Proyecto final de Ciberseguridad

Objetivo: un servidor crítico comprometido – Linux Debian

Organización: 4Geeks Academy

Fecha: 7 julio 2025

Versión: 1.0

Report para:

4geeks entrega del proyecto final del bootcamp

Índice:

Fase 1: reconocimiento y recolección de evidencias

1. Objetivo del Análisis Forense

El objetivo de esta Fase 1 del análisis forense es realizar un reconocimiento inicial y la recopilación de evidencias para identificar los servicios comprometidos, el posible vector de acceso del atacante y proporcionar recomendaciones iniciales para bloquear la explotación, corregir vulnerabilidades y prevenir la escalada del ataque.

2. Metodología y Herramientas Utilizadas

El análisis se realizó de forma remota desde una máquina Kali Linux (IP: 192.168.56.101) utilizando las siguientes herramientas:

- **ip a**: Para verificar la configuración de la red local.
- **nmap**: Para escanear puertos, detectar servicios y versiones, y descubrir hosts en la red.
- **ftp (cliente)**: Para interactuar con el servicio FTP anónimo.
- **nikto**: Para analizar aplicaciones web en busca de vulnerabilidades.
- **gobuster**: Para enumerar directorios y archivos en servidores web.
- **Navegador web**: Para la inspección manual de recursos web.

3. Hallazgos y evidencia recopilada

El análisis remoto reveló la siguiente información sobre el servidor objetivo (192.168.56.102):

3.1. Configuración de la red objetivo

Dirección IP: 192.168.56.102

Dirección MAC: 08:00:27:3D:98:0C (NIC de VirtualBox) Sistema operativo detectado: Linux Debian (Kernel 4.X|5.X)

3.2. Servicios expuestos (basados en Nmap)

Puerto 21/tcp (FTP): vsftpd 3.0.3 (Estado: abierto)

Vulnerabilidad crítica: Se permite el inicio de sesión FTP anónimo.

Permisos: Se podía acceder a la raíz del sistema de archivos con el comando cd .. (acceder a la raíz del sistema de archivos). Sin embargo, los intentos de listar (ls -la) directorios sensibles como */var/www/html/ y /tmp/,* o de descargar (get) archivos como */etc/passwd*, registros de Apache o configuraciones *(/etc/apache2/apache2.conf*), resultaron en un error denegado o en un directorio vacío. No se pudo escribir en los directorios críticos.

Puerto 22/tcp (SSH): OpenSSH 9.2p1 Debian 2+deb12u3 (protocolo 2.0) (Estado: abierto)

Versión: Una versión relativamente reciente de OpenSSH, lo que indica una menor probabilidad de vulnerabilidades directas conocidas en el software. Puerto 80/tcp (HTTP): Apache httpd 2.4.62 (Debian) (Estado: abierto)

Página principal: Muestra la *"Página predeterminada de Apache2 Debian: Funciona"*.

WordPress identificado:

- **robots.txt** (*http://192.168.56.102/robots.txt*) contiene entradas relacionadas con WordPress (Deshabilitar: */wp-admin/,* Mapa del sitio: *http://localhost/wp-sitemap.xml*).
- Se accedió correctamente al archivo ***wp-login.php*** a través de un navegador web *(http://192.168.56.102/wp-login.php*), mostrando la página de inicio de sesión de WordPress.
- El directorio **/wp-content/uploads/** está configurado para permitir la indexación de directorios y se puede acceder a él a través de un navegador web (http://192.168.56.102/wp-content/uploads/). Se encontraron los subdirectorios 2024/ y 2025/.

Otras vulnerabilidades de configuración HTTP (Nikto):

- Falta de encabezados X-Frame-Options y X-Content-Type-Options.
- Fuga de inodos mediante ETags.
- Existencia de *xmlrpc.php* (históricamente objetivo de ataques).

3.3. Evidencia de compromiso y vector de acceso

Basado en evidencia recopilada remotamente

Vector de acceso más probable (compromiso inicial y persistencia):

- El análisis indica que el vector de acceso principal y la forma más probable en que el atacante estableció la persistencia es a través de una vulnerabilidad en la instalación de WordPress.
- El hallazgo crucial es que el directorio *http://192.168.56.102/wp-content/uploads/* está configurado para permitir el listado de directorios. Esta es una falla de seguridad grave que, combinada con una vulnerabilidad de carga de archivos (común en plugins o temas de WordPress obsoletos o con código deficiente) o una vulnerabilidad de credenciales, permitiría a un atacante cargar un shell web (por ejemplo, un archivo .php malicioso) en este directorio y ejecutarlo directamente a través del navegador.

Vulnerabilidad secundaria (Configuración insegura):

- El servicio FTP anónimo, capaz de navegar a la raíz del sistema de archivos (cd...), es una falla de seguridad grave. Si bien los permisos limitados de lectura y escritura en directorios críticos (como */var/www/html* y archivos del sistema) reducen la probabilidad de que se tratara de un vector directo para un *shell* web o una exfiltración masiva de datos, esta falla sigue representando un riesgo de reconocimiento y, junto con otras vulnerabilidades, podría haberse explotado de maneras más complejas.

3.4. Archivos y modificaciones sospechosos (Inferencia remota)

No fue posible inspeccionar directamente los archivos ni los procesos que se ejecutaban en el servidor.

Inferencia: Dado el acceso confirmado al directorio de cargas de WordPress, es muy probable que el atacante haya dejado uno o más shells web (por ejemplo, archivos .php con nombres inusuales) o puertas traseras dentro de la instalación de WordPress, especialmente en */wp-content/uploads/* o en directorios de plugins/temas comprometidos.

4. **Recomendaciones para bloquear la explotación y evitar su escalada (Acciones urgentes)**

Según análisis remotos, se **RECOMIENDAN URGENTEMENTE** las siguientes medidas para contener la explotación y evitar una mayor escalada:

Contención de WordPress:

- Desactivar inmediatamente la indexación de directorios: Modificar la configuración de Apache para el directorio */var/www/html/wp-content/uploads/* para evitar que el contenido se publique. Esto se puede hacer añadiendo "Options -Indexes" en un archivo *.htaccess* dentro del directorio */wp-content/uploads/* o en la configuración del host virtual de Apache.
- Deshabilitar temporalmente el acceso web: Si es posible, cierre el servicio Apache (*sudo systemctl stop apache2*) hasta que se realice una investigación más exhaustiva y se pueda solucionar el problema.

Contención de FTP:

- **Deshabilitar FTP anónimo**: Modifique la configuración de vsFTPd para deshabilitar completamente los inicios de sesión de usuarios anónimos.
- **Configuración de chroot**: Si el FTP anónimo es estrictamente necesario (lo cual es poco común en servidores de producción), configure chroot_local_user=YES para que el usuario anónimo no pueda navegar fuera de su directorio personal *(/srv/ftp* o similar).

5. **Recomendaciones para revertir cambios y corregir configuraciones (Acciones posteriores a la contención)**

Tras la contención inmediata, se **RECOMIENDAN** las siguientes acciones de erradicación y refuerzo para revertir los cambios del atacante y reforzar la seguridad del servidor.

Auditoría completa de la instalación de WordPress:

- Analice el directorio */var/www/html/wp-content/uploads/* y otros directorios de WordPress (plugins, temas) en busca de archivos .php extraños o archivos con fechas de modificación recientes sospechosas.
- Compare los archivos de la instalación actual de WordPress con versiones originales y limpias (núcleo, plugins y temas) para identificar modificaciones maliciosas.

- Elimine cualquier puerta trasera o shell web encontrado.
- Auditoría de usuarios y credenciales: Analice a los usuarios de WordPress en busca de cuentas desconocidas o no autorizadas. Restablezca las contraseñas de todos los usuarios (especialmente las de los administradores) a contraseñas seguras y únicas.
- Actualización de WordPress: Actualice WordPress, todos los plugins y temas a las últimas versiones.
- Eliminación de plugins/temas no utilizados: Desactive y elimine cualquier plugin o tema que no esté en uso.

Refuerzo del servidor Apache/HTTP:

- **Implemente encabezados** de seguridad HTTP como X-Frame-Options y X-Content-Type-Options en la configuración de Apache.
- **Desactive las fugas** de inodos mediante ETags.
- **Restringa el acceso** a xmlrpc.php o desactívelo si no se utiliza.

Refuerzo general del servidor:

- **Auditoría de registros**: Realice una revisión detallada de los registros del sistema (p. ej., */var/log/auth.log* para SSH, registros de Apache) para detectar actividad inusual o inicios de sesión no autorizados.
- **Auditoría de usuarios y grupos del sistema**: Compruebe la presencia de nuevos usuarios o grupos no autorizados.
- **Análisis de rootkits y malware**: Ejecute herramientas de análisis de rootkits (p. ej., chkrootkit, rkhunter) y antivirus (p. ej., ClamAV) en el servidor para detectar y eliminar malware.
- **Actualizaciones de paquetes**: Asegúrese de que todos los paquetes del sistema operativo estén actualizados (apt update y apt upgrade).
- **Cortafuegos**: Configure o refuerce las reglas del cortafuegos (p. ej., ufw, iptables) para permitir únicamente el tráfico esencial.

6. Recomendaciones para prevenir futuros ataques

Para reforzar la seguridad y prevenir futuros ataques similares, se proponen las siguientes recomendaciones:

- **Política de contraseñas seguras**: Exija el uso de contraseñas complejas y únicas para todos los usuarios (sistema, SSH, WordPress, base de datos).
- **Autenticación con clave SSH**: Desactive el inicio de sesión con contraseña para SSH y utilice únicamente la autenticación con clave.
- **Principio de privilegios mínimos**: Asegúrese de que todos los servicios y usuarios operen con los privilegios mínimos necesarios para sus funciones.
- **Copia de seguridad y recuperación**: Implemente una rutina robusta de copias de seguridad periódicas y comprobadas.
- **Monitoreo y alertas**: Configure sistemas de monitoreo de integridad de archivos (FIM) y análisis de registros (SIEM) para detectar actividad sospechosa en tiempo real.
- **Seguridad de aplicaciones web**:
    - Utilice un cortafuegos de aplicaciones web (WAF). Realizar pruebas de seguridad periódicas de las aplicaciones web (pruebas de penetración, análisis de vulnerabilidades).
    - Mantener la higiene del código e implementar parches de seguridad de forma oportuna.
- **Capacitación de concientización**: Educar a usuarios y administradores sobre las mejores prácticas.

Fase 2: detecta y corrige una vulnerabilidad diferente

1. Objetivo

El objetivo de este análisis forense de la Fase 2 fue escanear, detectar y explotar una vulnerabilidad de seguridad diferente a la identificada en la Fase 1 (vulneración a través de WordPress) y documentar el proceso de explotación y las medidas de remediación aplicadas.

2. Metodología y Herramientas Utilizadas

El análisis se realizó en una máquina Kali Linux (IP: 192.168.56.101) y, tras la vulneración, directamente en el mismo servidor.

Las herramientas utilizadas incluyen:

- **nmap**: Para abrir puertos, detectar servicios y versiones. (Resultados reevaluados de la Fase 1).
- **hydra**: Herramienta de fuerza bruta para descifrar credenciales. Aunque hemos utilizado otras herramientas, esta la nombramos y la he usado para dar más viabilidad a otras herramientas que no sean las comunes y conocidas.
- **ssh (cliente):** Para acceder al servidor mediante Secure Shell.
- **Comandos básicos de Linux:**
  - ls
  - grep
  - ps
  - crontab
  - cat
  - chmod
  - echo
  - nano
  - passwd
  - systemctl: Para la investigación interna del sistema y la aplicación de parches.
- **Navegador web**: Para el análisis posterior a la aplicación de parches.

3. Descubrimiento y explotación de nuevas vulnerabilidades

3.1. Análisis de reevaluación completa del sistema (Nmap)

Según las configuraciones iniciales de Nmap en la Fase 1, los servicios abiertos en

192.168.56.102 eran: FTP (puerto 21), SSH (puerto 22) y HTTP (puerto 80). Para la Fase 2, nos centramos en vulnerabilidades no directamente relacionadas con Apache/WordPress.

Los servicios SSH en el puerto 22 y FTP en el puerto 21 también son de interés.

3.2. Detección y explotación de vulnerabilidades (SSH - Credenciales defectuosas)

Vulnerabilidades detectadas: El servicio SSH (OpenSSH 9.2p1) no permitía la autenticación del servidor, y el usuario root tenía una contraseña extremadamente débil y fácil de comprobar.

Proceso de explotación:

Utilizamos la herramienta Hydra para realizar un ataque de fuerza bruta contra el servicio SSH en el puerto 22. Se empleó una combinación de listas de usuarios:

- *(/usr/share/nmap/nselib/data/usernames.lst*)
- mensajes */usr/share/wordlists/fasttrack.txt*)

El ataque de fuerza bruta se produjo rápidamente, identificando las credenciales de inicio de sesión del usuario ***root*** como *root:123456*.

```
┌──(kali㉿kali)-[~]
└─$ hydra -L /usr/share/nmap/nselib/data/usernames.lst -P
/usr/share/wordlists/fasttrack.txt
ssh://192.168.56.102
[22][ssh] host: 192.168.56.102 login: root password: 123456
(...)
```

Con las credenciales obtenidas, fue posible establecer una conexión SSH como usuario root con el otro servidor, lo que otorgó acceso total de administrador al sistema.

```
ssh root@192.168.56.102
Contraseña: 123456
root@debian:~# whoami
root
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root)
```

3.3. Documentación interna del sistema y evidencia adicional

Tras obtener acceso *root* al servidor, se realizó una investigación interna para identificar otros indicadores de vulnerabilidad o vulneraciones de seguridad.

