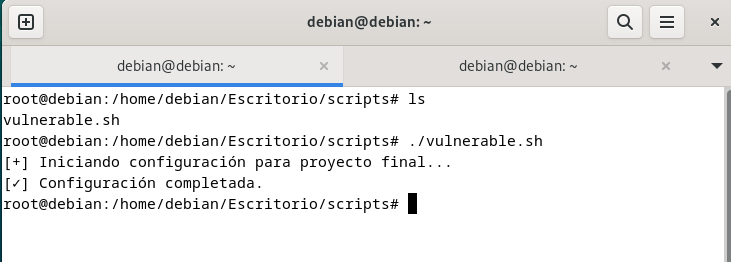




Coadyuvando a lo anterior y con fines de realizar el laboratorio con los parametros solicitados se procedio a ejecutar el script que se nos proporciono de la siguente manera:







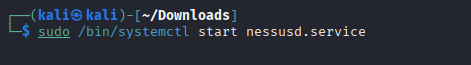
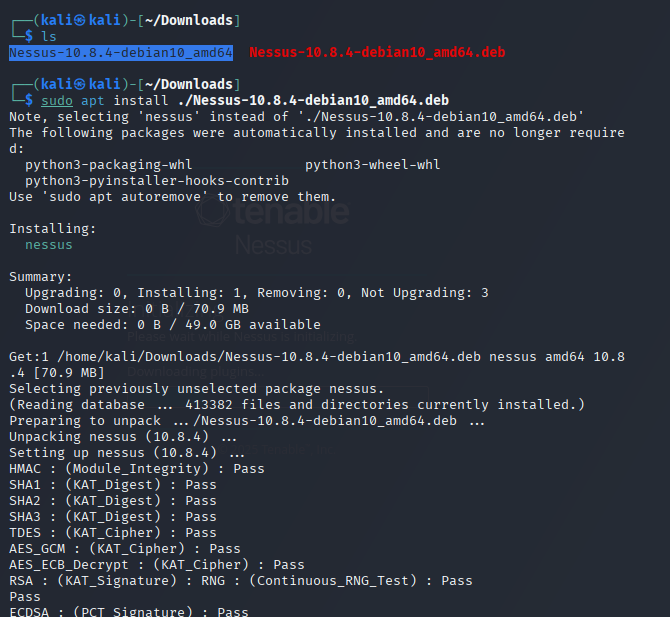
# 4.2 Instalación y configuración de Nessus en Kali

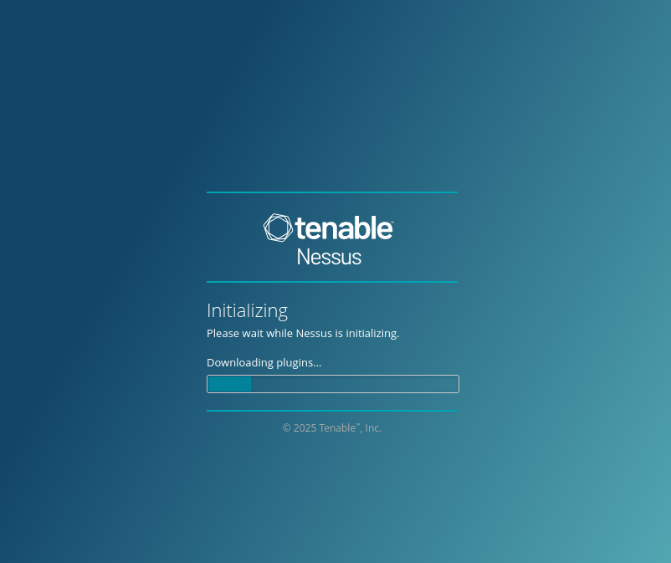
Se utilizó Kali Linux como sistema de ataque. Las herramientas instaladas son:

* **Nessus Essentials** (versión gratuita de Tenable).
* **Nmap v7.94**.

**Pasos realizados:**

1. Instalación de Nessus vía .deb en Kali.
2. Registro de cuenta gratuita para habilitar escaneo de vulnerabilidades.
3. Configuración de red con acceso al objetivo Debian.

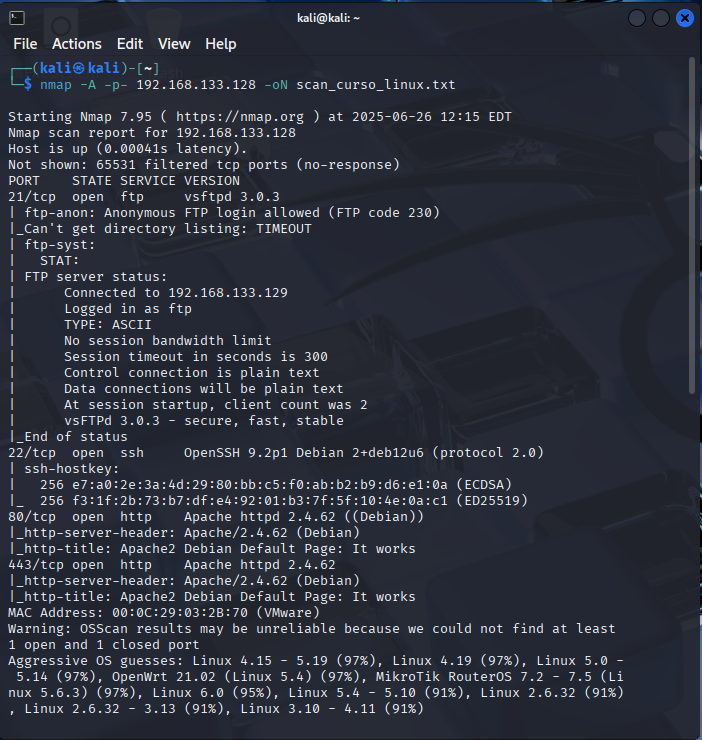
**Captura:**



# 4.3 Escaneo de vulnerabilidades

# Nmap:

Se ejecutó el siguiente comando para identificar los servicios activos, versiones y posibles vulnerabilidades en la máquina objetivo (192.168.133.128): **nmap -A -p- 192.168.133.128 -oN scan\_curso\_linux.txt**



Este escaneo incluyó detección de sistema operativo, servicios en todos los puertos TCP, versiones, y scripts de enumeración básicos.

**Información general del sistema:**

* **Sistema operativo estimado**: Linux (Kernel entre 4.x y 6.x, probablemente Debian 12).
* **Hostname detectado**: debian.irsi.local
* **MAC Address**: 00:0C:29:03:2B:70 (VMware)
* **Distancia de red**: 1 salto (misma red local)
* **Advertencia de Nmap**: OSScan no completamente fiable debido a falta de puertos cerrados.

**Puertos abiertos y servicios detectados:**

**Puerto 21/tcp – FTP**

* **Servicio**: vsftpd 3.0.3
* **Hallazgos**:
  + Acceso anónimo habilitado (ftp-anon: FTP code 230).
  + No se obtuvo listado de directorios (timeout), pero la autenticación anónima es funcional.
  + Transmisión en texto plano (sin SSL).
  + Servidor muestra banner con detalles de estado (ftp-syst).
* **Implicación**: Un atacante podría conectarse como anónimo y listar, descargar o cargar archivos si se permite. También podría usarse para enumeración de usuarios o realizar fuerza bruta.

**Puerto 22/tcp – SSH**

* **Servicio**: OpenSSH 9.2p1 Debian 2+deb12u6
* **Hallazgos**:
  + Claves de host visibles: ECDSA (256 bits), ED25519 (256 bits).
  + El script de laboratorio habilitó cifrados y algoritmos débiles: diffie-hellman-group1-sha1, hmac-md5, etc.
* **Implicación**: Aunque el servicio es reciente, el uso de algoritmos obsoletos puede permitir ataques tipo downgrade, MITM o ruptura de cifrado en redes inseguras.

**Puerto 80/tcp – HTTP**

* **Servicio**: Apache httpd 2.4.62 (Debian)
* **Hallazgos**:
  + Página por defecto activa (It works), sin contenido personalizado.
  + No se detectaron headers de seguridad adicionales.
* **Implicación**: Aunque no hay contenido expuesto directamente, esto indica una posible falta de hardening. Si se agregan archivos sensibles o módulos como mod\_php, el riesgo puede aumentar.

