**Ingeniería de Seguridad del Software.**

**Práctica de Laboratorio No. 1:**

**Análisis de Amenazas y vulnerabilidades en un servidor Web utilizando Kali Linux en un VNE controlado.**

**Objetivo:**

El objetivo de esta actividad es introducir a los estudiantes en el proceso de identificación de amenazas, vulnerabilidades y riesgos en un entorno controlado, utilizando Kali Linux y un escenario de red virtualizado en VirtualBox.

**Requerimientos:**

1. **Kali Linux** (instalado en una máquina virtual en VirtualBox).
2. **Equipo servidor vulnerable** (puede ser una máquina vulnerable como **Metasploitable2** o **OWASP Broken Web Applications**). También se pueden usar sistemas más sencillos como un Linux server con servicios comunes (HTTP, SSH).
3. **Equipo cliente,** puede ser Ubuntu desktop 24.04.2 LTS.
4. **VirtualBox** o cualquier otra herramienta de virtualización para gestionar las máquinas virtuales.
5. Acceso a Internet para descarga de herramientas y recursos.

**Entorno Virtual de Red:**

* **Máquina 1: Kali Linux** (máquina atacante)
* **Máquina 2: Ubuntu Server -** Máquina vulnerable (Metasploitable2 o un servidor web vulnerable)
* **Máquina 3: Ubuntu Desktop 24.04.2 LTS.**

**Desarrollo:**

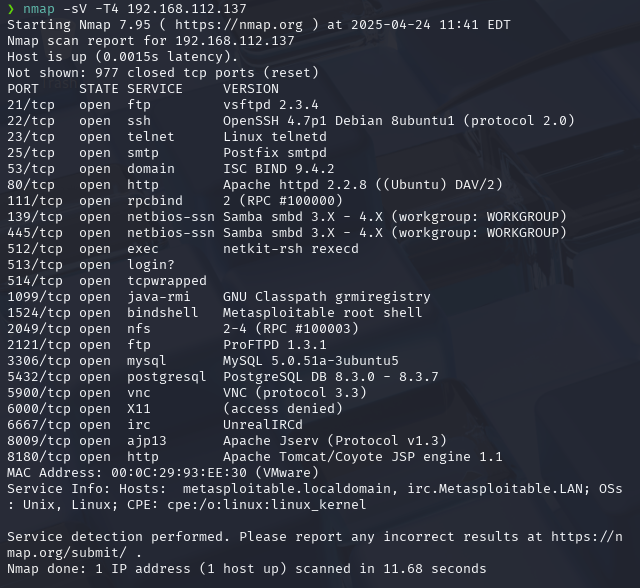
**1. Reconocimiento de red**

En Kali Linux, vamos a realizar un escaneo de la máquina vulnerable para identificar los servicios abiertos, lo que nos ayudará a identificar posibles puntos débiles.

**1.1 Escaneo de puertos con Nmap**

En Kali Linux, abre una terminal y utiliza Nmap para realizar un escaneo de puertos:

$ nmap -sV -T4 < 192.168.112.137 >



**Explicación:**

* **Amenaza:** Los servicios en la máquina vulnerable son posibles vectores de ataque.
* **Vulnerabilidad:** Algunos servicios, como HTTP o SSH, pueden estar desactualizados o mal configurados, lo que los hace vulnerables a ataques.
* **Riesgo:** Si un atacante logra explotar una vulnerabilidad en uno de esos servicios, podría comprometer la máquina vulnerable.

**Referencia Nmap:**

* Nmap Documentation: <https://nmap.org/docs.html>

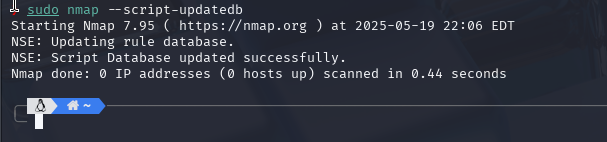
**2. Identificación de vulnerabilidades**

* En esta etapa, utilizaremos **Nmap** (con sus scripts NSE) para buscar vulnerabilidades conocidas en los servicios detectados durante el escaneo previo. De este modo, no emplearemos OpenVAS, sino que aprovecharemos las capacidades integradas de Nmap para auditoría de seguridad.