Permisos inseguros en el directorio UPLOADS de WordPress:

Se inspeccionó el directorio */var/www/html/wp-content/uploads/* (confirmado en la Fase 1 como vulnerable a la indexación de directorios), revelando permisos drwxrwxrwx (777). Este permiso autoriza a cualquier usuario del sistema a leer, guardar y ejecutar archivos en este directorio, o bien, a detectar una falla de seguridad grave que facilita la carga de shells web.

```
root@debian:~# ls -la /var/www/html/wp-content/uploads/
total 16
drwxrwxrwx 4 www-data www-data 4096 2 jul. 2024 14:45
drwxrwxrwx 5 www-data www-data 4096 2 jul. 2024 14:45
drwxrwxrwx 4 www-data www-data 4096 8 oct. 2024 2024
drwxrwxrwx 3 www-data www-data 4096 2 jul. 2024 14/02/2025 14:45
```

Registros de autenticación faltantes (auth.log):

Intentar acceder a */var/log/auth.log* resultará en el error "No existe el archivo o directorio". La ausencia de este registro es altamente sospechosa y sugiere que el atacante anterior podría haber eliminado o configurado el sistema para que no registre eventos de autenticación con el fin de ocultar sus actividades.

```
root@debian:~# grep "sshd" /var/log/auth.log
grep: /var/log/auth.log: No existe el archivo o directorio
```

Procesos en ejecución (ps aux):

El análisis de dos procesos en ejecución no revela procesos anormales ni maliciosos en tiempo real. Los procesos observados son consistentes con un sistema Debian estándar que ejecuta Apache, MySQL, SSH y servicios del sistema.

```
root@debian:~# ps aux
(...) (Salida completa de ps aux) (...)
```

Tareas programadas (Tareas cron):

No se encontraron trabajos cron inusuales ni sospechosos para el usuario *root* ni a nivel de sistema *(/etc/crontab, /etc/cron.*)*. Esto indica que no hay persistencia activa evidente a través de trabajos cron.

```
root@debian:~# crontab -l
no crontab para root
root@debian:~# cat /etc/crontab
(...) (Salida estándar de crontab) (...)
root@debian:~# ls -la /etc/cron.*
(...) (Salida estándar de directorios de cron) (...)
```

robots.txt Inconsistente:

Aunque se puede acceder a *http://192.168.56.102/robots.txt* mediante un navegador, no se encontró el archivo */var/www/html/robots.txt* en el sistema de archivos, lo que indica

una configuración inusual de Apache o el uso de un archivo robots.txt dinámico o un directorio diferente.

4. Medidas aplicadas para remediar la vulnerabilidad

Tras la detección y explotación de la vulnerabilidad de credenciales SSH débiles y la investigación interna, se aplicaron las siguientes medidas de remediación directamente al servidor.

Cambio de la contraseña del usuario root (SSH):

La contraseña del usuario *root* se cambió inmediatamente a una contraseña segura y compleja para evitar futuros accesos no autorizados vía SSH con la contraseña anterior.

```
root@debian:~# passwd root
Nueva contraseña: [new_strong_password]
Reescriba la nueva contraseña: [new_strong_password]
passwd: contraseña actualizada correctamente
```

Corregir los permisos del directorio wp-content/uploads/ (WordPress):

Para mitigar el riesgo de carga y ejecución de web shell, los permisos del directorio */var/www/html/wp-content/uploads/* y sus subdirectorios se cambiaron de 777 a 755.

Además, se deshabilitó el listado de directorios mediante *.htaccess*.

```
root@debian:~# echo "Options -Indexes" > /var/www/html/wp-content/uploads/.htaccess
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/2024/
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/2025/
```

Verificación del navegador:

Tras la solicitud, se confirmó que *http://192.168.56.102/wp-content/uploads/* ahora devuelve un error "403 Prohibido", lo que indica que el listado de directorios se ha deshabilitado correctamente.

Desactivación del acceso FTP anónimo:

El inicio de sesión FTP anónimo, una vulnerabilidad predeterminada, se ha desactivado para evitar cualquier reconocimiento o posible explotación de sus capacidades de navegación.

```
root@debian:~# nano /etc/vsftpd.conf
# (Se modificó 'anonymous_enable=YES' a 'anonymous_enable=NO')
root@debian:~# systemctl restart vsftpd
```

```
Comprobación de Kali:
```

Se confirmó que el inicio de sesión anónimo a través del cliente FTP de Kali ya no está permitido.

5. Conclusión y recomendaciones adicionales

La fase 2 demostró la detección y explotación exitosa de una vulnerabilidad de credenciales débiles en el servicio SSH, que otorgaba acceso *root* al servidor. Esta falla, junto con el exceso de permisos en el directorio de subidas de WordPress, representó un riesgo de seguridad significativo para el sistema. Las acciones de remediación implementadas solucionaron estas vulnerabilidades directas.

Recomendaciones adicionales de prevención y refuerzo (a considerar en un plan de seguridad a largo plazo).

Refuerzo de SSH:

- Desactive el inicio de sesión *root* directo mediante SSH en el archivo */etc/ssh/sshd_config*.
- Configure la autenticación basada en claves SSH para todos los usuarios.
- Implemente fail2ban o una herramienta similar para bloquear las IP después de varios intentos fallidos de inicio de sesión por SSH.

Refuerzo de WordPress:

- Realice una auditoría completa de seguridad de WordPress para identificar y eliminar cualquier puerta trasera, *shell* web o archivo modificado (aunque no se encuentre en el momento, los permisos 777 indican la posibilidad).
- Asegúrese de que WordPress, todos los plugins y temas estén siempre actualizados a las últimas versiones.
- Revise y refuerce todas las contraseñas de usuario de WordPress.
- Implemente un firewall de aplicaciones web (WAF) para proteger la aplicación.

Seguridad de registros:

- Investigue la causa de la ausencia de */var/log/auth.log* y restaure o configure correctamente los registros de seguridad del sistema para garantizar una auditoría adecuada de eventos futuros.

Principio de privilegios mínimos:

- Revise los permisos de archivos y directorios en todo el servidor, garantizando que los servicios y usuarios operen con los privilegios mínimos necesarios.

Monitoreo:

- Implemente soluciones de monitoreo continuo para detectar actividad inusual, inicios de sesión sospechosos y cambios en los archivos.

Fase 3: Plan de respuesta de incidentes y certificación

1. Objetivo

El objetivo de la Fase 3 es ir más allá de la respuesta reactiva a incidentes (abordada en las Fases 1 y 2) para establecer una postura de seguridad proactiva y resiliente. Esto implica desarrollar un plan de respuesta a incidentes sólido basado en las directrices NIST SP 800-61 y desarrollar un Sistema de Gestión de Seguridad de la Información (SGSI) conforme a la norma ISO 27001, con el objetivo de proteger continuamente los activos de información de la organización.

2. Metodología

Esta fase se desarrolló con base en las mejores prácticas reconocidas mundialmente en seguridad de la información:

NIST SP 800-61 Rev. 2 (Guía de Manejo de Incidentes de Seguridad Informática): Se utiliza como marco para la estructura y las fases del Plan de Respuesta a Incidentes (PRI).

ISO/IEC 27001:2022 (Sistemas de Gestión de Seguridad de la Información - Requisitos):

Base para el desarrollo del SGSI, que abarca el análisis de riesgos, las políticas de seguridad y la selección de controles.

ISO/IEC 27002:2022 (Controles de Seguridad de la Información): Guía para la implementación de controles de seguridad que respaldan la norma ISO 27001.

3. Plan de Respuesta a Incidentes (PRI) - Según NIST SP 800-61

El Plan de Respuesta a Incidentes detalla los pasos que la organización debe seguir antes, durante y después de un incidente de seguridad.

3.1. Fases del Plan de Respuesta a Incidentes

3.1.1. Preparación

Política de Respuesta a Incidentes: Desarrollar y aprobar una política formal de respuesta a incidentes, articulando el compromiso de la alta dirección.

Equipo de Respuesta a Incidentes (CSIRT/CIRC): Formalizar la formación de un equipo dedicado con roles y responsabilidades claros (p. ej., Gestor de Incidentes, Analistas de Seguridad, Especialistas en Redes, Relaciones Públicas, Departamento Legal).

Infraestructura y Herramientas: Implementar y configurar herramientas como SIEM (para la agregación y análisis de registros), EDR (para la detección y respuesta de endpoints), NIDS/NIPS (para la monitorización de la red), Firewall de Aplicaciones Web (WAF) para la protección de aplicaciones web y herramientas forenses.

Documentación y Procedimientos: Crear y mantener manuales de estrategias para incidentes comunes (p. ej., ataque de fuerza bruta SSH, vulnerabilidad de WordPress, detección de malware), listas de verificación y procedimientos operativos estándar (POE).

Capacitación y Concientización: Realizar capacitaciones periódicas para el personal del CSIRT y sesiones de concientización sobre seguridad para todos los empleados sobre la identificación y el reporte de actividades sospechosas.

Comunicación: Establecer un plan de comunicación interno (personal, gerencia) y externo (clientes, medios de comunicación, reguladores, socios, fuerzas del orden) para diferentes escenarios de incidentes.

Pruebas: Realizar simulacros de incidentes (p. ej., ejercicios de simulación, recorridos) y pruebas prácticas (p. ej., equipo rojo vs. equipo azul) anualmente para validar la eficacia del plan.

3.1.2. Detección y análisis

Fuentes de detección: Supervisar activamente los registros de autenticación (SSH, sistemas, aplicaciones), los registros de Apache/WordPress, las alertas de las herramientas de seguridad (SIEM, WAF, EDR), los informes de usuarios y la inteligencia de amenazas.

Triaje y clasificación: Una vez detectado un evento, el equipo debe determinar rápidamente si se trata de un incidente real, clasificarlo (p. ej., ataque a una aplicación web, acceso no autorizado, malware) y evaluar su gravedad e impacto potencial.

Análisis del alcance: Identificar qué sistemas, datos y servicios se vieron afectados, el vector inicial de la vulneración y el alcance de la intrusión.

Documentación inicial: Registrar fecha, hora, tipo de incidente, sistemas involucrados y acciones iniciales.

3.1.3. Contención, erradicación y recuperación

Contención:

Corto plazo: Aislar los sistemas comprometidos (p. ej., desconectarlos de la red, detener servicios como Apache/SSH), bloquear las IP maliciosas en el firewall y suspender las cuentas de usuario comprometidas.

Largo plazo: Implementar soluciones temporales para mantener la continuidad del negocio mientras se lleva a cabo la erradicación.

Erradicación:

Eliminación de la causa raíz: Eliminar el malware y corregir las vulnerabilidades (p. ej., aplicar parches de software, corregir configuraciones como los permisos 777 en los directorios de carga, cambiar contraseñas débiles).

Análisis forense: Realizar un análisis forense detallado para garantizar la eliminación completa del atacante y sus artefactos.

Recuperación:

Restauración: Restaurar los datos y los sistemas a partir de copias de seguridad limpias y fiables (asegurándose de que no contengan la amenaza).

Pruebas: Realice pruebas rigurosas en los sistemas recuperados para garantizar su funcionalidad y la ausencia de reinfecciones.

Monitoreo: Incremente el monitoreo. o de los sistemas recuperados durante un período antes de volver a la operación normal.

3.1.4. Actividades posteriores al incidente

Lecciones aprendidas: Realizar una reunión formal de análisis posterior al incidente para revisarlo, identificar los aspectos que funcionaron bien y las áreas de mejora en el proceso de respuesta.

Informe posterior al incidente: Documentar un informe detallado del incidente, incluyendo el cronograma, las medidas adoptadas, los resultados, el impacto y las recomendaciones para evitar su recurrencia.

Actualización del proceso: Revisar y actualizar las políticas, los procedimientos y los manuales de respuesta a incidentes con base en las lecciones aprendidas.

Fortalecimiento continuo: Implementar las acciones correctivas y preventivas identificadas (p. ej., nuevos controles de seguridad, capacitación adicional).

3.2. Respuesta a un ataque similar a un hackeo (Ejemplos de Fases 1 y 2)

Considerando credenciales SSH débiles y vulnerabilidades de WordPress/ataques de permisos excesivos, la organización respondería de la siguiente manera:

Detección:

Alerta de fuerza bruta SSH: SIEM o fail2ban (si se implementa) generaría alertas para múltiples intentos fallidos de inicio de sesión SSH desde la misma IP.

Alerta de inicio de sesión root directo: SIEM alertaría de cualquier inicio de sesión root exitoso a través de SSH, especialmente si proviene de una fuente no autorizada o mediante contraseña.

Alerta de actividad de WordPress: WAF o Monitoreo de integridad de archivos (FIM) detectaría intentos de subir archivos maliciosos a /wp-content/uploads/ o cambios no autorizados en archivos críticos de WordPress.

Contención:

SSH: fail2ban bloquearía automáticamente la IP del atacante. El equipo de respuesta cambiaría inmediatamente la contraseña del usuario root e idealmente deshabilitaría el inicio de sesión root mediante SSH.

WordPress: WAF bloquearía la carga maliciosa. Si el ataque persistía, el directorio /wp-content/uploads/ se convertiría en de solo lectura o el servicio Apache se detendría temporalmente.

Erradicación:

SSH: Implementar la autenticación con clave SSH y deshabilitar por completo la autenticación con contraseña para el usuario root.

WordPress: Eliminar cualquier archivo sospechoso subido, corregir los permisos del directorio (chmod 755 para subidas), deshabilitar la indexación de directorios mediante .htaccess, actualizar WordPress, los plugins y los temas a las últimas versiones.

