Proyecto Final De Ciberseguridad

Francisco Félix Rodríguez Pellicer

Índice

-Introducción
-Herramientas y Entornos
-Primera Fase: Reconocimiento
-Segunda Fase: Análisis de Vulnerabilidades
-Tercera Fase: Mitigación
-Conclusión

INTRODUCCIÓN

-Este informe, es un documento con toda la información recabada de una máquina Debian, proporcionada por 4Geeks, una vez se a analizado tras una intrusión a esta misma. El objetivo principal ha sido detectar la intrusión del atacante y analizar vulnerabilidades adicionales del sistema.

-Los objetivos de esta practica son:

-Identificar evidencias del atacante.

-Comprobar vulnerabilidades existentes y configuraciones débiles.

-Documentar los hallazgos con sus mitigaciones.

-Acciones correctivas para los distintos puntos de vulnerabilidades y fallos encontrados.

Herramientas y Entornos

-Durante los procesos de análisis y explotación de vulnerabilidades de la máquina comprometida, se utilizaron las siguientes herramientas:

-Nmap
-Wpscan
-Comandos y terminal linux

-El análisis se hizo sobre la máquina virtual Debian, la cual fue atacada, y contaba con diferentes riesgos, con el objetivo de localizar la persona que accedió, además de

averiguar de que forma pudo acceder, junto con distintos fallos y errores que tenían partes de la configuración de seguridad de esta máquina. Todo esto se hizo desde una segunda máquina kali, desde la que se recabo toda la información pertinente para el caso.

Primera fase: Reconocimiento

-Una vez tenemos la máquina Debian (Atacada) y la máquina Kali (Análisis y Reconocimiento), hacemso un reconocimiento de la máquina Debian, en la que descubrimos lo siguiente:

-Lo primero fue que al usar la herramienta de nmap se encontraron los puertos 21(FTP), 21(SSH), y 80(HTTP) abiertos con sus respectivas versiones, y las posibles vulnerabilidades de estas mismas, por ejemplo:

-FTP: Versión vsftpd 3.0.3, en esta versión, se debería comprobar si el acceso anónimo esta activo o mal configurado, ya que es uno de los posibles puntos de acceso.

-SSH: Versión OpenSSH 9.2p1, se debe comprobar si esta permitido el acceso directo al login por root (otra posible forma de acceso a la máquina).

-HTTP: Versión Apache httpd 2.4.62, para obtener mayor detalle de este puerto, se deberia hacer un análisis eb

profundidad para detectar distintas vulnerabilidades web como puedan ser inyecciones, XSS o LFI entre otros.

-Una vez analizados los puertos y sabiendo que servicios activos tenia, pasé a una examinación más profunda de distintos apartados de la máquina, como logs, el historial de comandos ejecutados por root en el sistema, si hubo accesos de forma remota y de que formas, además de revisar permisos de ciertas carpetas importantes.

-Para empezar, se examinó el historial mencionado previamente, en el que se descubrió que hubo una creación de un nuevo usuario en MySQL, **wordpressuser**, que además se creo con una contraseña extremadamente débil, **123456**, en texto plano, además de todo eso, este usuario tiene privilegios completos sobre la base de datos junto con el usuario user, que a su vez, este tiene la opción de otorgar privilegios a otros usuarios.

-Una vez habiendo terminado de examinar esa parte, se paso a la identificación de los accesos de forma remota de ese equipo, mediante la terminal, se consiguió encontrar al que parecía ser el atacante, el cual accedió de forma remota por el root, desde la **ip 192.168.0.134**, junto con el puerto desde el que accedio, y la fecha y hora del mismo, ya sabiendo la forma en la que accedió al sistema, se cambió la configuración del archivo **sshd_config** y se cambio el acceso de forma remota por root al mismo, para evitar que se pueda volver a acceder de la misma forma y evitar futuros riesgos por esta parte.

-Por último, se examinaron los permisos de las carpetas raiz de la instalación de **Wordpress** en el servidor: **/ver/www/html**, gracias a esto se pudo ver que las carpetas y archivos, tenían todos los permisos concedidos, además de los permisos excesivos se encontraron contraseñas en la base de datos expuesta del archivo **wp-config.php**, con lo que se ajustaron los permisos en base de si eran carpetas o archivos.

Segunda fase: Análisis de vulnerabilidades

-Una vez terminada la primera fase, pasamos al análisis de vulnerabilidades, como ya se hizo un escaneo de los puertos y versiones, sabemos que están:

-Puerto 21 – FTP

-Puerto 22 – SSH

-Puerto 80 – HTTP

-Puerto 3306 - MySQL

-Sabiendo los puertos, servicios y versiones descubrimos algunas vulnerabilidades como:

Vulnerabilidad FTP

-Tras el primer análisis de puertos descubrimos que existía un puerto FTP abierto con la versión **vsftpd 3.0.3**, que está configurado con un acceso anónimo habilitado, y que permite tanto el acceso remoto a traves del cliente FTP, como comandos de escritura, que permiten la autenticación sin credenciales de esta, mediante el usuario **anonymous**, con el que podemos acceder sin autenticación con una conexión remota al servidor con **ftp 10.0.2.20.**

-Para verificar esto se puede acceder al archivo **/etc/vsftpd.conf** desde la propia Debian para verificar que estan habilitados para acceso anónimo y escritura. Además de esto si hacemos un análisis más intensivo, podemos apreciar otras dos vulnerabilidades extras:

FTP	CVE	Peligro Potencial	Descripción
	2021-30047	7,5	Permite la ejecución de comandos remotos
	2021-3618	7,4	Fallo en la seguridad que puede llegar a comprometer la integridad del servicio

-Tras ver estas vulnerabilidades más la ya mencionada previamente, se puede apreciar que:

-Faltan actualizaciones en el servidor ya que parece no estar actualizado.

