

Escáner de puertos abiertos utilizando Kali Linux

Autor: Cristhian Daniel Mauna Ledezma

Curso: Ciberseguridad nivel explorador

Popayán 2025

Introducción:

Se presentará un escáner de puertos abiertos utilizando Kali Linux, Python 3, nmap y una máquina virtual de metasploitable que es un sistema operativo de Linux vulnerable para simular ataques

El escáner de puertos abiertos es importante en ciberseguridad porque permite identificar los servicios que están activos y accesibles en un sistema o red.

Las herramientas que se van a utilizar son virtual box que es donde se virtualizaron los sistemas (Kali Linux y metasploitable) Python 3, nmap y subl.text que es el editor de texto que es donde monte el código para ejecutarlo.

Justificación

Detección de servicios expuestos

Cada puerto abierto puede estar asociado a un servicio (como HTTP, FTP, SSH, etc.). Si un puerto está abierto, significa que ese servicio está disponible y puede ser un punto de entrada para un atacante.

Marco teórico

Los puertos abiertos permiten la comunicación entre distintos dispositivos, pero también pueden ser de provecho por diferentes atacantes si están mal configurados. Las herramientas como Nmap permiten identificar servicios en funcionamiento y sus posibles vulnerabilidades que están expuestas. El escaneo syn, permite detectar puertos abiertos sin establecer una conexión completa reduciendo la visibilidad del escaneo.

Metodología

Se utilizaron dos máquinas virtuales una Kali Linux que es donde se va a montar todo el proceso. La herramienta principal es Nmap, completada con un script en Python 3 utilizando la librería de Python-nmap y la otra máquina es la de metasploitable que es donde va a ir dirigido el ataque y me despliegue que puertos están abiertos.



```
kali-linux-2025.1c-virtualbox-amd64 [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

(nmap_env)root@kali: /home/kali

Archivo  Acciones  Editar  Vista  Ayuda

(kali@kali)~$ sudo su
[sudo] contraseña para kali:
(kali@kali)~$ source nmap_env/bin/activate
(nmap_env)~$ pip install python-nmap
Requirement already satisfied: python-nmap in ./nmap_env/lib/python3.13/site-packages (0.7.1)
(nmap_env)~$
```

Desarrollo del escáner

Se diseñó un script en Python que automatiza el uso de Nmap para escanear los puertos abiertos a continuación se mostrara el código completo.

```

1  #!/usr/bin/python3
2  #importacion de los modulos
3  import nmap #es la herramienta nmap desde python a través de su libreria
4  import tkinter as tk #añadi esta función para crear la interfaz grafica
5  from tkinter import messagebox, scrolledtext #se agrega un área de texto con scroll para mostrar resultados
6
7  #Funcion principal: escanear()
8  def escanear():
9      ip = entry_ip.get() #se ejecuta cuando el usuario hace clic en el boton "Escanear"
10     if not ip: #Validacion de la entrada
11         messagebox.showerror("Error", "Por favor ingresa una dirección IP") #valida que esté vacío
12         return
13
14     #Funcion de los puertos escanear con Nmap
15     try:
16         nm = nmap.PortScanner()
17         results = nm.scan(hosts=ip, arguments="-sT -n -Pn -T4")
18
19         salida.delete(1.0, tk.END) # Limpiar resultados anteriores
20         salida.insert(tk.END, f"Escaneando IP: {ip}\n")
21         salida.insert(tk.END, f"Estado del host: {nm[ip].state()}\n")
22
23         puertos_abiertos = [] #procesa protocolos y puertos
24         for proto in nm[ip].all_protocols():
25             salida.insert(tk.END, f"\nProtocolo: {proto}\n") #muestra resultados basicos
26             lport = sorted(nm[ip][proto].keys())
27             for port in lport:
28                 estado = nm[ip][proto][port]["state"]
29                 salida.insert(tk.END, f"Puerto: {port}\tEstado: {estado}\n")
30                 if estado == "open":
31                     puertos_abiertos.append(str(port)) #guarda los puertos abiertos en una lista
32
33             if puertos_abiertos: #muestra resumen de los puestos abiertos
34                 salida.insert(tk.END, f"\nPuertos abiertos: -p {'.'.join(puertos_abiertos)} {ip}\n")
35             else:
36                 salida.insert(tk.END, "\nNo se encontraron puertos abiertos.\n")
37     except Exception as e: #manejo de errores
38         messagebox.showerror("Error", str(e))
39
40     # Crear ventana principal creacion de la interfaz gráfica
41     ventana = tk.Tk()
42     ventana.title("Escáner de Puertos con Nmap") # se le asigna un titulo
43     ventana.geometry("600x400") # se da el tamaño de la ventana
44
45     # Etiqueta e input para IP
46     tk.Label(ventana, text="IP Objetivo:").pack(pady=5)
47     entry_ip = tk.Entry(ventana, width=40)
48     entry_ip.pack(pady=5)

```

```
# Botón para escanear
btn_escanear = tk.Button(ventana, text="Escanear", command=escanear)
btn_escanear.pack(pady=5)

# Área de resultados
salida = scrolledtext.ScrolledText(ventana, width=70, height=15)
salida.pack(padx=10, pady=10)

# Ejecutar GUI
ventana.mainloop()
```

Para ejecutar este código hay que ir a la línea de comandos en Kali Linux y ejecutar el comando

python3 scan.py

```
(nmap_env)-(root@kali)-[/home/kali]
# python3 scan.py
```

Resultados



```

msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:d8:13:a5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.127/24 brd 192.168.10.255 scope global eth0
    inet6 2800:484:ab89:f000:a00:27ff:fed8:13a5/64 scope global dynamic
        valid_lft 1167671sec preferred_lft 562871sec
    inet6 fe80::a00:27ff:fed8:13a5/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ _

```

```

Puerto: 513 Estado: open
Puerto: 514 Estado: open
Puerto: 1099 Estado: open
Puerto: 1524 Estado: open
Puerto: 2049 Estado: open
Puerto: 2121 Estado: open
Puerto: 3306 Estado: open
Puerto: 5432 Estado: open
Puerto: 5900 Estado: open
Puerto: 6000 Estado: open
Puerto: 6667 Estado: open
Puerto: 8009 Estado: open
Puerto: 8180 Estado: open

Puertos abiertos: -p 21,22,23,25,53,80,111,139,445,512,513,514,1099,15

```

Al hacer el escaneo a la dirección IP 192.168.10.127 que es la dirección IP de la máquina de metasploitable se identificaron múltiples puertos abiertos y servicios asociados como el SSH, HTTP entre otros.

Conclusiones y Recomendaciones

El proyecto permitió entender el funcionamiento y lo que se debe tener en cuenta, la importancia de los puertos en ciberseguridad. Como recomendación actualizar de manera regular los servicios expuestos y limitar el acceso desde redes externas porque al conectarse a este tipo de redes podemos quedar expuestos atacantes y puede ser peligroso.