FTPS: Deshabilitar el acceso FTP anónimo y eliminar la posibilidad de navegar por directorios.

Limpieza de registros: Si se confirma la ausencia de registros, se iniciará un proceso para restaurar la funcionalidad de registro y una investigación para determinar la causa de la eliminación o modificación del registro.

Prevención de recurrencias:

Políticas de contraseñas: Implementar políticas de contraseñas robustas en toda la organización y forzar el cambio a contraseñas robustas.

MFA: Implementar la autenticación multifactor para todos los accesos administrativos (SSH, WordPress).

Refuerzo: Aplicar reglas de refuerzo a los servidores (p. ej., pruebas de rendimiento de seguridad), incluyendo la eliminación de servicios innecesarios y la configuración de firewalls.

Gestión de vulnerabilidades: Realizar análisis de vulnerabilidades y pruebas de penetración con regularidad.

Actualización continua: Mantener todos los sistemas y aplicaciones (especialmente WordPress) siempre actualizados.

Monitorización activa: Mantener SIEM y otras herramientas de monitorización activas y ajustadas para detectar anomalías.

4. Mecanismos de protección de datos

Para proteger la información crítica de la empresa, los siguientes mecanismos de protección de datos son esenciales:

4.1. Copias de seguridad periódicas y comprobadas

Frecuencia y retención: Implementar una política de copias de seguridad granulares: copias de seguridad diarias para datos críticos (RPO bajo), semanales para datos menos sensibles y copias de seguridad completas mensuales. Retención de 7, 30 y 90 días, respectivamente.

Ubicación: Almacenar las copias de seguridad en múltiples ubicaciones: local (para una recuperación rápida) y remota (en la nube o en cinta física en una bóveda segura) para protegerse contra desastres locales.

Aislamiento: Las copias de seguridad deben estar aisladas lógica o físicamente de la red de producción para protegerlas contra ataques de ransomware.

Pruebas de restauración: Realice pruebas de restauración mensuales de las copias de seguridad para garantizar su integridad, funcionalidad y tiempo de recuperación (RTO).

4.2. Cifrado de datos confidenciales

Datos en tránsito: Aplique estrictamente el uso de TLS 1.2+ (Seguridad de la capa de transporte) en todas las comunicaciones de red (HTTPS para aplicaciones web, SSH para acceso remoto, VPN para acceso a la red interna).

Datos en reposo: Cifrado de disco completo (FDE): Para todos los servidores y estaciones de trabajo que almacenan datos confidenciales (p. ej., LUKS para Linux).

Cifrado a nivel de aplicación/base de datos: Para información altamente confidencial (p. ej., datos de clientes, información empresarial, financieros, etc.) almacenados en bases de datos, mediante cifrado de columnas o TDE (Cifrado Transparente de Datos).

4.3. Implementación de Controles de Acceso Estrictos

Principio de Mínimo Privilegio (PoLP): Todos los usuarios, sistemas y procesos deben tener solo los permisos mínimos necesarios para realizar sus funciones.

Autenticación Multifactor (MFA): Exigir el uso de MFA para todos los accesos administrativos (SSH, administración de WordPress, paneles de control) e, idealmente, para todos los usuarios.

Contraseñas Seguras: Implementar políticas de contraseñas complejas, con una longitud mínima (más de 12 caracteres), uso de caracteres mixtos y rotación periódica.

Gestión de Identidad y Acceso (IAM): Utilizar un sistema centralizado para aprovisionar, administrar y desaprovisionar usuarios y sus permisos.

Segregación de Funciones (SoD): Asegurarse de que las tareas críticas requieran la participación de varias personas, evitando que una sola persona abuse de sus privilegios.

Revisiones de acceso: Realizar revisiones trimestrales de los permisos de acceso para garantizar que estén actualizados y alineados con las responsabilidades de los usuarios.

5. Implementación de un SGSI (Sistema de Gestión de Seguridad de la Información) conforme a la norma ISO 27001

Un SGSI conforme a la norma ISO 27001 es un marco sistemático para la gestión de la seguridad de la información, que opera bajo el ciclo Planificar-Hacer-Verificar-Actuar (PDCA).

5.1. Análisis de riesgos (ISO 27001 - Cláusula 6.1.2)

Proceso: La organización debe establecer y mantener un proceso para evaluar los riesgos de seguridad de la información.

Pasos:

- **Identificación de activos**: Crear un inventario completo de todos los activos de información (hardware, software, datos, servicios, personas), definiendo a sus propietarios y su valor para el negocio.
- **Identificación de amenazas**: Enumerar las posibles amenazas a los activos (p. ej., ciberataques, fallos de hardware, errores humanos, desastres naturales).
- **Identificación de Vulnerabilidades**: Identificar las debilidades internas y externas que podrían ser explotadas por amenazas (p. ej., software obsoleto, falta de capacitación, credenciales deficientes, configuraciones incorrectas).
- **Evaluación de Riesgos**: Analizar la probabilidad de que una amenaza explote una vulnerabilidad y el impacto potencial (financiero, reputacional, operativo, legal) resultante. Calcular el nivel de riesgo.
- **Tratamiento de Riesgos**: Definir e implementar las opciones de tratamiento de riesgos (mitigar, transferir, evitar, aceptar) con base en los niveles de riesgo evaluados y la tolerancia al riesgo de la organización.
- **Declaración de Aplicabilidad (DdA):** Desarrollar una DdA que documente los controles ISO 27002 seleccionados para su implementación (y las justificaciones para su inclusión o exclusión) con base en el análisis de riesgos.

5.2. Definición de Políticas de Seguridad (ISO 27001 - Cláusula 5.2)

Las políticas de seguridad son documentos formales que guían las acciones y decisiones de seguridad en toda la organización. Deben ser aprobadas por la alta dirección y comunicadas a todos los empleados relevantes.

Ejemplos de Políticas Clave:

- **Política Maestra de Seguridad de la Información**: Resumen y objetivos generales.
- **Política de Control de Acceso**: Cómo se otorga, gestiona y revoca el acceso a los sistemas y la información.
- **Política de Uso Aceptable de Activos**: Directrices para el uso de los equipos y recursos de la empresa.
- **Política de Seguridad de la Red**: Reglas para configurar y proteger la infraestructura de red.
- **Política de Copias de Seguridad y Recuperación**: Detalles sobre la frecuencia, el tipo y el almacenamiento de las copias de seguridad.
- **Política de Gestión de Vulnerabilidades y Parches**: Cómo se identifican, evalúan y solucionan las vulnerabilidades.
- **Política de Respuesta a Incidentes**: Marco para la gestión de incidentes de seguridad.
- **Política de Escritorio y Pantalla despejados**: Para proteger la información confidencial en las áreas de trabajo.

5.3. Planes de acción para proteger la información crítica del negocio (basados en los controles de la norma ISO 27002)

Con base en las políticas y el análisis de riesgos, se desarrollarán planes de acción detallados para implementar controles específicos de la norma ISO 27002.

A. Controles organizacionales

Establecer una gobernanza de seguridad, incluyendo responsabilidades claras y un comité de seguridad.

Implementar procesos de concientización y capacitación en seguridad.

B. Controles de personal

Definir términos y condiciones laborales que aborden la seguridad de la información.

Realizar verificaciones de antecedentes para los nuevos empleados.

C. Controles físicos

Restringir el acceso físico a áreas seguras (salas de servidores, centros de datos).

Proteger los equipos de cortes de energía y daños ambientales.

D. Controles tecnológicos (ejemplos específicos para este incidente)

Gestión de identidad y acceso:

- Implemente MFA para todos los inicios de sesión administrativos (SSH, administrador de WordPress).
- Revise y refuerce los permisos de archivos y directorios críticos (p. ej., */var/www/html/wp-content/uploads/* a 755).
- Desactive la autenticación de contraseña para el usuario root mediante SSH.
- Protección contra malware:
- Implemente y mantenga un software antivirus/EDR en todos los servidores.
- Realice análisis periódicos de malware y rootkits.

Gestión de vulnerabilidades:

- Implemente un proceso continuo de análisis de vulnerabilidades (interno y externo).
- Mantenga los sistemas operativos, las aplicaciones (WordPress, plugins, temas) y los servicios (Apache, MySQL, OpenSSH, vsFTPd) actualizados con los últimos parches de seguridad.

Seguridad de red:

- Configure los firewalls (basados en el host y perimetrales) para permitir solo el tráfico esencial (p. ej., puertos 22, 80, 443).
- Desactive los servicios innecesarios (p. ej., FTP anónimo). Implementar la segmentación de la red para aislar los servicios críticos (p. ej., bases de datos).

Copia de seguridad y recuperación:

- Automatizar los procesos de copia de seguridad, comprobar su integridad y restaurar las capacidades periódicamente.

Registro y supervisión:

- Garantizar que todos los registros de seguridad (de autenticación, del sistema, de la aplicación, WAF) estén habilitados, configurados con el nivel de detalle adecuado y centralizados en un SIEM para su análisis y alertas en tiempo real.

- Investigar y corregir la causa de la falta de registros críticos, como */var/log/auth.log*.

E. Evaluación y mejora del rendimiento

Realizar auditorías internas periódicas del SGSI.

Realizar revisiones periódicas de la dirección para garantizar la idoneidad y eficacia continuas del SGSI.

Impulsar la mejora continua del SGSI en función de los resultados de las auditorías, los incidentes y los cambios en el entorno de amenazas.

6. Conclusión final

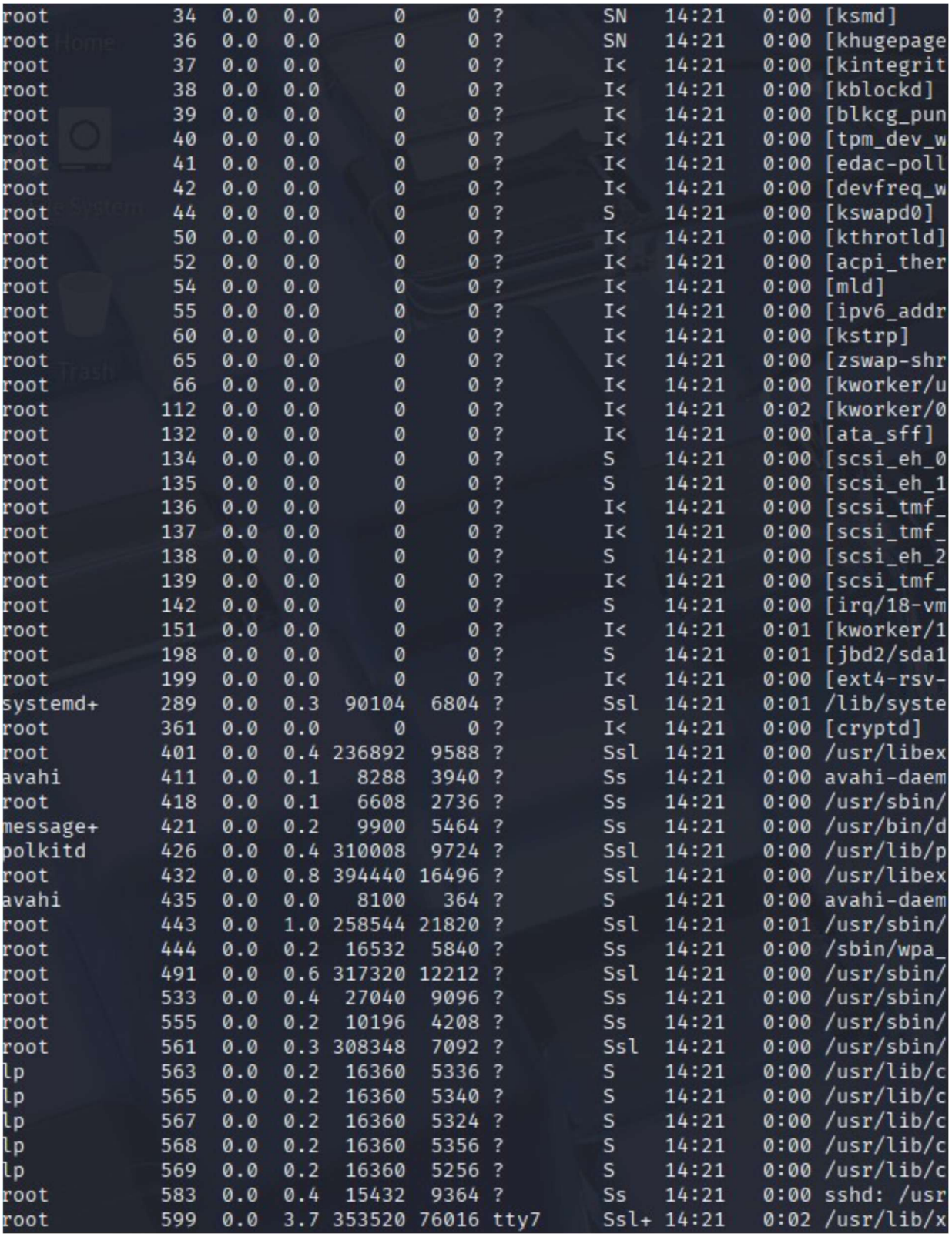
Las fases 1 y 2 identificaron y remediaron vulnerabilidades críticas, como credenciales SSH débiles, permisos de directorio incorrectos en WordPress y la exposición de FTP anónimo.

La fase 3 complementa estas acciones reactivas con un marco proactivo.