-Un atacante con acceso a la red sería capaz de localizar estas vulnerabilidades, junto con la anterior, para acceder de forma no autorizada y ejecutar código de forma remota, lo que representa un peligro altamente potencial si no se soluciona rápido.

Vulnerabilidades en HTTP

-Para el escaneo de vulnerabilidades en el servicio **HTTP**, empleamos la herramienta de **wpscan**, tras un escaneo, podemos encontrar cosas tales como:

-El servidor y la versión del mismo (**Apache/2.4.62** (Debian))

-Documentos/archivos .txt y nos revelan rutas internas como pueden ser: **/wp-admin/admin-ajax.php** o **/wp-admin.**

-Archivos **readme.html** que confirman la instalación de **Wordpress.**

-Un directorio que permite navegar por archivos subidos en la ruta **/wp-content/uploads**

-Además de todo esto se encuentra una interfaz XML-RPC habilitada lo que si es explotado puede llegar a ataques DdoS, Fuerza bruta, Pingback attack entre otros.

-Tras haber identificado estas vulnerabilidades, accedemos de forma remota desde una máquina Kali por el navegador a una de las rutas descubiertas previamente: **10.0.2.20/wp-content/uploads/**

-Al acceder podemos ver varias cosas, la primera, un directorio con carpetas creadas por **WordPress** de forma automática para agrupar los distintos tipos de archivos (en este caso entre los años 2024/2025), junto con fechas y la estructura que tienen dichas carpetas, lo que puede llevar a ataques fácilmente dirigido a archivos sensible o confidenciales, ademas del uso de **shells** si el entorno permite la carga y ejecución de las mismas. Además de lo comentado previamente, al acceder a alguna subcarpeta, podemos llegar a un apartado en el que encontramos un archivo **functions.php**, que si en el se encuentran ciertas funciones, pueden ser explotadas si existe otro vector de entrada para **carga de archivos o XSS**, ya no solo por lo comentado, sino que esta exposición facilita la identificación de la versión del tema, lo que puede llevar a más CVEs o exploids conocidos.

-Para finalizar esta parte, durante la primera fase de reconocimiento se detecto que el navegador tenía almacenadas credenciales de acceso al panel de administración del **WordPress**, lo que nos permitió el acceso sin la necesidad de usar un ataque de fuerza bruta.

-Usuario: wordpress-user

-Contraseña: wordpressuser123456

-Con esto, los atacantes tienen la posibilidad de crear cuentas, instalar plugins, subir código, modificar contenido, etc, en resumidas cuentas acceso completo al **Backend** de este Wordpress.

Tercera fase: Mitigación

-Una vez finalizado el reconocimiento y la explotación de vulnerabilidades, voy a proponer ciertas soluciones en base a los distintos puntos o partes tratadas:

Puertos y servicios:

-Cierre de puertos innecesarios y eliminación de servicios innecesarios.

-Aplicación de iptables y UFW para el firewall o reglas más restrictivas, ademas de bloquear la ip del atacante.

FTP:

-Desactivación de acceso anónimo.

-Uso e implementación de SFTP para mayor seguridad.

-Limitar o ajustar el acceso con permisos chroot.

MySQL:

-Uso de contraseñas únicas y robustas.

-Limitación del acceso remoto.

-Implementación de cifrado en backups y conexiones.

SSH:

-Deshabilitación del root login.

-Cambio del puerto por defecto y limitación de IPs permitidas.

-Uso de autenticación por clave pública

Permisos de carpetas y archivos wp-config:

-Restringir permisos con chmod.

-Ajuste de permisos de directorios y carpetas según la necesaria relevancia de los mismos.

-Uso de copias de seguridad cifradas.

-Mover wp-config.php fuera del root del sitio web.

-Uso y monitorización de agentes de Wazuh.

Web y navegador:

-Uso de plugins de seguridad en Wordpress.

-Restricción de acceso a directorios.

-Revisión y limpieza de archivos expuestos.

-Deshabilitación de listado de directorios.

-Cambio de credenciales a unas más seguras, junto con autentificación en 2 pasos.

-Configurar políticas de expiración de sesiones.

Conclusión

-Tras un análisis exhaustivo de la máquina Debian comprometida, se han identificado múltiples vulnerabilidades críticas que comprometían gravemente la seguridad del sistema. Entre ellas destacan accesos remotos no autorizados mediante credenciales débiles, servicios expuestos sin la debida configuración de seguridad, y permisos inapropiados sobre archivos sensibles.

-Gracias al uso de herramientas como Nmap y WPScan, se logró realizar una detección precisa de los vectores de ataque y los posibles métodos de explotación. Asimismo, se plantearon soluciones prácticas y específicas para mitigar cada riesgo identificado, alineadas con buenas prácticas en ciberseguridad.

-Este proyecto pone de manifiesto la importancia de una gestión proactiva de la seguridad informática, el mantenimiento actualizado de los sistemas, y la implementación de políticas de hardening, monitoreo continuo y respuesta ante incidentes. Aplicar las recomendaciones aquí descritas no solo evitará futuras intrusiones, sino que fortalecerá de forma significativa la postura de seguridad de cualquier infraestructura similar.