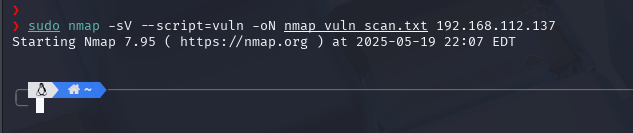
### **2.1 Uso de Nmap para detección de vulnerabilidades**

1. **Preparar entorno en Kali Linux**   
    – Verifica que Nmap esté instalado (por defecto Kali lo incluye).  
    – Actualiza la base de datos de scripts NSE para asegurarte de contar con las últimas comprobaciones:

$ sudo nmap --script-updatedb



* Esto sincroniza los scripts locales con la versión más reciente disponible en el repositorio de NSE.
* **Escaneo de versiones y vulnerabilidades con el agrupador “vuln”**   
   Ejecuta el siguiente comando desde la máquina Kali (192.168.112.133) apuntando a Metasploitable (192.168.112.137):

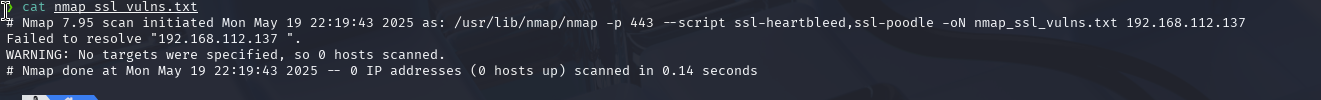


* -sV → detecta la versión de cada servicio en puertos abiertos.
* --script=vuln → ejecuta todos los scripts NSE categorizados como “vuln” (comprobaciones automáticas de vulnerabilidades conocidas de servicios comunes).
* -oN nmap\_vuln\_scan.txt → guarda el resultado en un fichero de texto plano para su análisis posterior.

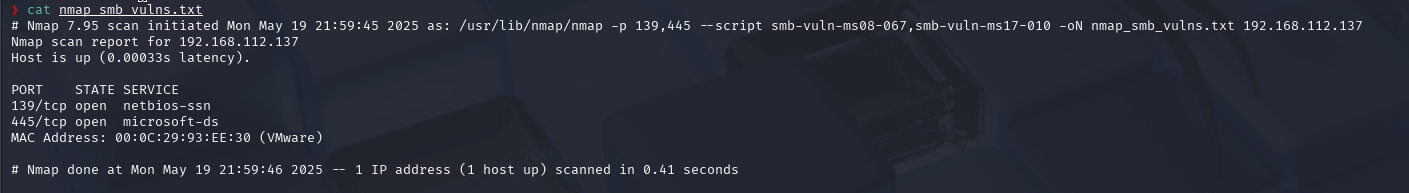
**Escaneos dirigidos a vulnerabilidades específicas**   
 Además del agrupador general --script=vuln, puedes invocar scripts NSE individuales o por categoría para auditar servicios concretos:

* **Comprobación de SSL/TLS (Heartbleed, Poodle, etc.)**

$ sudo nmap -p 443 --script ssl-heartbleed,ssl-poodle -oN nmap\_ssl\_vulns.txt 192.168.112.137

* 
* -p 443 → escanea únicamente el puerto HTTPS.
* ssl-heartbleed → detecta si el servicio es vulnerable a Heartbleed (CVE-2014-0160).
* ssl-poodle → verifica si hay vulnerabilidad POODLE en SSLv3.
* **Verificación de SMB (MS08-067, MS17-010/EternalBlue, etc.)**

$ sudo nmap -p 139,445 --script smb-vuln-ms08-067,smb-vuln-ms17-010 -oN nmap\_smb\_vulns.txt 192.168.112.137