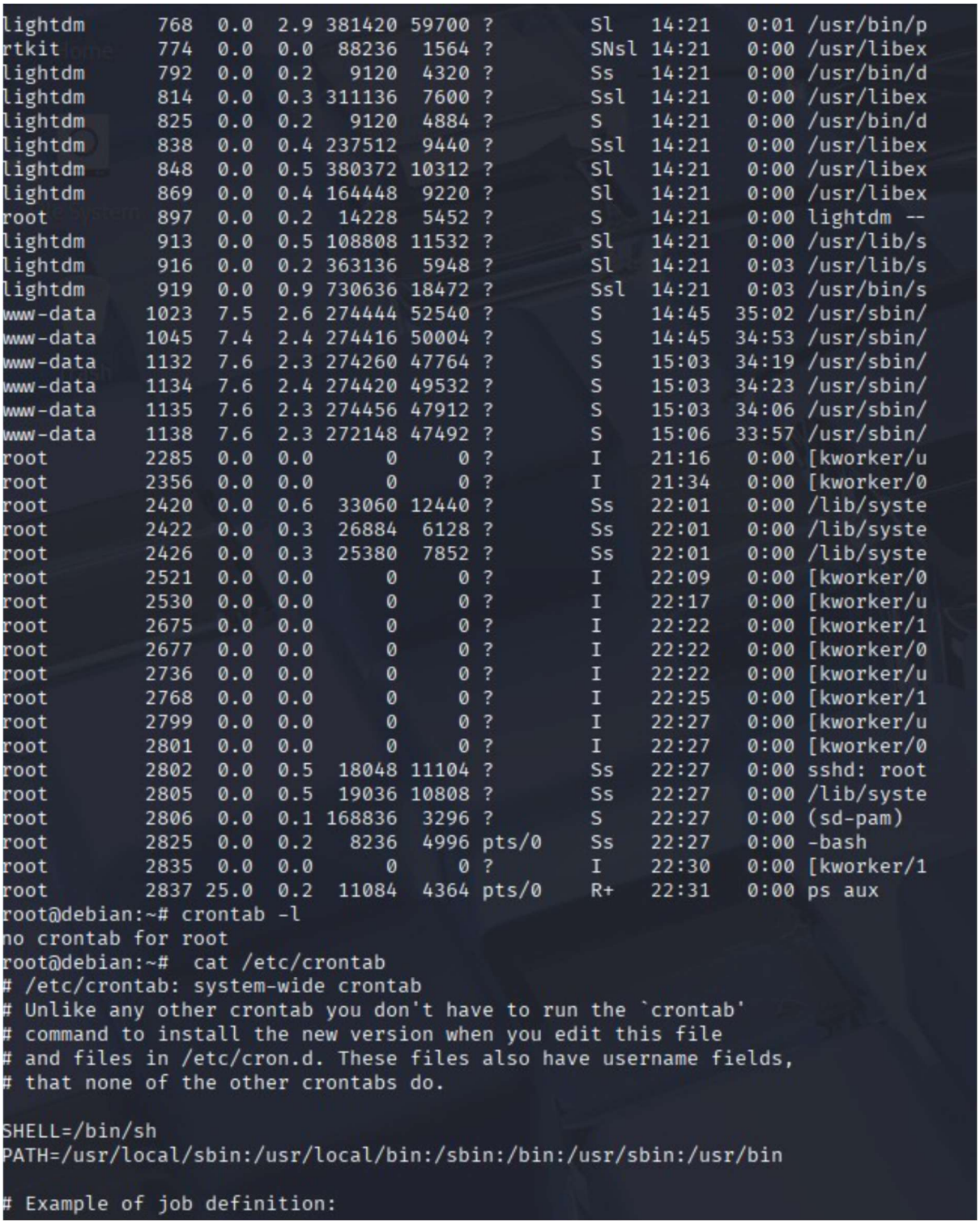
Al implementar este Plan de Respuesta a Incidentes basado en NIST SP 800-61 y desarrollar un SGSI conforme a la norma ISO 27001, la organización estará significativamente mejor preparada para prevenir, detectar, responder y recuperarse de futuros incidentes de seguridad. Esta es una inversión continua en resiliencia y en la protección de los activos de información más valiosos de la organización.

ANEXO I. EVIDENCIAS GRAFICAS

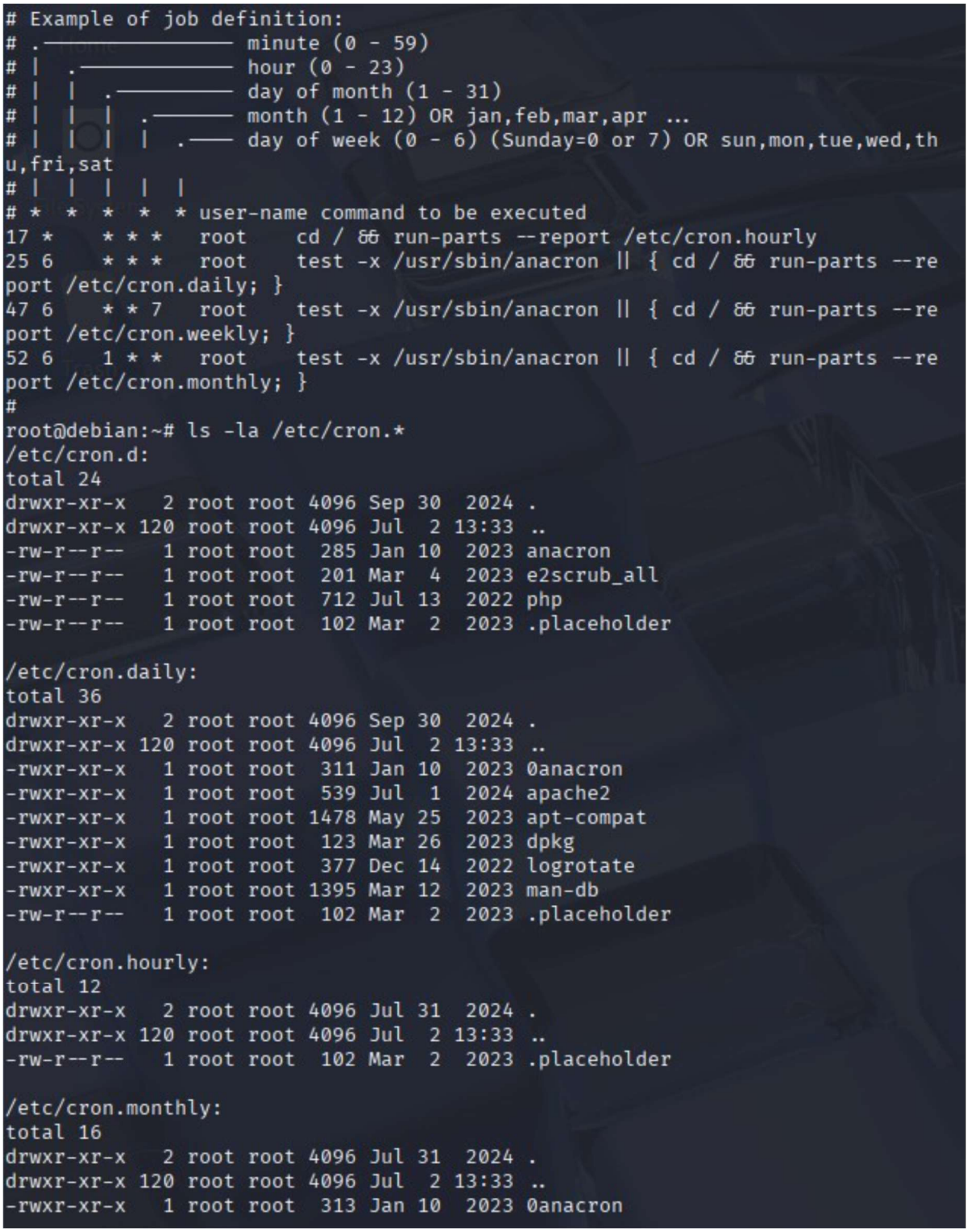
```
┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.56.102
root@192.168.56.102's password:
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-
26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul  2 22:25:30 2025 from 192.168.56.101
root@debian:~# ls -la /var/www/html/wp-content/uploads/
total 16
drwxrwxrwx 4 www-data www-data 4096 Jul  2 14:45 .
drwxrwxrwx 5 www-data www-data 4096 Jul  2 14:45 ..
drwxrwxrwx 4 www-data www-data 4096 Oct  8  2024 2024
drwxrwxrwx 3 www-data www-data 4096 Jul  2 14:45 2025
root@debian:~# grep "sshd" /var/log/auth.log | less
root@debian:~# grep "Accepted password for root" /var/log/auth.log | less
root@debian:~# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.6 167724 12344 ?        Ss   14:21   0:02 /sbin/init
root           2  0.0  0.0      0     0 ?        S    14:21   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   14:21   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   14:21   0:00 [rcu_par_g
root           5  0.0  0.0      0     0 ?        I<   14:21   0:00 [slub_flus
root           6  0.0  0.0      0     0 ?        I<   14:21   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   14:21   0:00 [kworker/0
root          10  0.0  0.0      0     0 ?        I<   14:21   0:00 [mm_percpu
root          11  0.0  0.0      0     0 ?        I    14:21   0:00 [rcu_tasks
root          12  0.0  0.0      0     0 ?        I    14:21   0:00 [rcu_tasks
root          13  0.0  0.0      0     0 ?        I    14:21   0:00 [rcu_tasks
root          14  0.0  0.0      0     0 ?        S    14:21   0:06 [ksoftirqd
root          15  0.1  0.0      0     0 ?        I    14:21   0:41 [rcu_preem
root          16  0.0  0.0      0     0 ?        S    14:21   0:00 [migration
root          18  0.0  0.0      0     0 ?        S    14:21   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    14:21   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    14:21   0:01 [migration
root          21  0.2  0.0      0     0 ?        S    14:21   0:59 [ksoftirqd
root          23  0.0  0.0      0     0 ?        I<   14:21   0:00 [kworker/1
root          26  0.0  0.0      0     0 ?        S    14:21   0:00 [kdevtmpfs
root          27  0.0  0.0      0     0 ?        I<   14:21   0:00 [inet_frag
root          28  0.0  0.0      0     0 ?        S    14:21   0:00 [kauditd]
root          29  0.0  0.0      0     0 ?        S    14:21   0:00 [khungtask
root          30  0.0  0.0      0     0 ?        S    14:21   0:00 [oom_reape
root          32  0.0  0.0      0     0 ?        I<   14:21   0:00 [writeback
root          33  0.0  0.0      0     0 ?        S    14:21   0:01 [kcompactd
root          34  0.0  0.0      0     0 ?        SN   14:21   0:00 [ksmd]
```