**Puerto 443/tcp – HTTPS**

* **Servicio**: Apache httpd 2.4.62
* **Hallazgos**:
  + Página por defecto también activa sobre HTTPS.
  + El servidor permite todos los protocolos (SSLProtocol all), lo que puede incluir SSLv2 y SSLv3 si están habilitados.
* **Implicación**: Protocolos antiguos como SSLv2 y SSLv3 son inseguros y permiten ataques como POODLE. Si un cliente los usa, la conexión podría ser vulnerable.

**Resumen de hallazgos críticos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Puerto** | **Servicio** | **Vulnerabilidad principal** | **Riesgo** |
| 21 | vsftpd 3.0.3 (FTP) | Acceso anónimo, texto plano, sin cifrado | **Alto** |
| 22 | OpenSSH 9.2p1 | Algoritmos de cifrado obsoletos habilitados | **Medio** |
| 80 | Apache 2.4.62 (HTTP) | Página por defecto activa, falta de hardening | **Bajo – Medio** |
| 443 | Apache 2.4.62 (HTTPS) | Posible soporte de SSLv2/SSLv3, configuración insegura | **Medio** |

**Tabla de vulnerabilidades detectadas con NMAP y recomendaciones :**

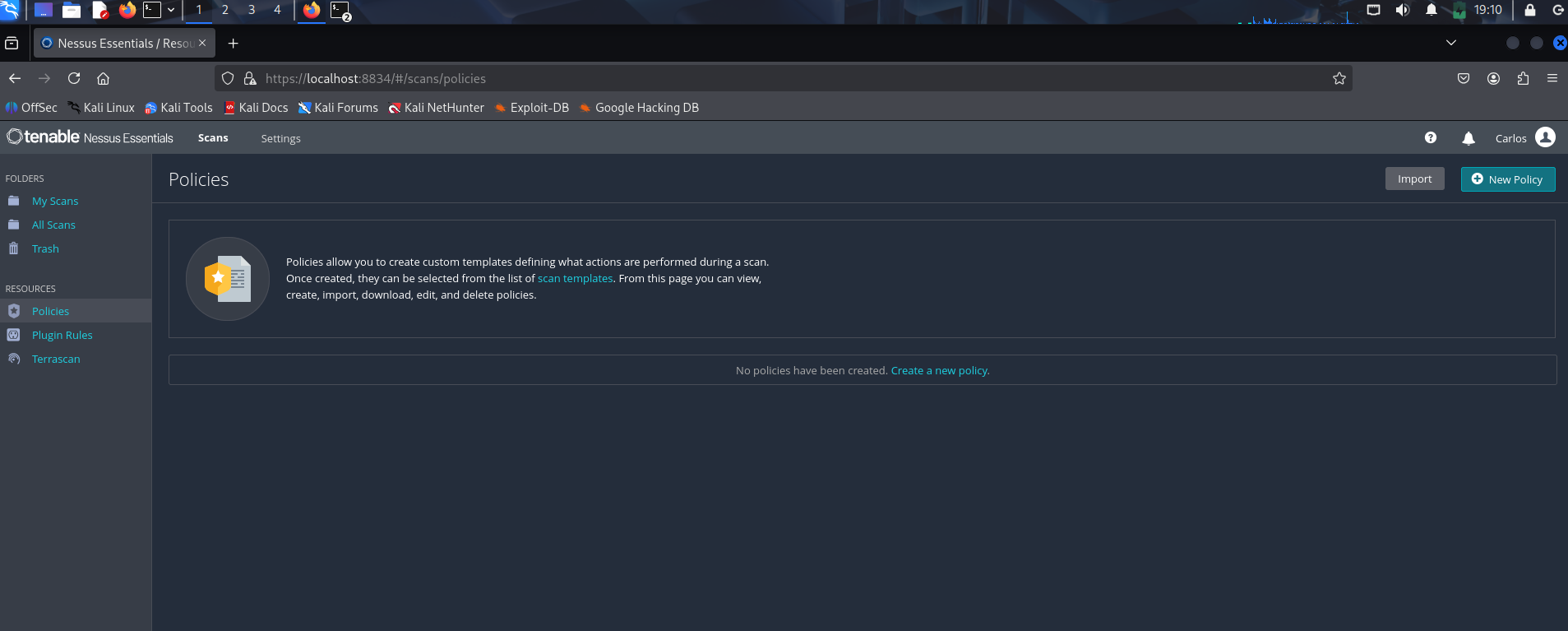
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Puerto** | **Servicio** | **Descripción del hallazgo** | **Nivel de riesgo** | **Recomendación** |
| 21/tcp | vsftpd 3.0.3 (FTP) | Acceso anónimo habilitado (ftp-anon). Control y datos sin cifrado. | **Alto** | Deshabilitar acceso anónimo (anonymous\_enable=NO) y activar SSL (ssl\_enable=YES). |
| 22/tcp | OpenSSH 9.2p1 | Uso de algoritmos criptográficos débiles configurados manualmente (ej. diffie-hellman-group1-sha1, hmac-md5). | **Medio** | Restringir a cifrados modernos en sshd\_config (Ciphers, KexAlgorithms, MACs). |
| 80/tcp | Apache 2.4.62 (HTTP) | Página por defecto activa (It works). Posible falta de hardening. | **Bajo – Medio** | Reemplazar página por contenido personalizado. Revisar módulos activos. |
| 443/tcp | Apache 2.4.62 (HTTPS) | Configuración SSL permite protocolos inseguros (SSLProtocol all). Potencial exposición a SSLv2/SSLv3. | **Medio** | Configurar ssl.conf para deshabilitar SSLv2/3. Usar solo TLS 1.2/1.3 con ciphers fuertes. |

