Lab 7

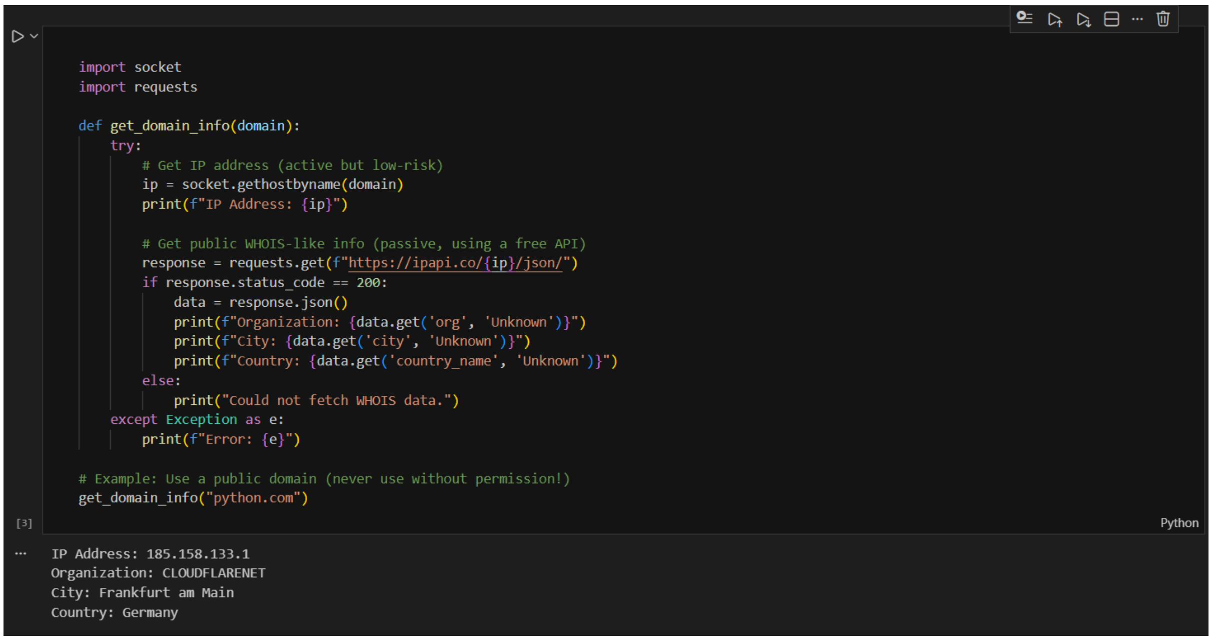
The purpose of this lab was to introduce the **core concepts of penetration testing (pen testing)** by focusing on the **early stages of an attack**, specifically:

- Reconnaissance (information gathering)
- Scanning and enumeration

Instead of attacking real systems, all testing was carried out in a **legal, ethical, and controlled environment**, using only:

- Authorised targets
- Localhost
- Public test domains

This reflects how **ethical hackers work professionally** before any exploitation takes place.

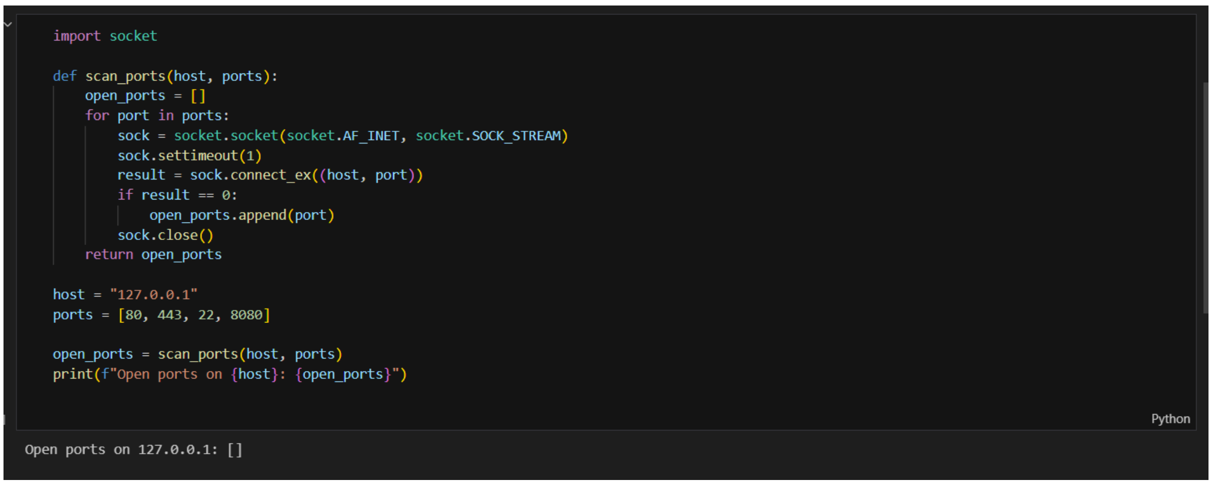


```python
import socket
import requests

def get_domain_info(domain):
    try:
        # Get IP address (active but low-risk)
        ip = socket.gethostbyname(domain)
        print(f"IP Address: {ip}")

        # Get public WHOIS-like info (passive, using a free API)
        response = requests.get(f"https://ipapi.co/{ip}/json/")
        if response.status_code == 200:
            data = response.json()
            print(f"Organization: {data.get('org', 'Unknown')}")
            print(f"City: {data.get('city', 'Unknown')}")
            print(f"Country: {data.get('country_name', 'Unknown')}")
        else:
            print("Could not fetch WHOIS data.")
    except Exception as e:
        print(f"Error: {e}")

# Example: Use a public domain (never use without permission!)
get_domain_info("python.com")
```

```
IP Address: 185.158.133.1
Organization: CLOUDFLARENET
City: Frankfurt am Main
Country: Germany
```