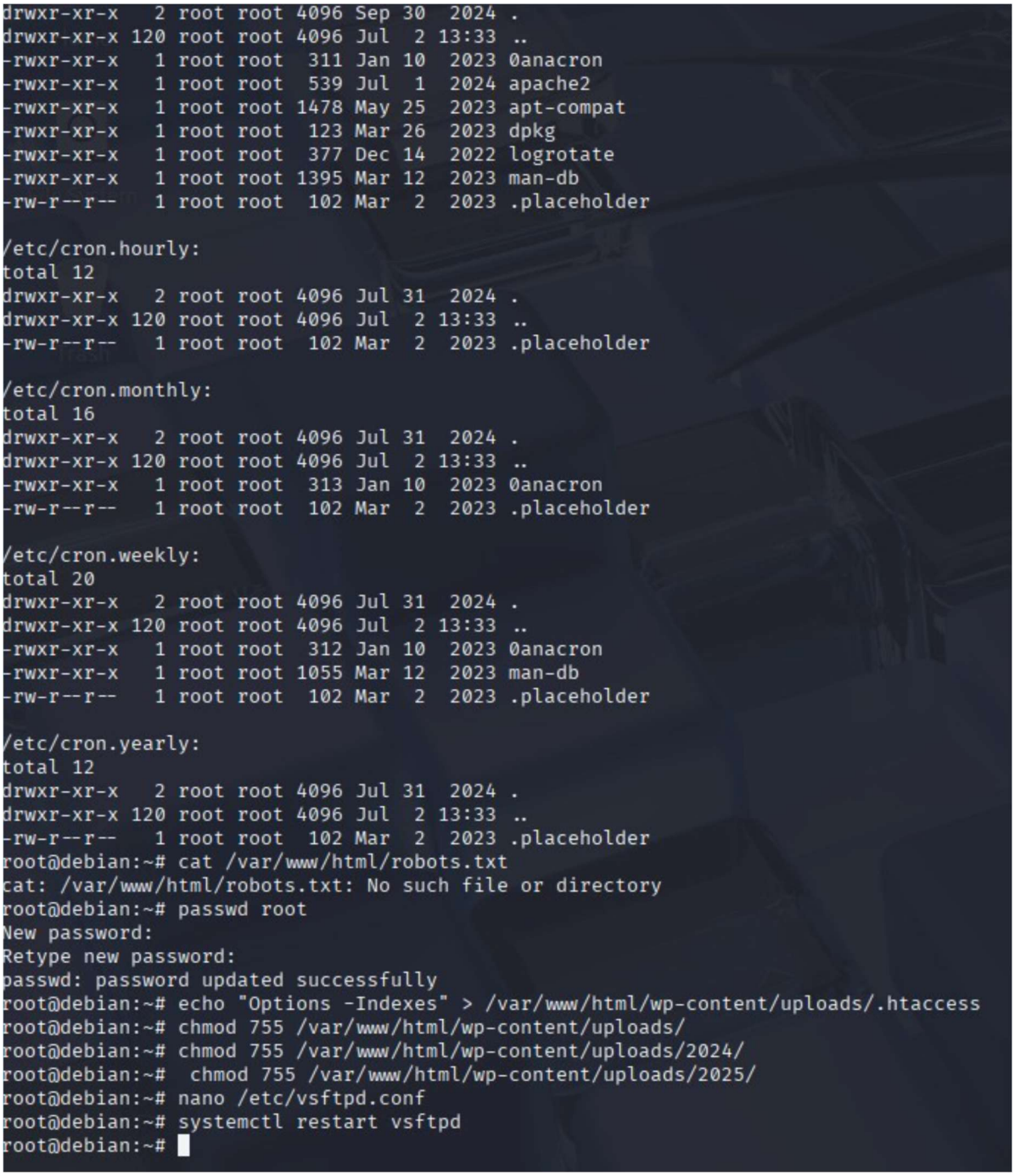
```
root          34  0.0  0.0      0     0 ?        SN   14:21   0:00 [ksmd]
root          36  0.0  0.0      0     0 ?        SN   14:21   0:00 [khugepage
root          37  0.0  0.0      0     0 ?        I<   14:21   0:00 [kintegrit
root          38  0.0  0.0      0     0 ?        I<   14:21   0:00 [kblockd]
root          39  0.0  0.0      0     0 ?        I<   14:21   0:00 [blkcg_pun
root          40  0.0  0.0      0     0 ?        I<   14:21   0:00 [tpm_dev_w
root          41  0.0  0.0      0     0 ?        I<   14:21   0:00 [edac-poll
root          42  0.0  0.0      0     0 ?        I<   14:21   0:00 [devfreq_w
root          44  0.0  0.0      0     0 ?        S    14:21   0:00 [kswapd0]
root          50  0.0  0.0      0     0 ?        I<   14:21   0:00 [kthrotld]
root          52  0.0  0.0      0     0 ?        I<   14:21   0:00 [acpi_ther
root          54  0.0  0.0      0     0 ?        I<   14:21   0:00 [mld]
root          55  0.0  0.0      0     0 ?        I<   14:21   0:00 [ipv6_addr
root          60  0.0  0.0      0     0 ?        I<   14:21   0:00 [kstrp]
root          65  0.0  0.0      0     0 ?        I<   14:21   0:00 [zswap-shr
root          66  0.0  0.0      0     0 ?        I<   14:21   0:00 [kworker/u
root         112  0.0  0.0      0     0 ?        I<   14:21   0:02 [kworker/0
root         132  0.0  0.0      0     0 ?        I<   14:21   0:00 [ata_sff]
root         134  0.0  0.0      0     0 ?        S    14:21   0:00 [scsi_eh_0
root         135  0.0  0.0      0     0 ?        S    14:21   0:00 [scsi_eh_1
root         136  0.0  0.0      0     0 ?        I<   14:21   0:00 [scsi_tmf_
root         137  0.0  0.0      0     0 ?        I<   14:21   0:00 [scsi_tmf_
root         138  0.0  0.0      0     0 ?        S    14:21   0:00 [scsi_eh_2
root         139  0.0  0.0      0     0 ?        I<   14:21   0:00 [scsi_tmf_
root         142  0.0  0.0      0     0 ?        S    14:21   0:00 [irq/18-vm
root         151  0.0  0.0      0     0 ?        I<   14:21   0:01 [kworker/1
root         198  0.0  0.0      0     0 ?        S    14:21   0:01 [jbd2/sda1
root         199  0.0  0.0      0     0 ?        I<   14:21   0:00 [ext4-rsv-
systemd+     289  0.0  0.3  90104  6804 ?        Ssl  14:21   0:01 /lib/syste
root         361  0.0  0.0      0     0 ?        I<   14:21   0:00 [cryptd]
root         401  0.0  0.4 236892  9588 ?        Ssl  14:21   0:00 /usr/libex
avahi        411  0.0  0.1   8288  3940 ?        Ss   14:21   0:00 avahi-daem
root         418  0.0  0.1   6608  2736 ?        Ss   14:21   0:00 /usr/sbin/
message+     421  0.0  0.2   9900  5464 ?        Ss   14:21   0:00 /usr/bin/d
polkitd      426  0.0  0.4 310008  9724 ?        Ssl  14:21   0:00 /usr/lib/p
root         432  0.0  0.8 394440 16496 ?        Ssl  14:21   0:00 /usr/libex
avahi        435  0.0  0.0   8100   364 ?        S    14:21   0:00 avahi-daem
root         443  0.0  1.0 258544 21820 ?        Ssl  14:21   0:01 /usr/sbin/
root         444  0.0  0.2  16532  5840 ?        Ss   14:21   0:00 /sbin/wpa_
root         491  0.0  0.6 317320 12212 ?        Ssl  14:21   0:00 /usr/sbin/
root         533  0.0  0.4  27040  9096 ?        Ss   14:21   0:00 /usr/sbin/
root         555  0.0  0.2  10196  4208 ?        Ss   14:21   0:00 /usr/sbin/
root         561  0.0  0.3 308348  7092 ?        Ssl  14:21   0:00 /usr/sbin/
lp           563  0.0  0.2  16360  5336 ?        S    14:21   0:00 /usr/lib/c
lp           565  0.0  0.2  16360  5340 ?        S    14:21   0:00 /usr/lib/c
lp           567  0.0  0.2  16360  5324 ?        S    14:21   0:00 /usr/lib/c
lp           568  0.0  0.2  16360  5356 ?        S    14:21   0:00 /usr/lib/c
lp           569  0.0  0.2  16360  5256 ?        S    14:21   0:00 /usr/lib/c
root         583  0.0  0.4  15432  9364 ?        Ss   14:21   0:00 sshd: /usr
root         599  0.0  3.7 353520 76016 tty7     Ssl+ 14:21   0:02 /usr/lib/x
```

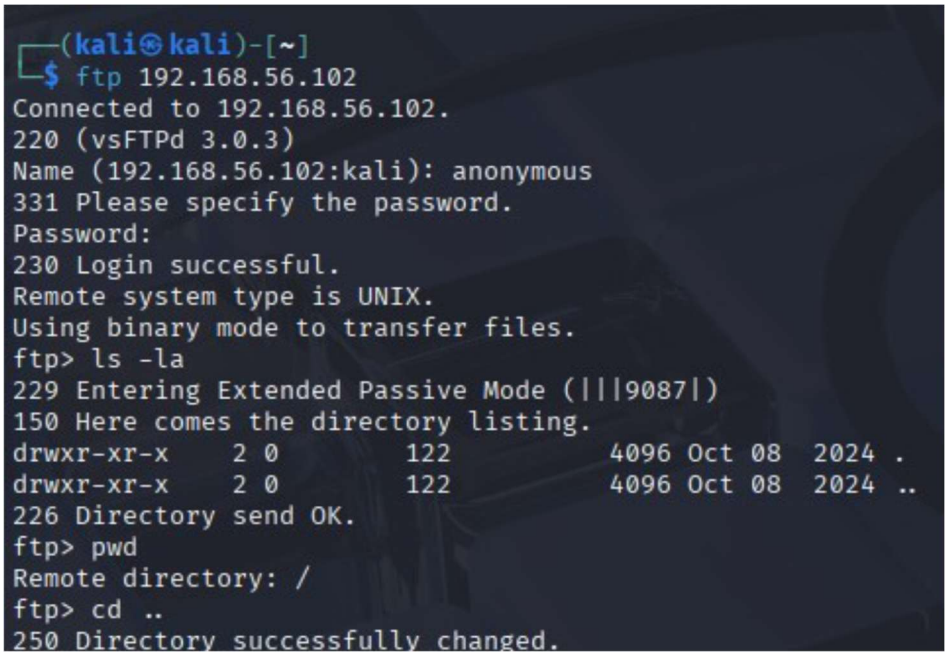


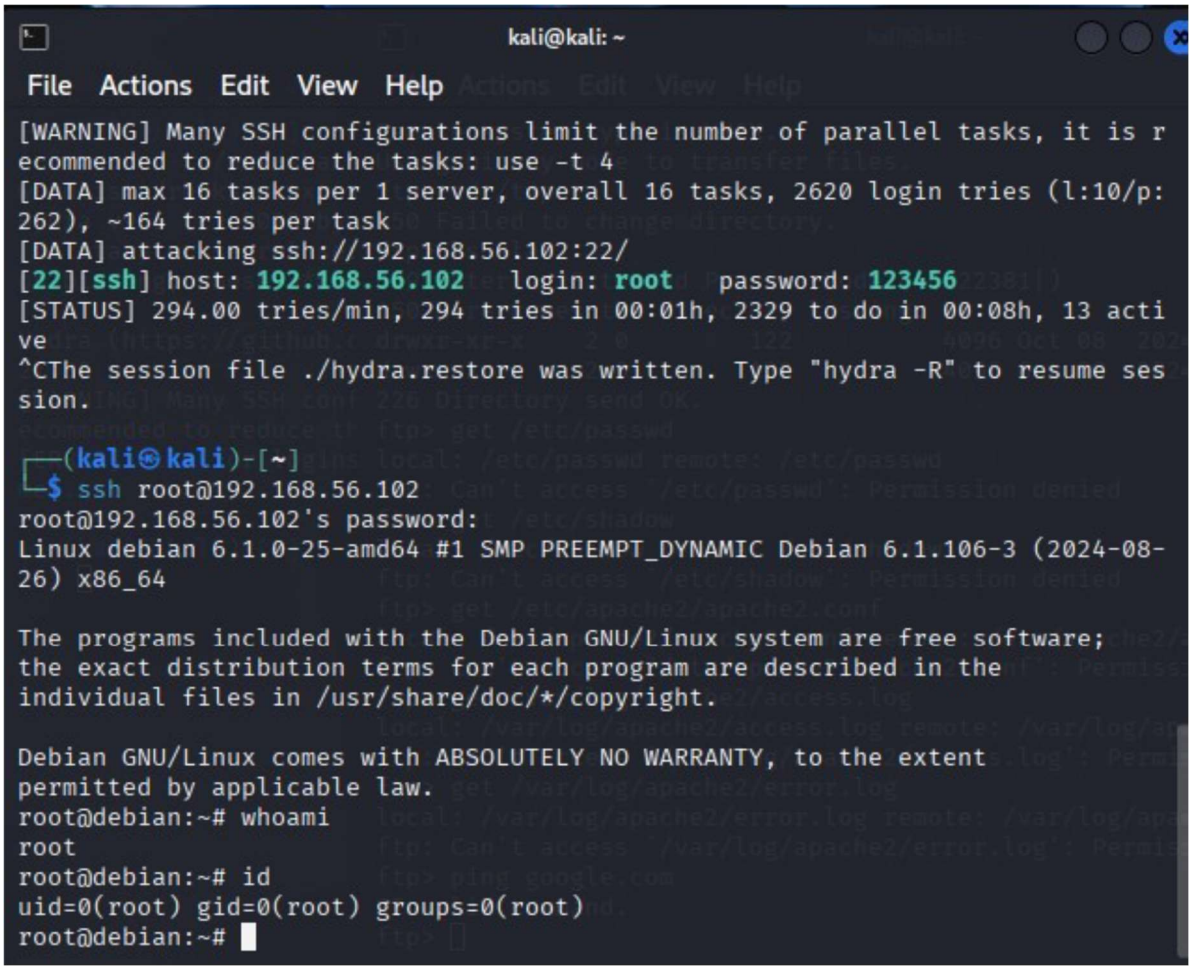
```
lightdm      768  0.0  2.9 381420 59700 ?        Sl   14:21   0:01 /usr/bin/p
rtkit        774  0.0  0.0  88236  1564 ?        SNsl 14:21   0:00 /usr/libex
lightdm      792  0.0  0.2   9120  4320 ?        Ss   14:21   0:00 /usr/bin/d
lightdm      814  0.0  0.3 311136  7600 ?        Ssl  14:21   0:00 /usr/libex
lightdm      825  0.0  0.2   9120  4884 ?        S    14:21   0:00 /usr/bin/d
lightdm      838  0.0  0.4 237512  9440 ?        Ssl  14:21   0:00 /usr/libex
lightdm      848  0.0  0.5 380372 10312 ?        Sl   14:21   0:00 /usr/libex
lightdm      869  0.0  0.4 164448  9220 ?        Sl   14:21   0:00 /usr/libex
root         897  0.0  0.2  14228  5452 ?        S    14:21   0:00 lightdm --
lightdm      913  0.0  0.5 108808 11532 ?        Sl   14:21   0:00 /usr/lib/s
lightdm      916  0.0  0.2 363136  5948 ?        Sl   14:21   0:03 /usr/lib/s
lightdm      919  0.0  0.9 730636 18472 ?        Ssl  14:21   0:03 /usr/bin/s
www-data    1023  7.5  2.6 274444 52540 ?        S    14:45  35:02 /usr/sbin/
www-data    1045  7.4  2.4 274416 50004 ?        S    14:45  34:53 /usr/sbin/
www-data    1132  7.6  2.3 274260 47764 ?        S    15:03  34:19 /usr/sbin/
www-data    1134  7.6  2.4 274420 49532 ?        S    15:03  34:23 /usr/sbin/
www-data    1135  7.6  2.3 274456 47912 ?        S    15:03  34:06 /usr/sbin/
www-data    1138  7.6  2.3 272148 47492 ?        S    15:06  33:57 /usr/sbin/
root        2285  0.0  0.0      0     0 ?        I    21:16   0:00 [kworker/u
root        2356  0.0  0.0      0     0 ?        I    21:34   0:00 [kworker/0
root        2420  0.0  0.6  33060 12440 ?        Ss   22:01   0:00 /lib/syste
root        2422  0.0  0.3  26884  6128 ?        Ss   22:01   0:00 /lib/syste
root        2426  0.0  0.3  25380  7852 ?        Ss   22:01   0:00 /lib/syste
root        2521  0.0  0.0      0     0 ?        I    22:09   0:00 [kworker/0
root        2530  0.0  0.0      0     0 ?        I    22:17   0:00 [kworker/u
root        2675  0.0  0.0      0     0 ?        I    22:22   0:00 [kworker/1
root        2677  0.0  0.0      0     0 ?        I    22:22   0:00 [kworker/0
root        2736  0.0  0.0      0     0 ?        I    22:22   0:00 [kworker/u
root        2768  0.0  0.0      0     0 ?        I    22:25   0:00 [kworker/1
root        2799  0.0  0.0      0     0 ?        I    22:27   0:00 [kworker/u
root        2801  0.0  0.0      0     0 ?        I    22:27   0:00 [kworker/0
root        2802  0.0  0.5  18048 11104 ?        Ss   22:27   0:00 sshd: root
root        2805  0.0  0.5  19036 10808 ?        Ss   22:27   0:00 /lib/syste
root        2806  0.0  0.1 168836  3296 ?        S    22:27   0:00 (sd-pam)
root        2825  0.0  0.2   8236  4996 pts/0    Ss   22:27   0:00 -bash
root        2835  0.0  0.0      0     0 ?        I    22:30   0:00 [kworker/1
root        2837 25.0  0.2  11084  4364 pts/0    R+   22:31   0:00 ps aux
root@debian:~# crontab -l
no crontab for root
root@debian:~#  cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
```

