REPORTE DE VULNERABILIDAD

Fecha: 19 de febrero de 2025
Autor: Marcos Maldonado Eremia.
Objetivo: Evaluar la seguridad de la máquina con IP 192.168.1.10 mediante un escaneo con Nmap para identificar puertos abiertos, servicios activos y posibles vulnerabilidades.

---

1. Instalación y Uso de Nmap

Para realizar el escaneo, primero nos aseguramos de que Nmap esté instalado en la máquina Kali Linux:

"sudo apt-get install nmap"

Una vez instalado, ejecutamos los siguientes comandos para el análisis:

2. Escaneo de Puertos y Servicios

Se ejecutó un escaneo básico a la IP objetivo:

"nmap 192.168.1.10"

Resultado: Se identificó que el host está activo y tiene servicios corriendo.

Para identificar los servicios y versiones, ejecutamos:

"nmap -sV 192.168.1.10"

Resultados obtenidos:

- **Puerto 80/tcp abierto**
- **Servicio: Apache HTTPD 2.4.62 (Debian)**
- **Dirección MAC: 08:00:27:D1:65:C7 (VirtualBox NIC)**

3. Escaneo de Vulnerabilidades

Se realizó un escaneo con scripts de detección de vulnerabilidades:

"nmap -sV --script=vuln 192.168.1.10"

Resultados:

- Se detectaron versiones de WordPress antiguas en varias rutas:
    - WordPress 2.2, 2.5, 2.6, 2.7
    - WordPress principal en versión 6.7.1
- No se encontraron vulnerabilidades CSRF o XSS conocidas en la versión de Apache detectada.

Captura de pantalla del escaneo:

4. Búsqueda de Vulnerabilidades

Para los servicios detectados, se realizó una búsqueda en bases de datos públicas:

- **Apache 2.4.62:**
    - Se recomienda revisar en la NVD: https://nvd.nist.gov/
    - También en CVE Details: https://www.cvedetails.com/
- **WordPress versiones antiguas (2.2 - 2.7):**
    - Históricamente, estas versiones han sido vulnerables a ataques de ejecución remota de código y XSS.
    - Se recomienda verificar en la Exploit Database: https://www.exploit-db.com/

---

5. Conclusiones y Recomendaciones

- **Actualizar Apache HTTPD** a la última versión estable.
- **Actualizar WordPress** a la versión más reciente para mitigar vulnerabilidades conocidas.
- **Habilitar un firewall** para restringir el acceso a puertos no utilizados.
- **Revisar los permisos de acceso** en las páginas de administración de WordPress.
- **Monitorear registros de acceso** para detectar posibles intentos de explotación.