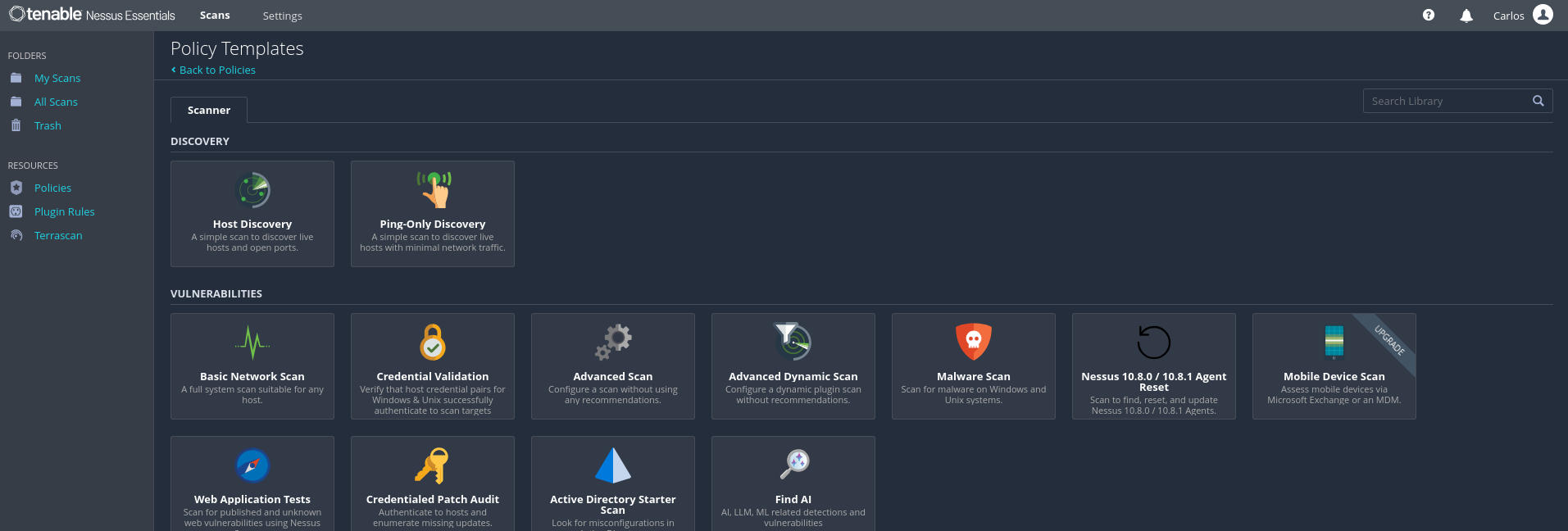
**Observaciones adicionales:**

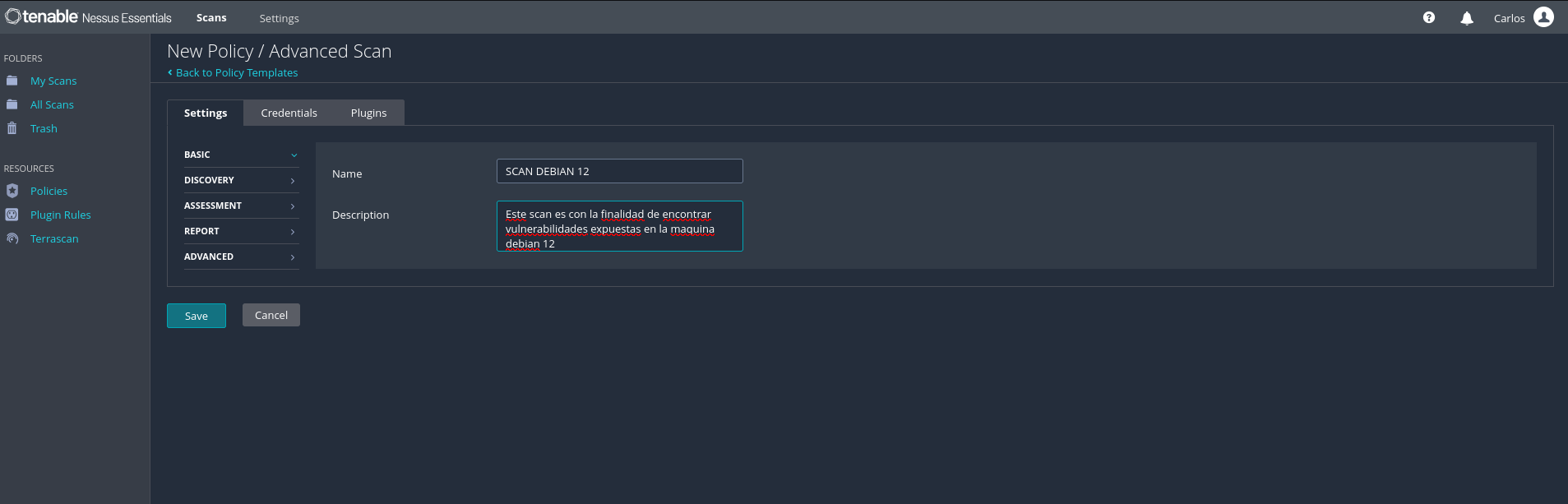
* El puerto FTP permitió login anónimo con código de respuesta 230, aunque el listado de directorios falló por timeout. Esto sigue representando una amenaza si no se controla el acceso a archivos sensibles.
* El servicio SSH, si bien moderno, fue reconfigurado por el script de laboratorio para aceptar algoritmos obsoletos, lo cual intencionalmente reduce su seguridad para efectos educativos.
* Apache, tanto en HTTP como en HTTPS, está activo con su configuración básica, lo que permite detectar posibles vectores comunes como exposiciones de contenido o protocolos no cifrados.

## Nessus:

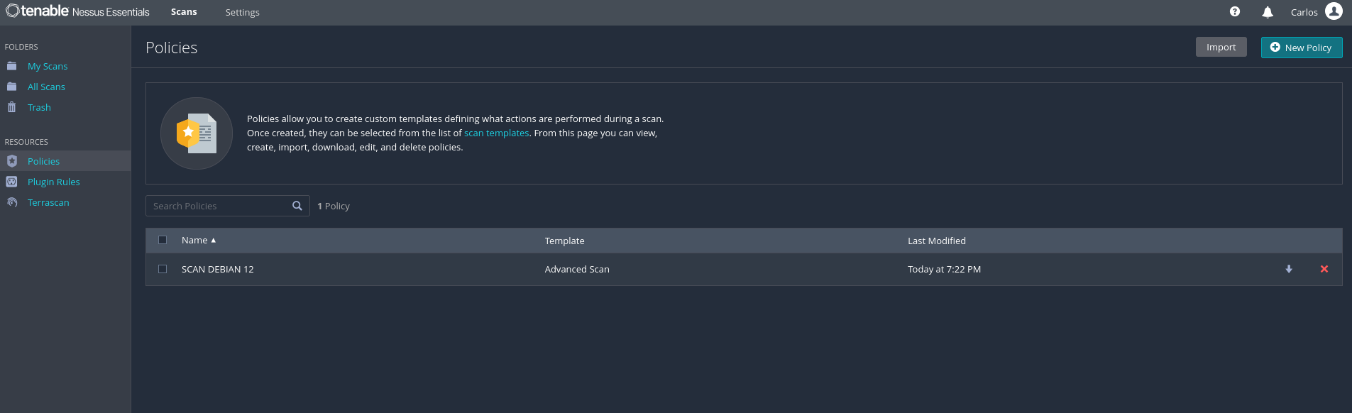
Como primer punto tenemos la configuracion de nuestra politica de scaneo la cual se la aplicaremos a la maquina o host que deseamos auditar.



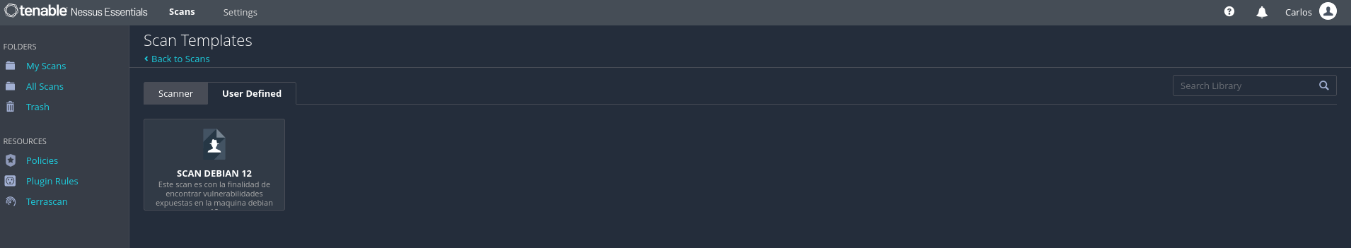




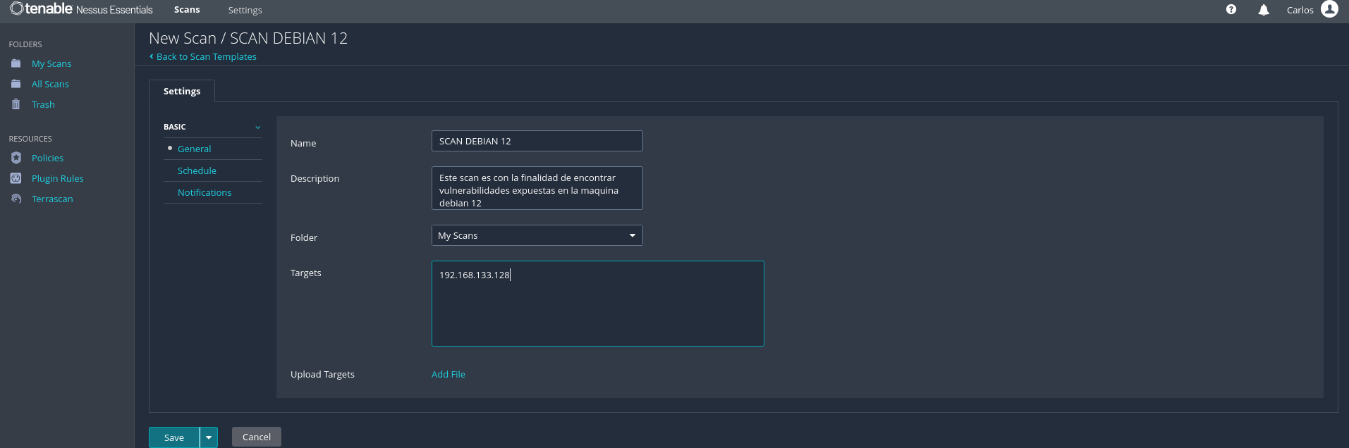
Ya con figurada la politica que usaremos nos quedara de esta manera lista para ejecutarla, harememos la ejecucion de esta politica atravez del boton de new scan user defined