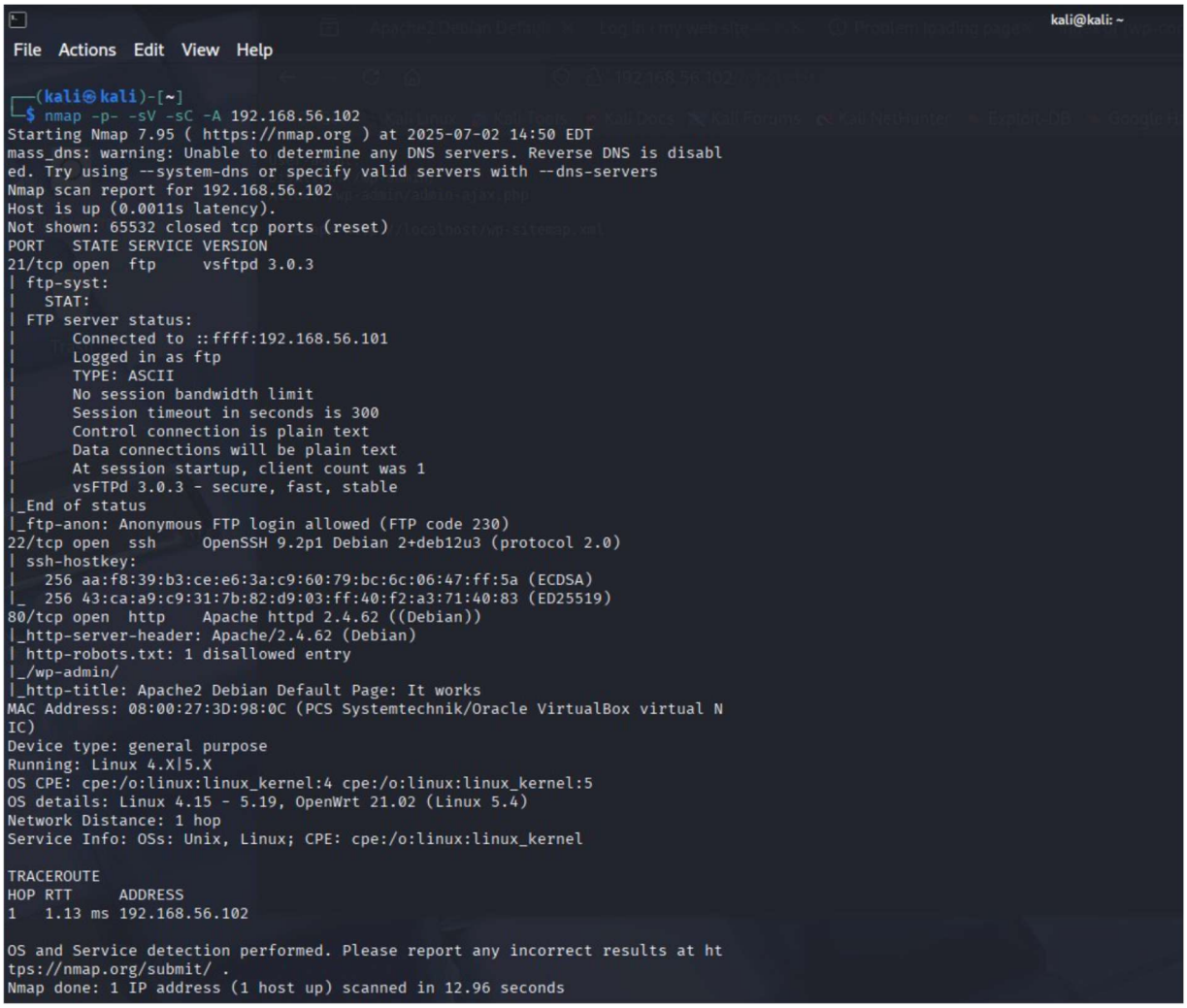


```
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr  ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,th
u,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --re
port /etc/cron.daily; }
47 6    * * 7   root    test -x /usr/sbin/anacron || { cd / && run-parts --re
port /etc/cron.weekly; }
52 6    1 * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --re
port /etc/cron.monthly; }
#
root@debian:~# ls -la /etc/cron.*
/etc/cron.d:
total 24
drwxr-xr-x   2 root root 4096 Sep 30  2024 .
drwxr-xr-x 120 root root 4096 Jul  2 13:33 ..
-rw-r--r--   1 root root  285 Jan 10  2023 anacron
-rw-r--r--   1 root root  201 Mar  4  2023 e2scrub_all
-rw-r--r--   1 root root  712 Jul 13  2022 php
-rw-r--r--   1 root root  102 Mar  2  2023 .placeholder

/etc/cron.daily:
total 36
drwxr-xr-x   2 root root 4096 Sep 30  2024 .
drwxr-xr-x 120 root root 4096 Jul  2 13:33 ..
-rwxr-xr-x   1 root root  311 Jan 10  2023 0anacron
-rwxr-xr-x   1 root root  539 Jul  1  2024 apache2
-rwxr-xr-x   1 root root 1478 May 25  2023 apt-compat
-rwxr-xr-x   1 root root  123 Mar 26  2023 dpkg
-rwxr-xr-x   1 root root  377 Dec 14  2022 logrotate
-rwxr-xr-x   1 root root 1395 Mar 12  2023 man-db
-rw-r--r--   1 root root  102 Mar  2  2023 .placeholder

/etc/cron.hourly:
total 12
drwxr-xr-x   2 root root 4096 Jul 31  2024 .
drwxr-xr-x 120 root root 4096 Jul  2 13:33 ..
-rw-r--r--   1 root root  102 Mar  2  2023 .placeholder

/etc/cron.monthly:
total 16
drwxr-xr-x   2 root root 4096 Jul 31  2024 .
drwxr-xr-x 120 root root 4096 Jul  2 13:33 ..
-rwxr-xr-x   1 root root  313 Jan 10  2023 0anacron
```

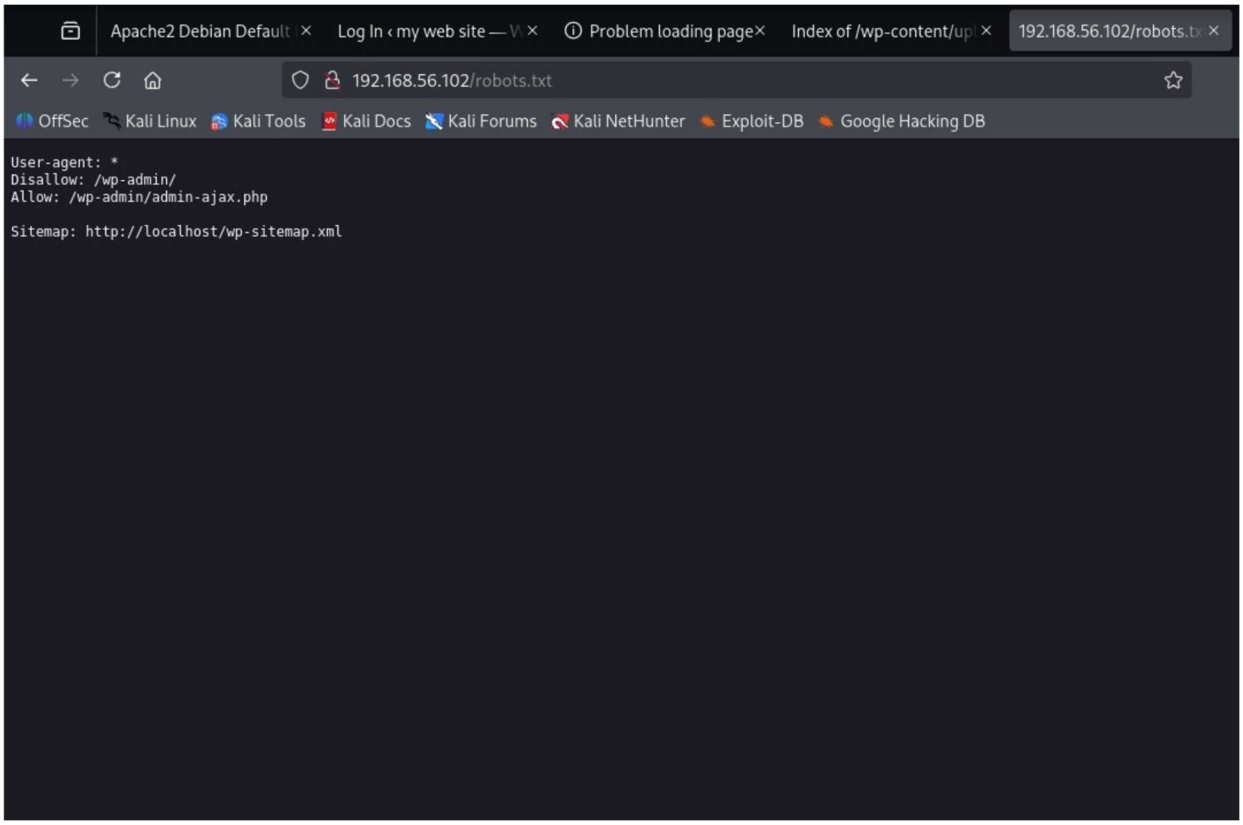

drwxr-xr-x   2 root root 4096 Sep 30 2024 .
drwxr-xr-x 120 root root 4096 Jul 2 13:33 ..
-rwxr-xr-x   1 root root 311 Jan 10 2023 0anacron
-rwxr-xr-x   1 root root 539 Jul 1 2024 apache2
-rwxr-xr-x   1 root root 1478 May 25 2023 apt-compat
-rwxr-xr-x   1 root root 123 Mar 26 2023 dpkg
-rwxr-xr-x   1 root root 377 Dec 14 2022 logrotate
-rwxr-xr-x   1 root root 1395 Mar 12 2023 man-db
-rw-r--r--   1 root root 102 Mar 2 2023 .placeholder
/etc/cron.hourly:
total 12
drwxr-xr-x   2 root root 4096 Jul 31 2024 .
drwxr-xr-x 120 root root 4096 Jul 2 13:33 ..
-rw-r--r--   1 root root 102 Mar 2 2023 .placeholder
/etc/cron.monthly:
total 16
drwxr-xr-x   2 root root 4096 Jul 31 2024 .
drwxr-xr-x 120 root root 4096 Jul 2 13:33 ..
-rwxr-xr-x   1 root root 313 Jan 10 2023 0anacron
-rw-r--r--   1 root root 102 Mar 2 2023 .placeholder
/etc/cron.weekly:
total 20
drwxr-xr-x   2 root root 4096 Jul 31 2024 .
drwxr-xr-x 120 root root 4096 Jul 2 13:33 ..
-rwxr-xr-x   1 root root 312 Jan 10 2023 0anacron
-rwxr-xr-x   1 root root 1055 Mar 12 2023 man-db
-rw-r--r--   1 root root 102 Mar 2 2023 .placeholder
/etc/cron.yearly:
total 12
drwxr-xr-x   2 root root 4096 Jul 31 2024 .
drwxr-xr-x 120 root root 4096 Jul 2 13:33 ..
-rw-r--r--   1 root root 102 Mar 2 2023 .placeholder
root@debian:~# cat /var/www/html/robots.txt
cat: /var/www/html/robots.txt: No such file or directory
root@debian:~# passwd root
New password:
Retype new password:
passwd: password updated successfully
root@debian:~# echo "Options -Indexes" > /var/www/html/wp-content/uploads/.htaccess
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/2024/
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/2025/
root@debian:~# nano /etc/vsftpd.conf
root@debian:~# systemctl restart vsftpd
root@debian:~#

