

## Ejercicio Práctico: Escaneo de Puertos en un Host Local

---

### Descripción

Este ejercicio tiene como objetivo realizar un escaneo básico de puertos abiertos en un host local (por ejemplo, `127.0.0.1`) utilizando **Python** y la librería **python-nmap**. El estudiante aprenderá a integrar Python con Nmap y a interpretar resultados básicos.

---

### Objetivos de aprendizaje

- Entender cómo Python puede automatizar tareas de escaneo con Nmap.
  - Realizar un escaneo básico de puertos en localhost o una IP segura.
  - Interpretar resultados simples: puertos abiertos, servicios y estados.
  - Fortalecer la conciencia ética en pruebas locales y autorizadas.
- 

### Instrucciones

1. Asegúrate de tener Nmap instalado en tu sistema.

Instala la librería **python-nmap** con:

```
pip install python-nmap
```

- 2.
3. Escribe un script en Python que:
  - Reciba como entrada una dirección IP (ej: `127.0.0.1`).

- Ejecute un escaneo básico de puertos del 1 al 1024.
  - Muestre por pantalla los puertos abiertos y su estado.
4. Ejecuta el script solo en entornos controlados, personales o de laboratorio.
5. Analiza los resultados y reflexiona:
- ¿Qué servicios están visibles?
  - ¿Hay puertos abiertos innecesarios?
- 

## Consideraciones Éticas

- Este ejercicio debe realizarse únicamente sobre **tu propia máquina** o en **ambientes de prueba autorizados**.
  - Nunca escanees redes o equipos ajenos sin permiso explícito.
  - El conocimiento en ciberseguridad debe usarse con responsabilidad.
- 

## Ejemplo de salida esperada (resumen):

Escaneando 127.0.0.1...  
Puerto 22: abierto  
Puerto 80: cerrado  
Puerto 443: abierto  
Escaneo completo.


---

## Solución: **escaneo\_basico.py**

```
import nmap

# Crear un escáner de Nmap
scanner = nmap.PortScanner()

# Dirección IP a escanear
ip_objetivo = "127.0.0.1"

print(f" Escaneando el host: {ip_objetivo}...\n")
```

```
# Ejecutar escaneo básico en el rango de puertos 1 al 1024
scanner.scan(ip_objetivo, '1-1024')
```

```
# Verificar si el host está activo
```

```
if scanner.all_hosts():
```

```
    for host in scanner.all_hosts():
```

```
        print(f"Resultados para {host}:")
```

```
        for protocolo in scanner[host].all_protocols():
```

```
            puertos = scanner[host][protocolo].keys()
```

```
            for puerto in sorted(puertos):
```

```
                estado = scanner[host][protocolo][puerto]['state']
```

```
                print(f" - Puerto {puerto}: {estado}")
```

```
else:
```

```
    print("❌ No se detectó ningún host activo.")
```

```
print("\n✅ Escaneo completado.")
```

---



### Posible salida en consola:



Escaneando el host: 127.0.0.1...

Resultados para 127.0.0.1:

- Puerto 22: open
- Puerto 80: closed
- Puerto 631: open



Escaneo completado.

---



### Explicación técnica

- Se usa la librería `python-nmap` para ejecutar el comando Nmap desde Python.
  - El escaneo se realiza sobre el rango de puertos comunes (1 al 1024).
  - Se imprime una lista con los puertos abiertos y cerrados para el protocolo TCP.
  - El script se limita a `localhost` para garantizar seguridad y legalidad.
- 



### Reflexión ética y técnica

- ¿Necesitas que estos servicios estén abiertos?
  - ¿Están debidamente protegidos por autenticación o firewall?
  - Automatizar escaneos puede ayudarte a detectar errores de configuración antes de que lo haga un atacante.
-