****

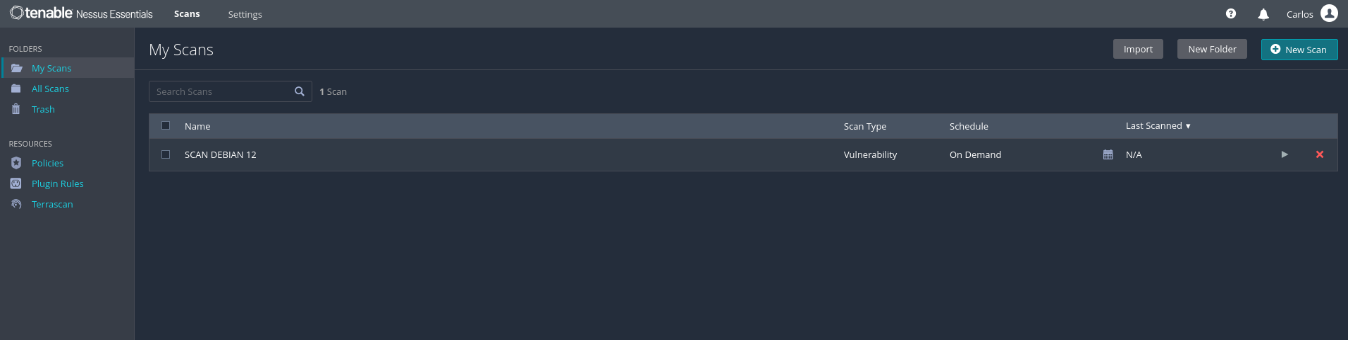
Tenemos listo nuestro scan para poderlo aplicar

****

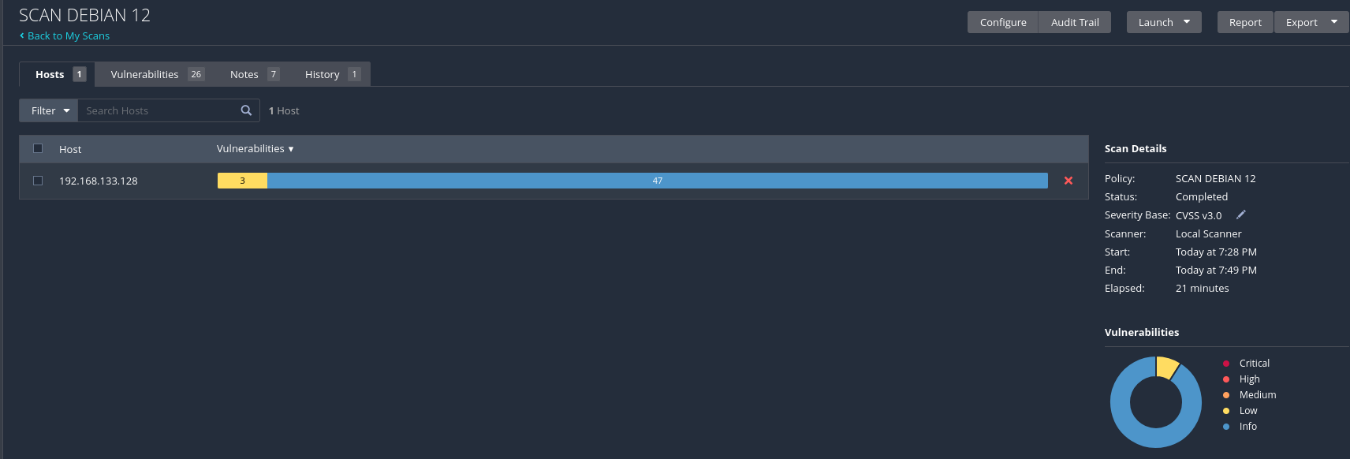
Le agregamos la ip o host que necesitamos auditar y le damos en save

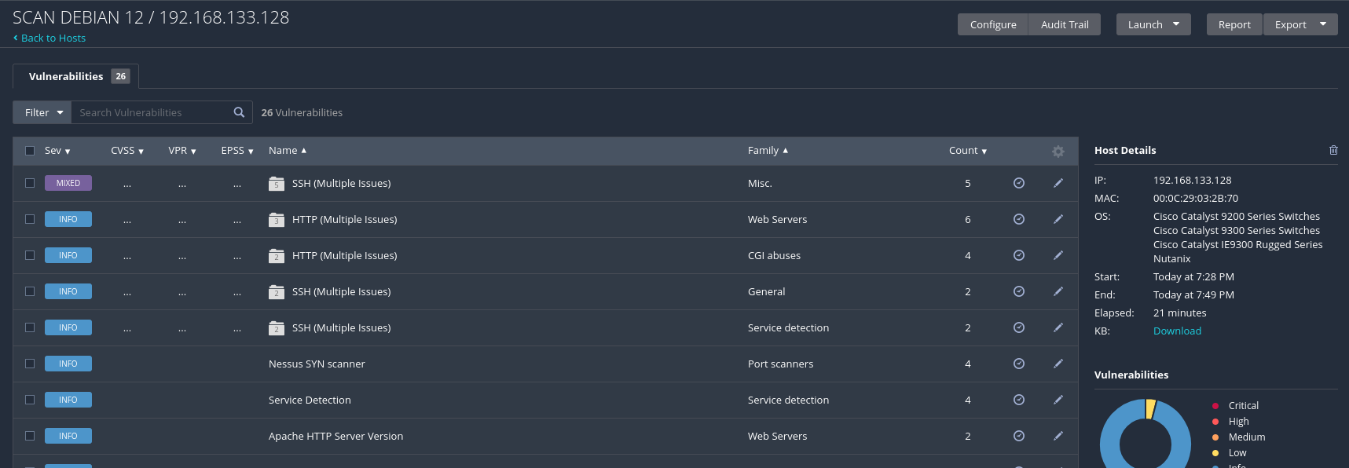
****

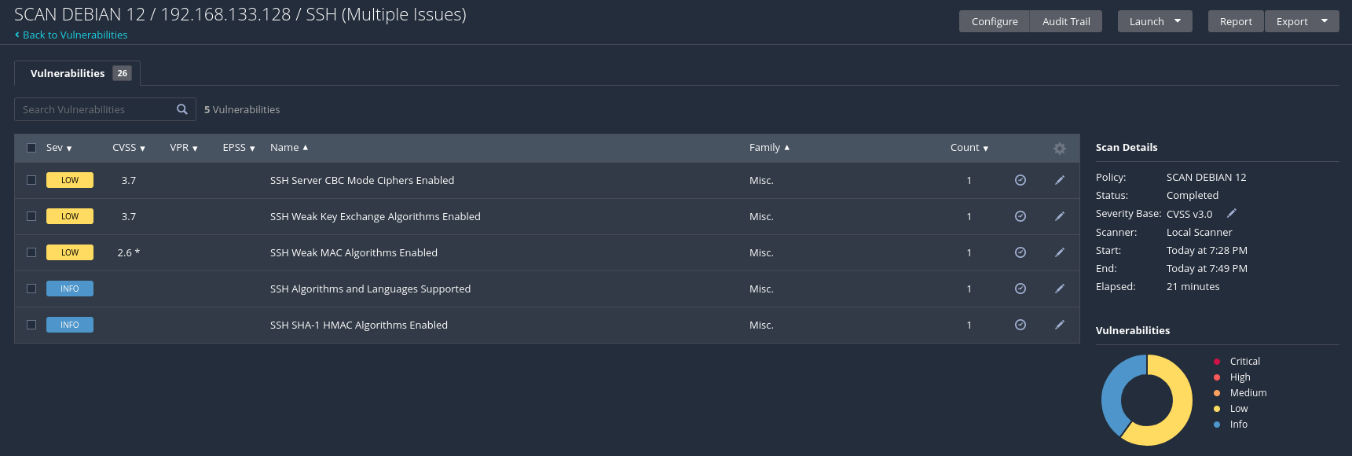
Le damos al boton de play y esperamos a que termine el scaneo