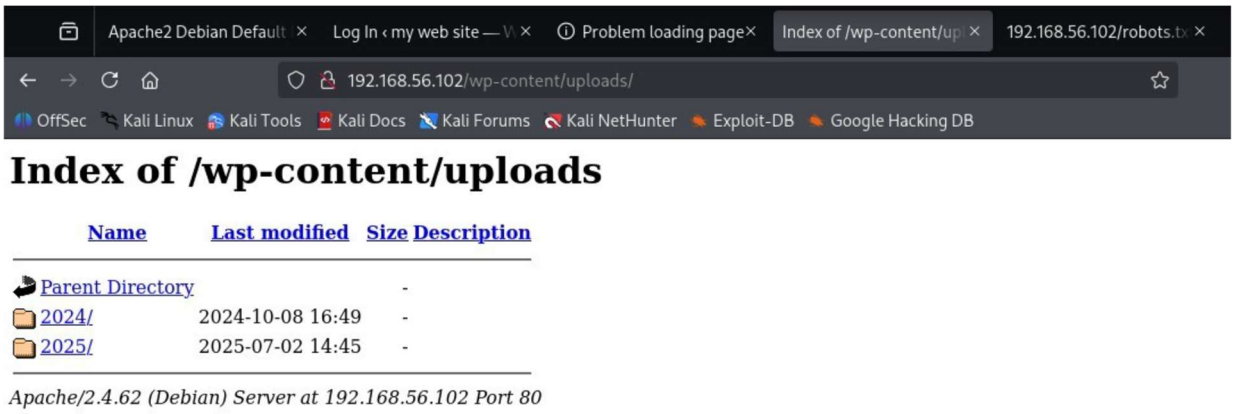

┌──(kali㉿kali)-[~]
└─$ ftp 192.168.56.102
Connected to 192.168.56.102.
220 (vsFTPd 3.0.3)
Name (192.168.56.102:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||9087|)
150 Here comes the directory listing.
drwxr-xr-x   2 0   122   4096 Oct 08 2024 .
drwxr-xr-x   2 0   122   4096 Oct 08 2024 ..
226 Directory send OK.
ftp> pwd
Remote directory: /
ftp> cd ..
250 Directory successfully changed.


kali@kali: ~
File Actions Edit View Help
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 2620 login tries (l:10/p:
262), ~164 tries per task
[DATA] attacking ssh://192.168.56.102:22/
[22][ssh] host: 192.168.56.102   login: root   password: 123456
[STATUS] 294.00 tries/min, 294 tries in 00:01h, 2329 to do in 00:08h, 13 acti
ve
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume ses
sion.
┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.56.102
root@192.168.56.102's password:
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-
26) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# whoami
root
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:~#


kali@kali: ~
File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ nmap -p- -sV -sC -A 192.168.56.102
Starting Nmap 7.95 (https://nmap.org) at 2025-07-02 14:50 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabl
ed. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.102
Host is up (0.0011s latency).
Not shown: 65532 closed tcp ports (reset)
PORT STATE SERVICE VERSION
21/tcp open ftp vsftpd 3.0.3
| ftp-syst:
| STAT:
| FTP server status:
| Connected to ::ffff:192.168.56.101
| Logged in as ftp
| TYPE: ASCII
| No session bandwidth limit
| Session timeout in seconds is 300
| Control connection is plain text
| Data connections will be plain text
| At session startup, client count was 1
| vsFTPd 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp open ssh OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
| 256 aa:f8:39:b3:ce:e6:3a:c9:60:79:bc:6c:06:47:ff:5a (ECDSA)
|_ 256 43:ca:a9:c9:31:7b:82:d9:03:ff:40:f2:a3:71:40:83 (ED25519)
80/tcp open http Apache httpd 2.4.62 ((Debian))
|_http-server-header: Apache/2.4.62 (Debian)
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-title: Apache2 Debian Default Page: It works
MAC Address: 08:00:27:3D:98:0C (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4)
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
TRACEROUTE
HOP RTT ADDRESS
1 1.13 ms 192.168.56.102
OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.96 seconds


root@debian:~# passwd root
New password:
Retype new password:
passwd: password updated successfully
root@debian:~# echo "Options -Indexes" > /var/www/html/wp-content/uploads/.htaccess
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/2024/
root@debian:~# chmod 755 /var/www/html/wp-content/uploads/2025/
root@debian:~# nano /etc/vsftpd.conf
root@debian:~# systemctl restart vsftpd
root@debian:~#

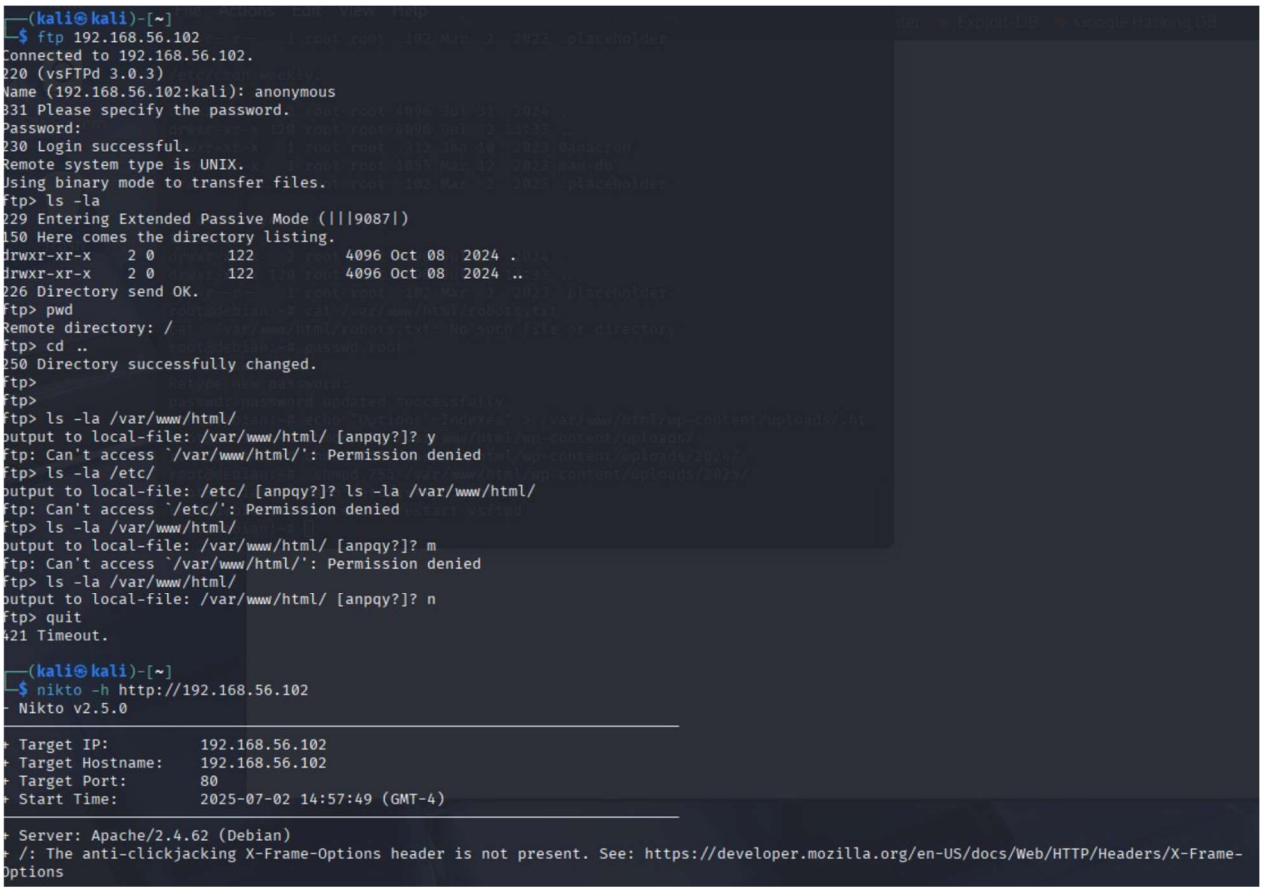




Index of /wp-content/uploads

Name	Last modified	Size	Description
Parent Directory		-	
2024/	2024-10-08 16:49	-	
2025/	2025-07-02 14:45	-	

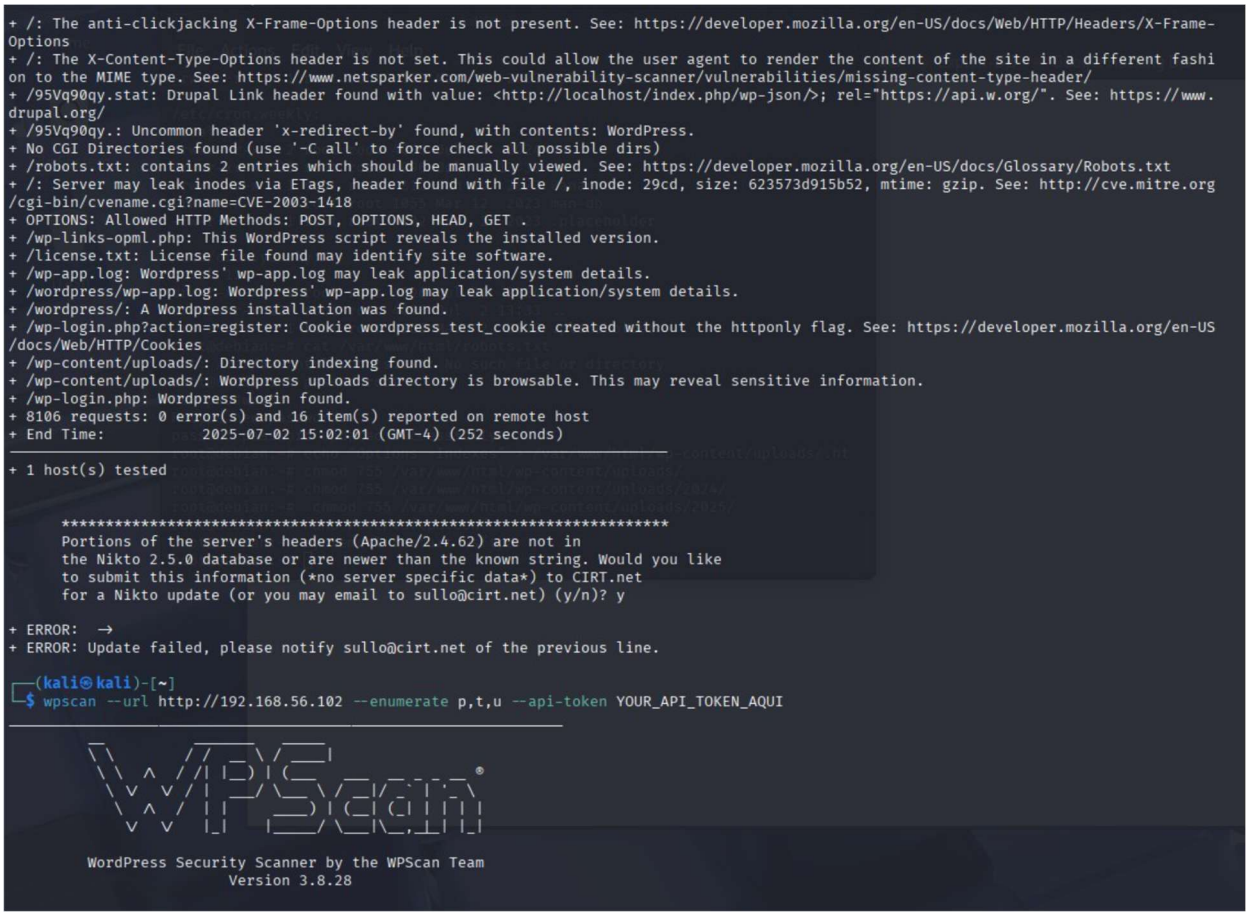
Apache/2.4.62 (Debian) Server at 192.168.56.102 Port 80