* 
* smb-vuln-ms08-067 → detecta la vulnerabilidad que permitió Stuxnet (CVE-2008-4250).
* smb-vuln-ms17-010 → comprueba si está expuesto a EternalBlue (CVE-2017-0144).
* **Revisión de aplicaciones web en HTTP (WordPress, Joomla, Apache Struts, etc.)**

**3. Explotación de una vulnerabilidad con Metasploit**

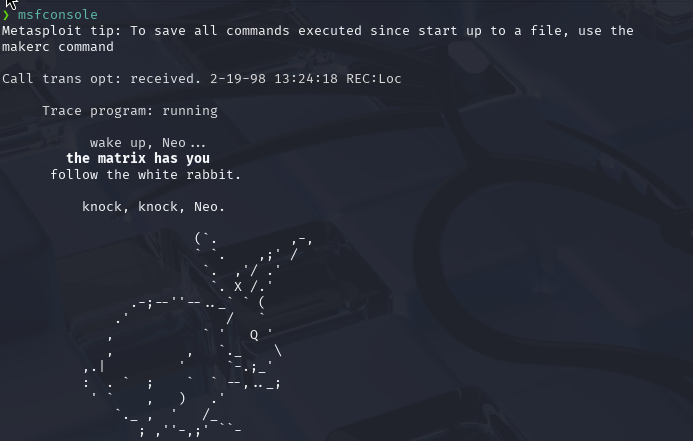
Una vez identificadas las vulnerabilidades, vamos a intentar explotarlas usando **Metasploit**, una herramienta de explotación automatizada.

**3.1 Uso de Metasploit para explotar una vulnerabilidad**

Por ejemplo, si descubrimos que un servicio HTTP es vulnerable a un **Remote Code Execution (RCE)**, podemos usar Metasploit para explotarlo.

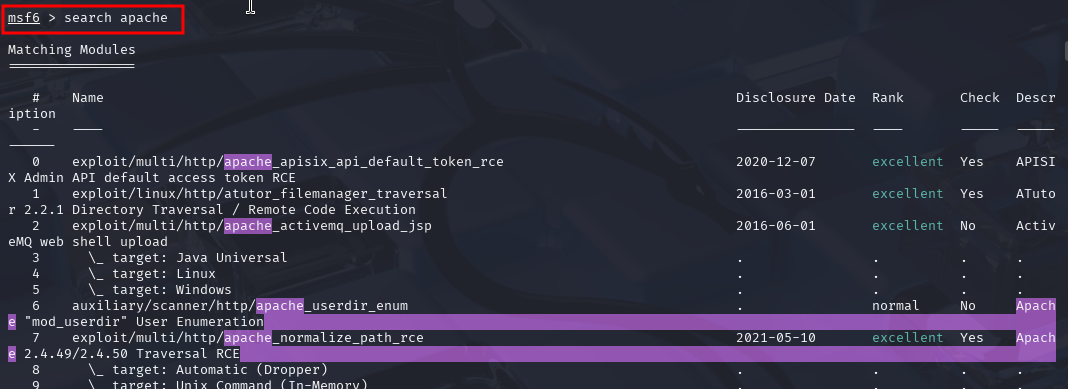
1. Inicia Metasploit en Kali Linux:

$ msfconsole



1. Busca un exploit relacionado con el servicio identificado (por ejemplo, un servicio HTTP):

$ search apache



1. Elige un exploit y configúralo:

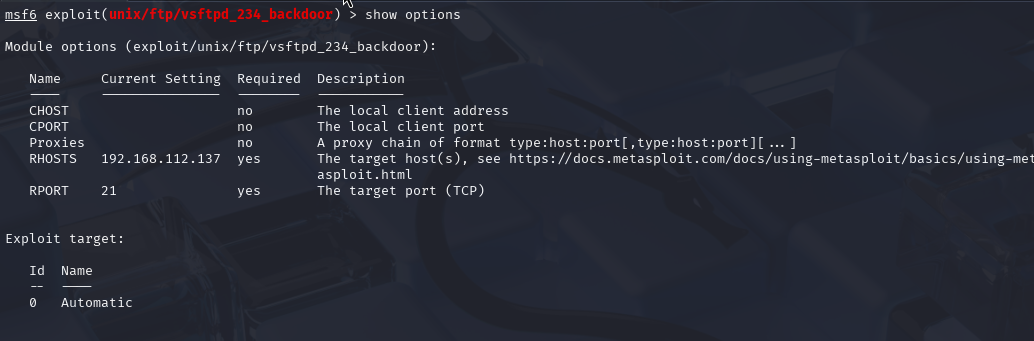
$ use exploit/unix/webapp/php\_cgi\_arg\_injection

$ set RHOST <IP\_de\_la\_máquina\_vulnerable>

$ set RPORT 80

$ run





**Explicación:**

* **Amenaza:** Si el atacante explota la vulnerabilidad, puede ejecutar comandos remotos en el servidor.
* **Vulnerabilidad:** El servidor web tiene una vulnerabilidad de ejecución remota de código debido a una mala configuración.
* **Riesgo:** El riesgo es que el atacante pueda tomar control total del servidor.

**Referencia Metasploit:**

* Metasploit Documentation: <https://docs.metasploit.com/>

1. **Investigar:**

En lugar de Metasploit, realiza la explotación de una vulnerabilidad con Nikto**.**

**Nikto documentation:** <https://www.kali.org/tools/nikto/>

1. **Discusión y conclusión**

Al final de la actividad, realiza una breve discusión sobre los resultados obtenidos:

1. **¿Qué amenazas se identificaron?**  
   (Por ejemplo, la posibilidad de que un atacante ejecute código remoto en el servidor web).
2. **¿Qué vulnerabilidades fueron explotadas?**  
   (Por ejemplo, un servicio web vulnerable a inyecciones de comandos o ejecución remota de código).
3. **¿Cuál es el riesgo de no mitigar estas vulnerabilidades?**  
   (Si no se corrige la vulnerabilidad, un atacante puede obtener acceso completo al sistema y ejecutar comandos maliciosos).
4. **¿Cuál herramienta es mejor, NMAP, Metasploit, OpenVAS o Nikto?** Argumenta tu respuesta con el alcance y objetivo que persigue cada una.

**Resumen de la Actividad:**

* **Amenaza:** Es cualquier posible evento o situación que podría comprometer la seguridad de la red o los sistemas.
* **Vulnerabilidad:** Es una debilidad que puede ser explotada por una amenaza, como un servicio desactualizado o una mala configuración.
* **Riesgo:** Es la probabilidad de que una amenaza explote una vulnerabilidad, lo que puede resultar en daño al sistema o acceso no autorizado.

**Conclusiones:**

Los participantes habrán aprendido cómo identificar, analizar y explotar vulnerabilidades en una red, comprendiendo la relación entre las amenazas, las vulnerabilidades y los riesgos. Utilizando herramientas como Nmap, OpenVAS y Metasploit y Nikto en Kali Linux, habrán ganado experiencia práctica en pruebas de penetración y evaluación de seguridad en un entorno controlado.

**Referencias Bibliográficas:**

* OpenVAS: The Open Vulnerability Assessment System. (2012). Syngress Publishing. ISBN: 978-1-59749-574-5.
* Nikto: A Web Server Scanner. (2010). Packt Publishing. ISBN: 978-1-84951-019-3.
* The Penetration Tester's Guide. (2011). No Starch Press. ISBN: 978-1-59327-288-3
* Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. (2009). Insecure Publishing. ISBN: 978-0-9799587-1-7.
* Nmap in the Enterprise: Your Guide to Network Scanning. (2008). Syngress Publishing. ISBN: 978-0-08-055874-5.