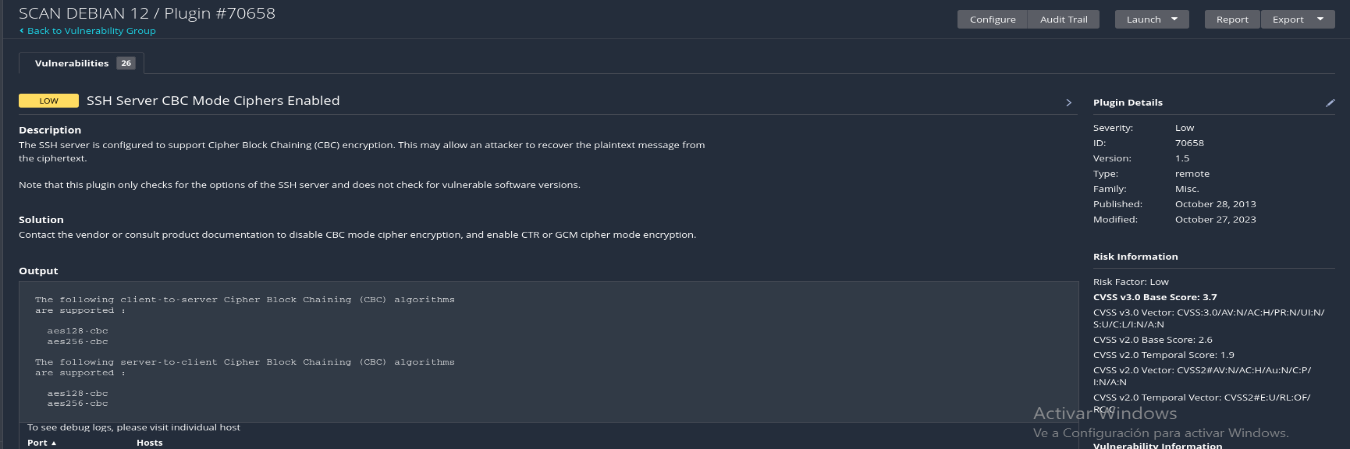
****

Y el resultado es el siguiente de nuestro scaneo:

****

****

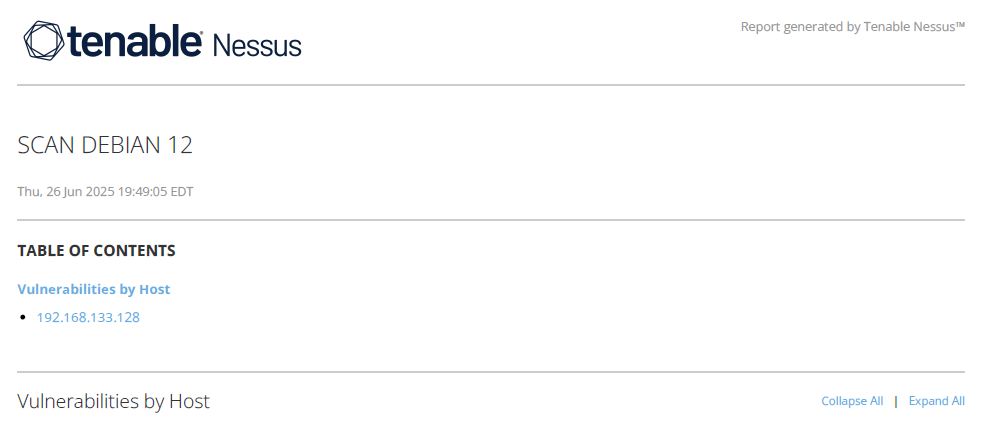
****

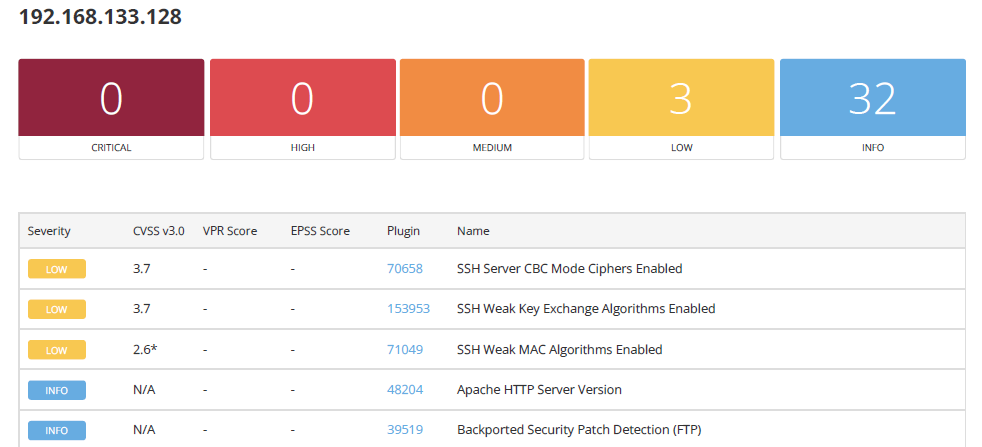
****

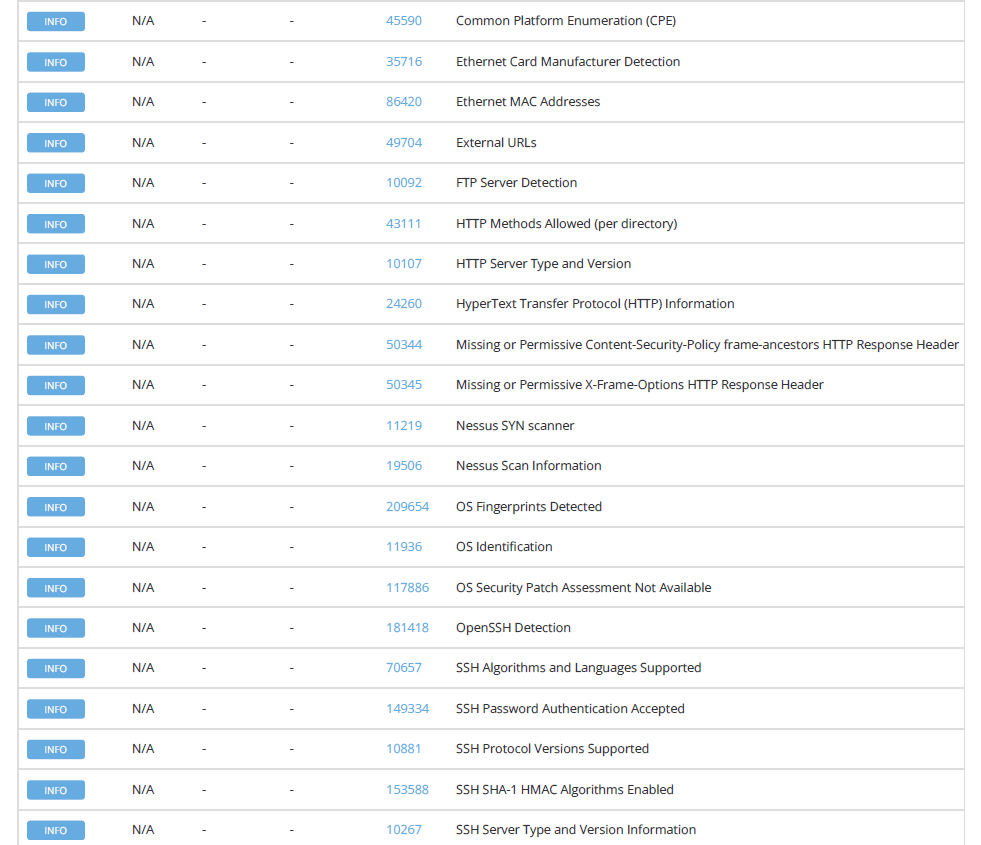
**Resumen detallado**

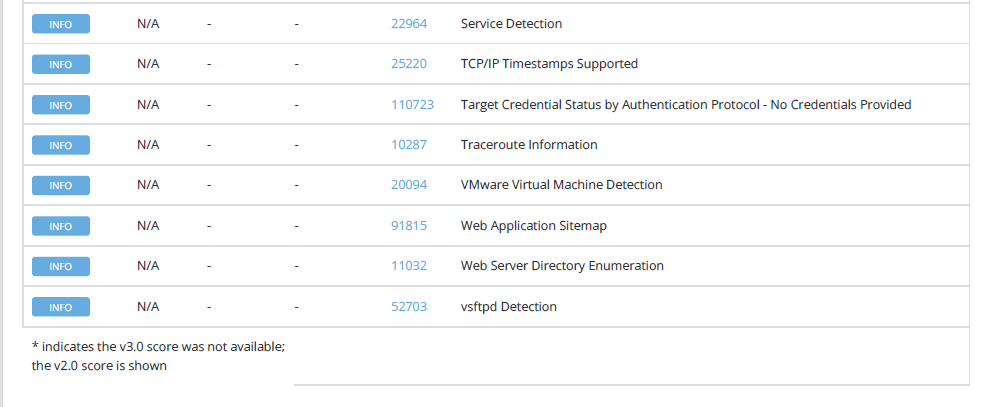
El escaneo realizado con Nessus detectó principalmente configuraciones inseguras en el servicio SSH y en las cabeceras HTTP de seguridad.

* En SSH, el servidor está utilizando cifrados CBC, algoritmos débiles para intercambio de claves y MACs poco seguros, lo que aumenta el riesgo de ataques de intermediarios (MitM) y puede comprometer la confidencialidad e integridad de la conexión.
* En los servicios web, faltan cabeceras de seguridad importantes (Content-Security-Policy y X-Frame-Options), lo que deja la puerta abierta a ataques como clickjacking, donde un atacante puede engañar a un usuario para que interactúe con contenido malicioso embebido en otro sitio.
* Aunque no se encontraron vulnerabilidades críticas, estas configuraciones representan riesgos conocidos que pueden ser mitigados con ajustes de configuración simples y efectivos.

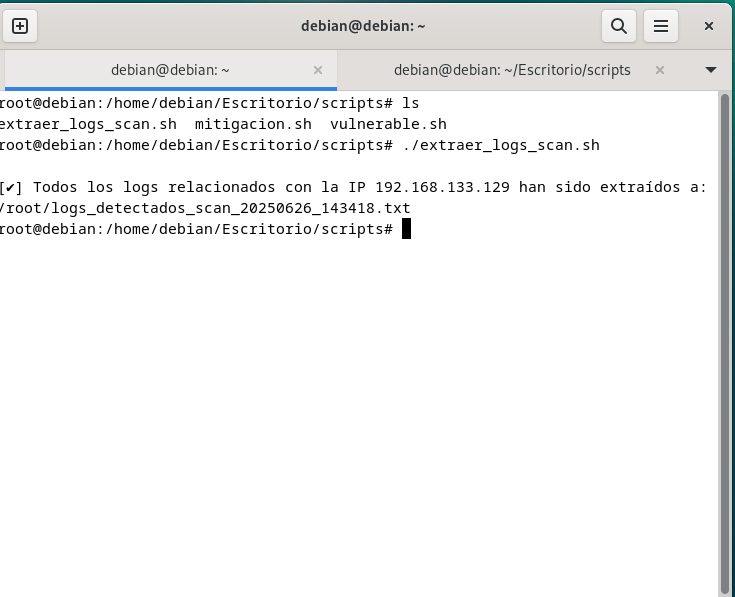


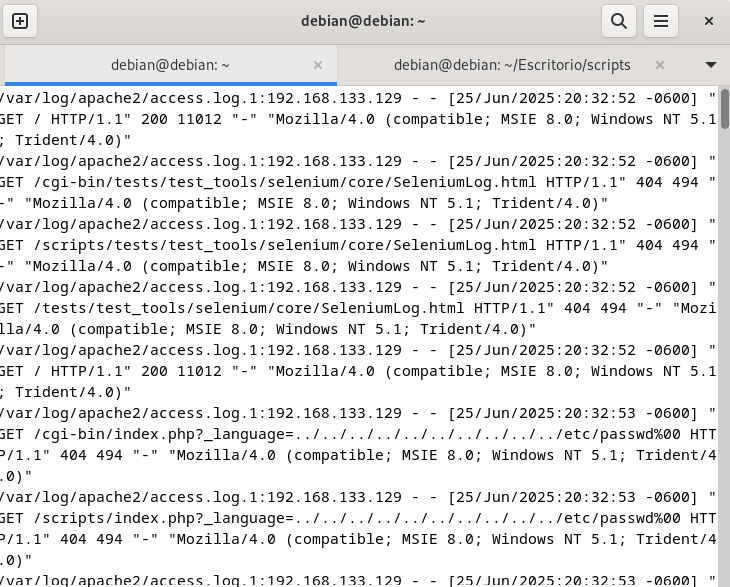






Asimismo despues de realizar los scaneos de nmap y nessus procedimos a realizar la extraccion de los logs en el sistema vulnerable.





Durante la auditoría de seguridad realizada a la máquina objetivo 192.168.133.128, se ejecutaron dos herramientas complementarias: Nmap y Nessus. Ambas permitieron identificar configuraciones inseguras, versiones de servicios y posibles vectores de ataque. A partir de los hallazgos se elaboró un análisis conjunto, unificando vulnerabilidades coincidentes y clasificándolas según su nivel de riesgo, impacto potencial y recomendaciones para su mitigación.

Se detectaron fallas de seguridad en servicios clave como **FTP**, **SSH** y **Apache HTTP/HTTPS**, destacando principalmente el uso de protocolos inseguros, cifrados obsoletos y configuraciones por defecto sin hardening. Este tipo de vulnerabilidades, aunque algunas sean consideradas de bajo nivel, representan puertas de entrada potenciales si se combinan con otros vectores o se dejan sin corregir.

El siguiente cuadro resume los hallazgos críticos, su análisis de impacto y las recomendaciones concretas para reducir la superficie de ataque del sistema.

# 5.Análisis Consolidado de Vulnerabilidades – Nmap y Nessus

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Puerto** | **Servicio** | **Vulnerabilidad / Hallazgo** | **Fuente** | **Severidad** | **Impacto** | **Recomendación** |
| 21/tcp | FTP (vsftpd 3.0.3) | Acceso anónimo habilitado y sin cifrado | Nmap | Alta | Posible acceso no autorizado, fuga o alteración de archivos, ataque de fuerza bruta. | Deshabilitar anonymous\_enable=NO, y habilitar SSL/TLS (ssl\_enable=YES) en vsftpd.conf. |
| 22/tcp | SSH (OpenSSH 9.2p1) | Cifrados CBC inseguros habilitados | Nessus | Baja (3.7) | Tráfico susceptible a ataques criptográficos (padding oracle). | Deshabilitar CBC y usar AES-GCM o ChaCha20 en sshd\_config. |
| 22/tcp | SSH | Algoritmos débiles de intercambio de claves (ej. diffie-hellman-group1-sha1) | Ambos | Baja (3.7) | Riesgo de ataque MITM, downgrade y comprometer el canal SSH. | Usar curve25519-sha256, diffie-hellman-group14-sha256. |
| 22/tcp | SSH | Algoritmos MAC débiles habilitados (ej. hmac-md5) | Nessus | Baja (2.6) | Posible manipulación de datos en el canal SSH. | Usar MACs modernos como hmac-sha2-256, hmac-sha2-512. |
| 80/tcp | HTTP (Apache 2.4.62) | Página por defecto activa (It works) | Nmap | Bajo–Medio | Indica falta de hardening y configuración insegura. | Reemplazar la página por contenido personalizado o restringir acceso. |
| 80/tcp | HTTP | Faltan cabeceras HTTP de seguridad (Content-Security-Policy, X-Frame-Options) | Nessus | Informativo | Vulnerabilidad a clickjacking, inyección de contenido. | Agregar cabeceras: Content-Security-Policy, X-Frame-Options. |
| 443/tcp | HTTPS (Apache 2.4.62) | Posible soporte de SSLv2/SSLv3 | Nmap | Media | Riesgo de ataque POODLE o MITM en conexiones inseguras. | Configurar solo TLSv1.2 y TLSv1.3, deshabilitar SSLv2/v3. |
| 443/tcp | HTTPS | Faltan cabeceras de seguridad (igual que HTTP) | Nessus | Informativo | Igual que en HTTP. | Igual que en HTTP. |