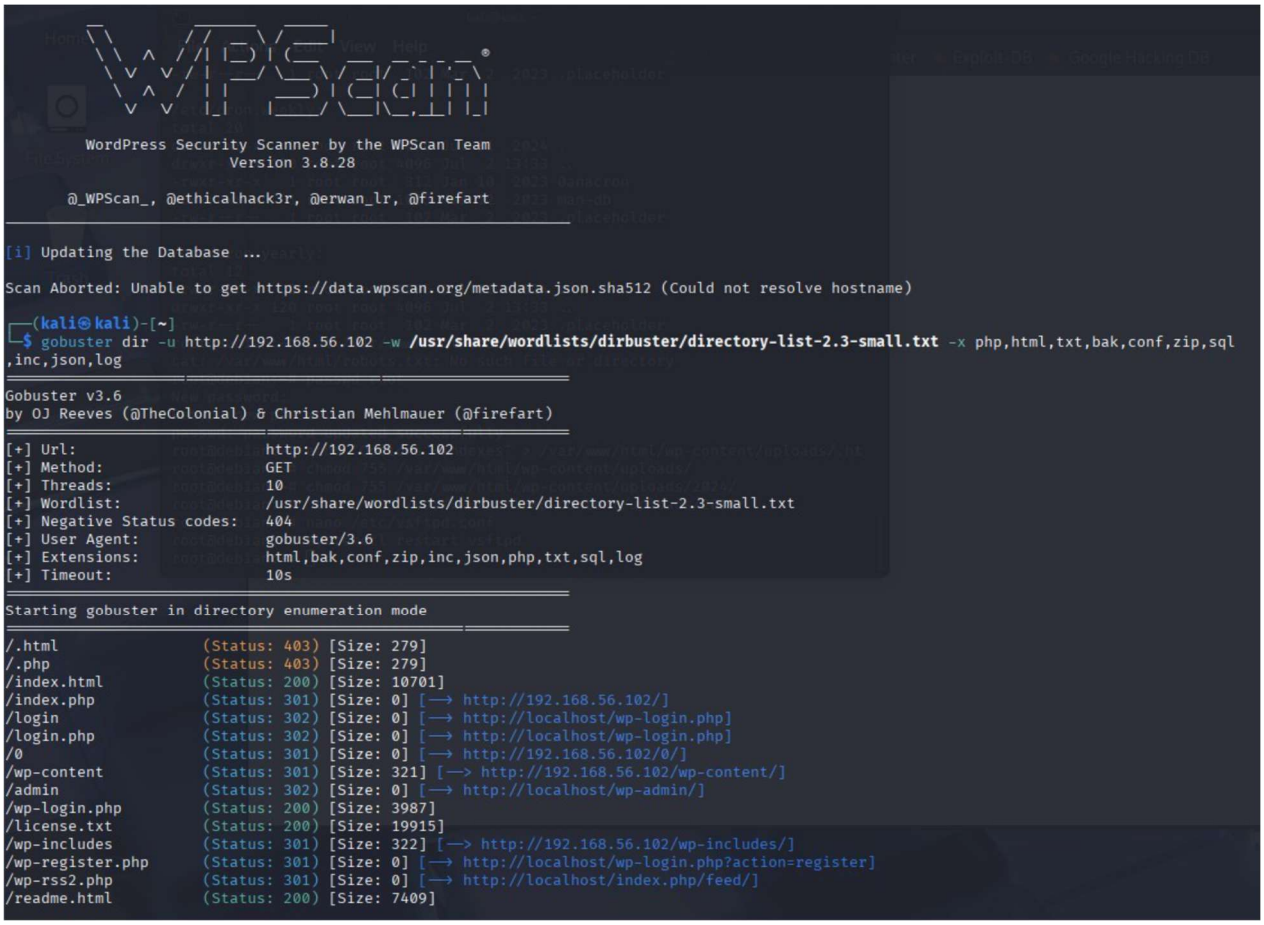

┌──(kali㉿kali)-[~]
└─$ ftp 192.168.56.102
Connected to 192.168.56.102.
220 (vsFTPd 3.0.3)
Name (192.168.56.102:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||9087|)
150 Here comes the directory listing.
drwxr-xr-x   2 0   122   4096 Oct 08 2024 .
drwxr-xr-x   2 0   122   4096 Oct 08 2024 ..
226 Directory send OK.
ftp> pwd
Remote directory: /
ftp> cd ..
250 Directory successfully changed.
ftp>
ftp>
ftp> ls -la /var/www/html/
output to local-file: /var/www/html/ [anpqy?]? y
ftp: Can't access `/var/www/html/': Permission denied
ftp> ls -la /etc/
output to local-file: /etc/ [anpqy?]? ls -la /var/www/html/
ftp: Can't access `/etc/': Permission denied
ftp> ls -la /var/www/html/
output to local-file: /var/www/html/ [anpqy?]? m
ftp: Can't access `/var/www/html/': Permission denied
ftp> ls -la /var/www/html/
output to local-file: /var/www/html/ [anpqy?]? n
ftp> quit
421 Timeout.

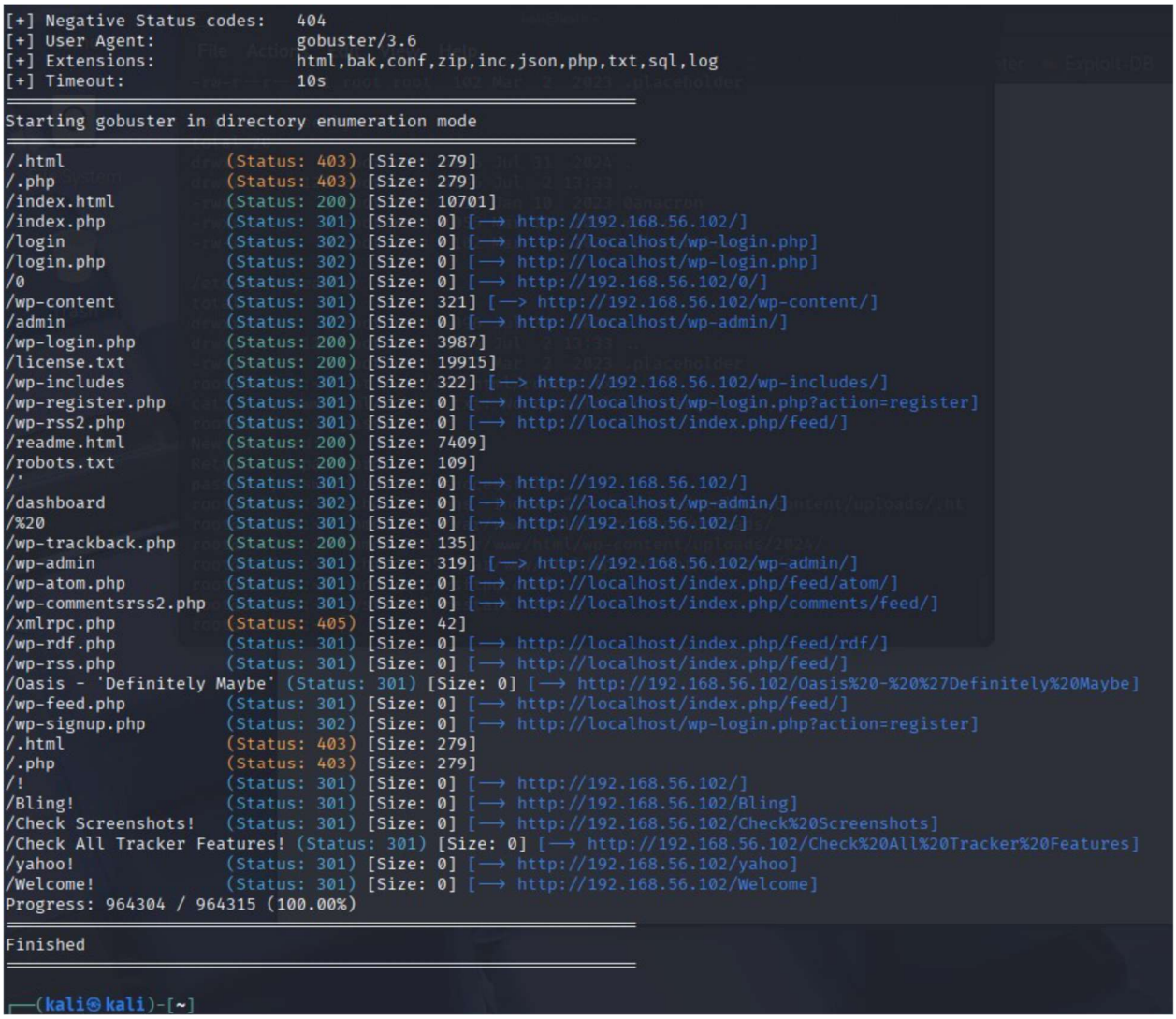
┌──(kali㉿kali)-[~]
└─$ nikto -h http://192.168.56.102
- Nikto v2.5.0

+ Target IP: 192.168.56.102
+ Target Hostname: 192.168.56.102
+ Target Port: 80
+ Start Time: 2025-07-02 14:57:49 (GMT-4)

+ Server: Apache/2.4.62 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options


+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: 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. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /95Vq90qy.stat: Drupal Link header found with value: <http://localhost/index.php/wp-json/>; rel="https://api.w.org/". See: https://www.drupal.org/
+ /95Vq90qy.: Uncommon header 'x-redirect-by' found, with contents: WordPress.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: contains 2 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ /: Server may leak inodes via ETags, header found with file /, inode: 29cd, size: 623573d915b52, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: POST, OPTIONS, HEAD, GET .
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /wordpress/wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /wordpress/: A Wordpress installation was found.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-content/uploads/: Directory indexing found.
+ /wp-content/uploads/: Wordpress uploads directory is browsable. This may reveal sensitive information.
+ /wp-login.php: Wordpress login found.
+ 8106 requests: 0 error(s) and 16 item(s) reported on remote host
+ End Time: 2025-07-02 15:02:01 (GMT-4) (252 seconds)
+ 1 host(s) tested
Portions of the server's headers (Apache/2.4.62) are not in
the Nikto 2.5.0 database or are newer than the known string. Would you like
to submit this information (*no server specific data*) to CIRT.net
for a Nikto update (or you may email to sullo@cirt.net) (y/n)? y
+ ERROR: →
+ ERROR: Update failed, please notify sullo@cirt.net of the previous line.
┌──(kali㉿kali)-[~]
└─$ wpscan --url http://192.168.56.102 --enumerate p,t,u --api-token YOUR_API_TOKEN_AQUI
WordPress Security Scanner by the WPScan Team
Version 3.8.28


WordPress Security Scanner by the WPScan Team
Version 3.8.28
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
[i] Updating the Database ...
Scan Aborted: Unable to get https://data.wpscan.org/metadata.json.sha512 (Could not resolve hostname)
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://192.168.56.102 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -x php,html,txt,bak,conf,zip,sql,inc,json,log
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url: http://192.168.56.102
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: html,bak,conf,zip,inc,json,php,txt,sql,log
[+] Timeout: 10s
Starting gobuster in directory enumeration mode
/.html (Status: 403) [Size: 279]
/.php (Status: 403) [Size: 279]
/index.html (Status: 200) [Size: 10701]
/index.php (Status: 301) [Size: 0] [--> http://192.168.56.102/]
/login (Status: 302) [Size: 0] [--> http://localhost/wp-login.php]
/login.php (Status: 302) [Size: 0] [--> http://localhost/wp-login.php]
/0 (Status: 301) [Size: 0] [--> http://192.168.56.102/0/]
/wp-content (Status: 301) [Size: 321] [--> http://192.168.56.102/wp-content/]
/admin (Status: 302) [Size: 0] [--> http://localhost/wp-admin/]
/wp-login.php (Status: 200) [Size: 3987]
/license.txt (Status: 200) [Size: 19915]
/wp-includes (Status: 301) [Size: 322] [--> http://192.168.56.102/wp-includes/]
/wp-register.php (Status: 301) [Size: 0] [--> http://localhost/wp-login.php?action=register]
/wp-rss2.php (Status: 301) [Size: 0] [--> http://localhost/index.php/feed/]
/readme.html (Status: 200) [Size: 7409]


[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: html,bak,conf,zip,inc,json,php,txt,sql,log
[+] Timeout: 10s
Starting gobuster in directory enumeration mode
/.html (Status: 403) [Size: 279]
/.php (Status: 403) [Size: 279]
/index.html (Status: 200) [Size: 10701]
/index.php (Status: 301) [Size: 0] [--> http://192.168.56.102/]
/login (Status: 302) [Size: 0] [--> http://localhost/wp-login.php]
/login.php (Status: 302) [Size: 0] [--> http://localhost/wp-login.php]
/0 (Status: 301) [Size: 0] [--> http://192.168.56.102/0/]
/wp-content (Status: 301) [Size: 321] [--> http://192.168.56.102/wp-content/]
/admin (Status: 302) [Size: 0] [--> http://localhost/wp-admin/]
/wp-login.php (Status: 200) [Size: 3987]
/license.txt (Status: 200) [Size: 19915]
/wp-includes (Status: 301) [Size: 322] [--> http://192.168.56.102/wp-includes/]
/wp-register.php (Status: 301) [Size: 0] [--> http://localhost/wp-login.php?action=register]
/wp-rss2.php (Status: 301) [Size: 0] [--> http://localhost/index.php/feed/]
/readme.html (Status: 200) [Size: 7409]
/robots.txt (Status: 200) [Size: 109]
/' (Status: 301) [Size: 0] [--> http://192.168.56.102/]
/dashboard (Status: 302) [Size: 0] [--> http://localhost/wp-admin/]
/%20 (Status: 301) [Size: 0] [--> http://192.168.56.102/]
/wp-trackback.php (Status: 200) [Size: 135]
/wp-admin (Status: 301) [Size: 319] [--> http://192.168.56.102/wp-admin/]
/wp-atom.php (Status: 301) [Size: 0] [--> http://localhost/index.php/feed/atom/]
/wp-commentsrss2.php (Status: 301) [Size: 0] [--> http://localhost/index.php/comments/feed/]
/xmlrpc.php (Status: 405) [Size: 42]
/wp-rdf.php (Status: 301) [Size: 0] [--> http://localhost/index.php/feed/rdf/]
/wp-rss.php (Status: 301) [Size: 0] [--> http://localhost/index.php/feed/]
/Oasis - 'Definitely Maybe' (Status: 301) [Size: 0] [--> http://192.168.56.102/Oasis%20-%20%27Definitely%20Maybe]
/wp-feed.php (Status: 301) [Size: 0] [--> http://localhost/index.php/feed/]
/wp-signup.php (Status: 302) [Size: 0] [--> http://localhost/wp-login.php?action=register]
/.html (Status: 403) [Size: 279]
/.php (Status: 403) [Size: 279]
/! (Status: 301) [Size: 0] [--> http://192.168.56.102/]
/Bling! (Status: 301) [Size: 0] [--> http://192.168.56.102/Bling]
/Check Screenshots! (Status: 301) [Size: 0] [--> http://192.168.56.102/Check%20Screenshots]
/Check All Tracker Features! (Status: 301) [Size: 0] [--> http://192.168.56.102/Check%20All%20Tracker%20Features]
/yahoo! (Status: 301) [Size: 0] [--> http://192.168.56.102/yahoo]
/Welcome! (Status: 301) [Size: 0] [--> http://192.168.56.102/Welcome]
Progress: 964304 / 964315 (100.00%)
Finished
┌──(kali㉿kali)-[~]