**Resumen del Análisis**

**Vulnerabilidades comunes en ambos escaneos:**

* **SSH – Algoritmos débiles**: Ambos detectan el uso de cifrados y algoritmos obsoletos en el servicio SSH, lo cual requiere ajustes inmediatos para evitar posibles ataques MITM y compromisos de cifrado.
* **Apache sin cabeceras de seguridad**: La falta de hardening y cabeceras HTTP fue detectada tanto en HTTP como en HTTPS.

**Vulnerabilidades críticas:**

* **Acceso FTP anónimo y sin cifrado**: La más crítica. Permite acceso sin autenticación con potencial de fuga o manipulación de archivos. Es esencial deshabilitar esta función y proteger con TLS.
* **Configuración insegura de SSL en HTTPS**: Usar SSLProtocol all puede habilitar versiones antiguas de SSL que son totalmente inseguras.

**Riesgos informativos o de bajo impacto (pero importantes a mitigar):**

* Página por defecto en Apache (It works).
* Ausencia de cabeceras de seguridad HTTP.

# 6.Recomendaciones Generales de Mitigación

|  |  |  |
| --- | --- | --- |
| **Servicio** | **Archivo a modificar** | **Acción recomendada** |
| **vsftpd (FTP)** | /etc/vsftpd.conf | anonymous\_enable=NOssl\_enable=YESConfigurar certificado TLS |
| **SSH** | /etc/ssh/sshd\_config | - Ciphers aes256-gcm@openssh.com,chacha20-poly1305@openssh.com- KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group14-sha256- MACs hmac-sha2-256,hmac-sha2-512 |
| **Apache (HTTP/HTTPS)** | /etc/apache2/sites-available/000-default.conf y/o .htaccess | - Agregar: Header always set Content-Security-Policy "frame-ancestors 'self'"Header always set X-Frame-Options "SAMEORIGIN"- Eliminar o reemplazar página por defecto |
| **SSL en Apache** | /etc/apache2/mods-available/ssl.conf | Asegurar que esté configurado:SSLProtocol all -SSLv2 -SSLv3SSLCipherSuite HIGH:!aNULL:!MD5 |

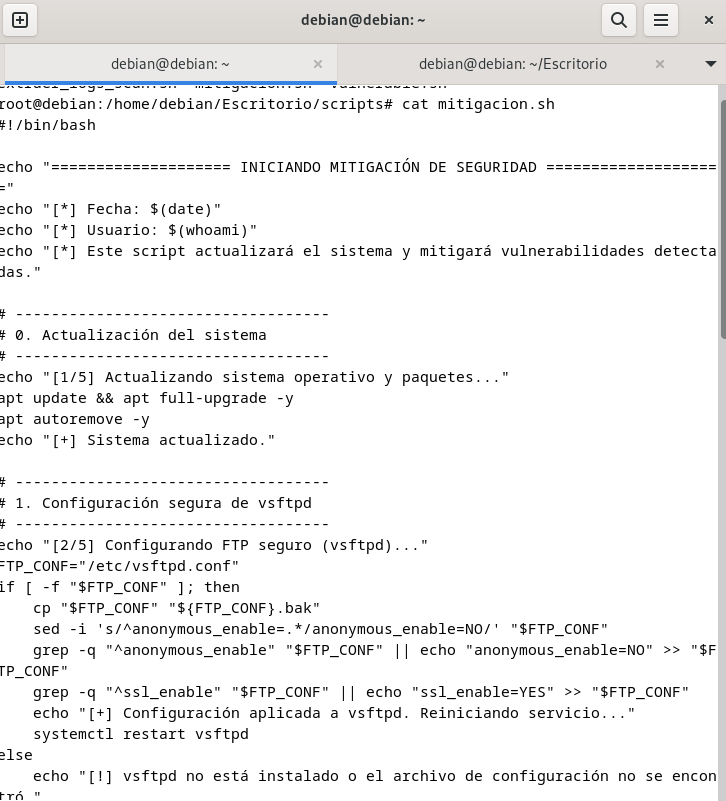
Para mitigar todas las vulnerabilidades detectadas, programé un script que automatiza las correcciones necesarias basándome en investigaciones sobre las mejores prácticas de seguridad. Primero, deshabilité el acceso anónimo en el servidor FTP y configuré el cifrado para proteger las credenciales y la información transmitida, ya que el acceso sin restricciones y el texto plano representan un riesgo alto.

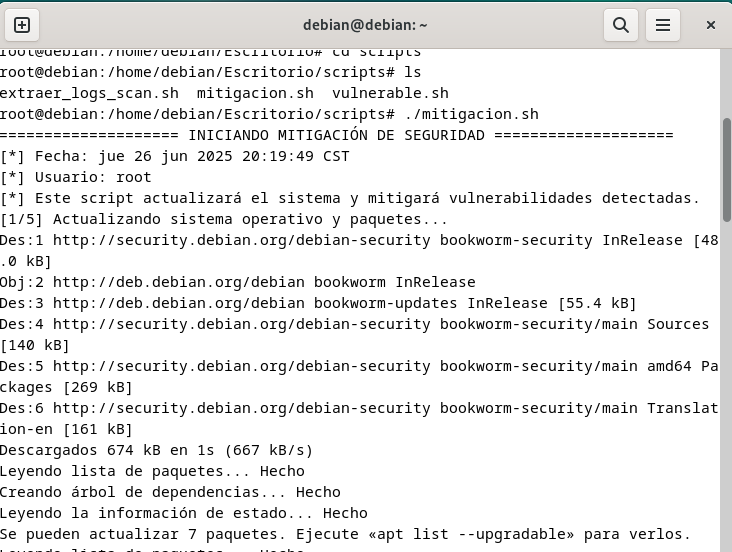
En el servicio SSH, identifiqué que se usaban algoritmos y cifrados obsoletos que pueden facilitar ataques tipo Man-in-the-Middle. Por eso, el script actualiza la configuración para que solo se usen cifrados modernos y seguros, como AES-GCM y curvas robustas para el intercambio de claves, reforzando la confidencialidad y la integridad de las conexiones.

Respecto al servidor Apache, eliminé la página por defecto para evitar exponer información innecesaria que pueda ser aprovechada por un atacante. Además, modifiqué la configuración SSL para deshabilitar los protocolos antiguos SSLv2 y SSLv3, dejando habilitados solo TLS 1.2 y TLS 1.3 con cifrados fuertes, lo cual previene ataques conocidos como POODLE.

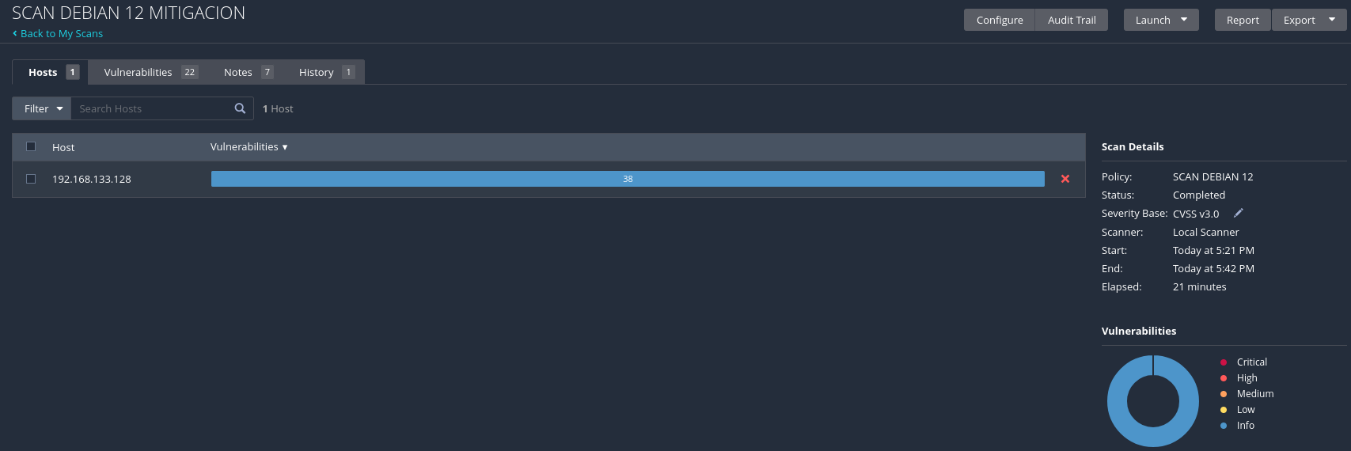
Por último, añadí cabeceras de seguridad HTTP como Content-Security-Policy y X-Frame-Options, con el fin de proteger contra ataques de clickjacking que pueden manipular la interacción del usuario con la página web.

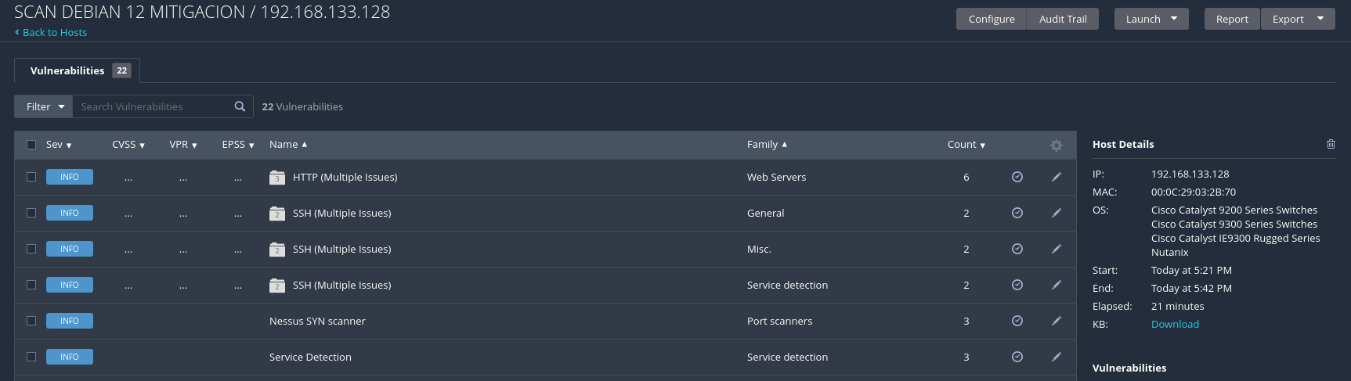
Con este script automatizado, garantizo que el sistema quede protegido de manera integral y preparado para pasar un nuevo escaneo sin que estas vulnerabilidades se repitan. Además, mantengo la recomendación constante de mantener el sistema actualizado para prevenir riesgos futuros. Dejo el script utilizado para realizar las mitigaciones y su debido analisis respectivo.

Como parte de las investigaciones realizadas se tomo recursos o conocimientos de los instructores de **udemy Santiago Hernandez y su curso completo de linux manejo de comando y shell script**, esto con la finalidad de aplicar un recurso automatizado para la mitigacion de vulnerabilidades ya que en entornos reales el factor tiempo es muy importante. Asi mismo se dejara en un drive el script implementado.

Se ejecuto el script previamente programado para verificar y mitigar todas las vulnerabilidades identificadas en el sistema operativo Debian 12, el cual fue escaneado utilizando las herramientas Nmap y Nessus. Este script automatiza la aplicación de medidas de seguridad basadas en las mejores prácticas, corrigiendo configuraciones inseguras detectadas en los servicios FTP, SSH, Apache HTTP/HTTPS, así como cabeceras web faltantes y protocolos criptográficos obsoletos. Con esto, se busca dejar el sistema reforzado y preparado para superar un nuevo escaneo sin que las vulnerabilidades previamente reportadas vuelvan a aparecer.









# 7.Resumen del laboratorio de mitigación de vulnerabilidades

Como parte del laboratorio, ejecuté un script en una máquina Debian para mitigar varias vulnerabilidades detectadas previamente con Nmap y Nessus. El script actualiza el sistema, refuerza la configuración de servicios como SSH, Apache y vsftpd, y activa el firewall UFW permitiendo solo los puertos necesarios (22, 80 y 443). Se aplicaron cabeceras de seguridad en Apache, se deshabilitó el acceso FTP anónimo y se reforzaron los cifrados de conexión segura.

Una vez aplicadas las configuraciones, realicé un nuevo escaneo completo con **Nmap** y **Nessus**, confirmando que ya no se detectan vulnerabilidades críticas ni servicios expuestos innecesarios. Los puertos visibles corresponden a servicios esenciales y ahora cuentan con configuraciones seguras.

En conclusión, las mitigaciones fueron exitosas y el sistema quedó reforzado, cumpliendo con los objetivos del laboratorio y se proporciona el siguiente link para reforzar el scaneo conteniendo los reportes realizados tanto del primer escaneo como del segundo ya con las vulnerabilidades resultas. <https://drive.google.com/drive/folders/1uieEyeF2oIqlZkWIot7fD73Gl_FuJcZ5?usp=sharing>

# 8. CONCLUSIONES

El laboratorio permitió reforzar habilidades en análisis y mitigación de vulnerabilidades. Mediante el uso combinado de herramientas de escaneo y automatización de medidas correctivas, se logró transformar un sistema vulnerable en un entorno seguro y preparado para auditorías técnicas.

# 9. GLOSARIO TÉCNICO

* Nessus: Herramienta de escaneo de vulnerabilidades desarrollada por Tenable.
* Nmap: Utilidad para el descubrimiento de hosts y servicios en una red.
* OpenVAS: Sistema de escaneo de vulnerabilidades de código abierto.
* Hardening: Proceso de asegurar un sistema mediante la reducción de vulnerabilidades.
* MitM (Man-in-the-Middle): Ataque en el que un tercero intercepta la comunicación entre dos partes.
* Vulnerability: Debilidad en un sistema que puede ser explotada.
* Patch: Actualización destinada a corregir errores o vulnerabilidades.
* Exploit: Código o técnica usada para aprovechar una vulnerabilidad.
* Penetration Testing (Pentest): Simulación de ataques para evaluar la seguridad de un sistema.
* Port Scanning: Técnica para identificar servicios activos en un host.
* SSH (Secure Shell): Protocolo para el acceso remoto seguro.
* FTP (File Transfer Protocol): Protocolo para la transferencia de archivos.
* Apache: Servidor web de código abierto.
* TLS (Transport Layer Security): Protocolo de cifrado para comunicaciones seguras.
* Firewall: Sistema que controla el tráfico de red entrante y saliente.
* MAC (Message Authentication Code): Técnica para verificar la integridad de los mensajes.
* Cipher: Algoritmo para cifrar y descifrar datos.
* Key Exchange: Proceso para establecer claves criptográficas compartidas.
* Clickjacking: Técnica de ataque que engaña al usuario para hacer clic en algo diferente.
* CIS Benchmark: Conjunto de buenas prácticas para asegurar configuraciones de sistemas.

# 10. FUENTES DE CONSULTA

* ISO/IEC 27001:2022
* CIS Benchmarks for Debian
* Nmap.org - Official Documentation
* Tenable.com - Nessus Documentation
* Udemy – Santiago Hernández: Curso completo de Linux
* IIRSI ACADEMIA SISAP