Whois input and output IP and info

Lab 7

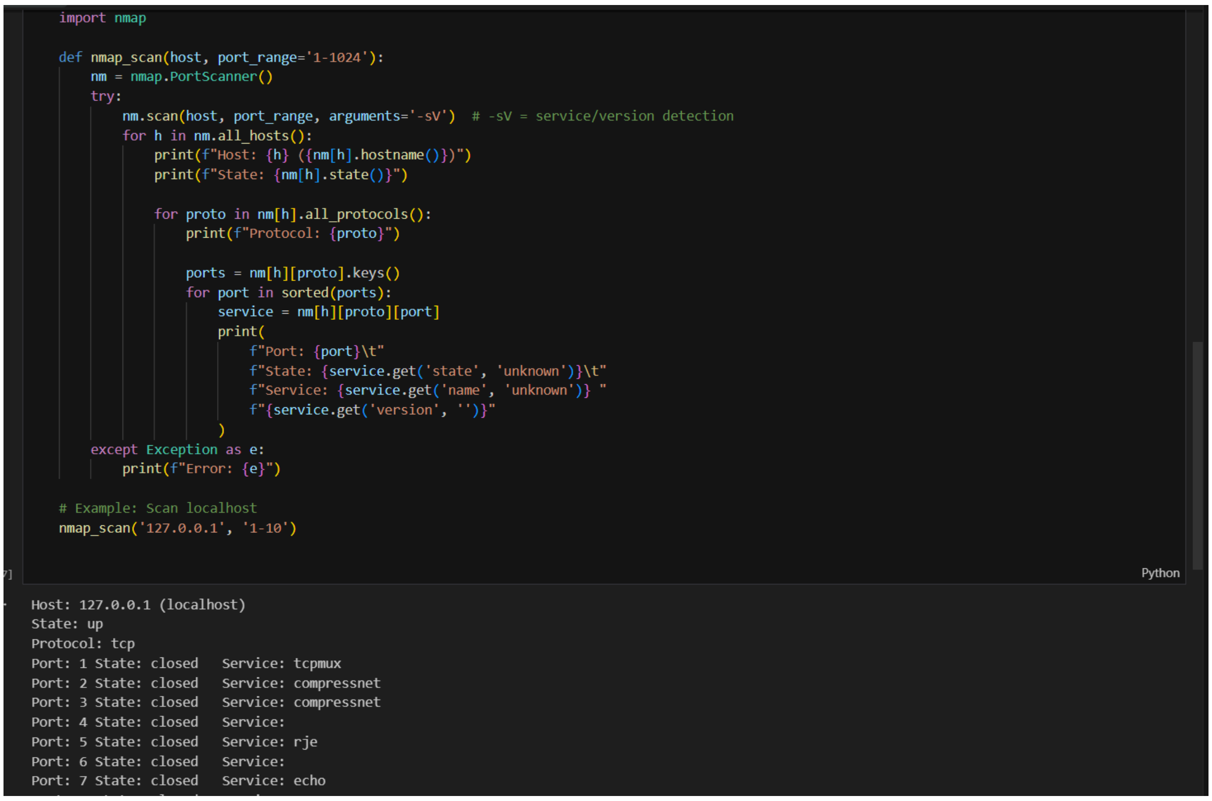


```python
import socket

def scan_ports(host, ports):
    open_ports = []
    for port in ports:
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.settimeout(1)
        result = sock.connect_ex((host, port))
        if result == 0:
            open_ports.append(port)
        sock.close()
    return open_ports

host = "127.0.0.1"
ports = [80, 443, 22, 8080]

open_ports = scan_ports(host, ports)
print(f"Open ports on {host}: {open_ports}")
```

Python

```
Open ports on 127.0.0.1: []
```

Ports information



```python
import nmap

def nmap_scan(host, port_range='1-1024'):
    nm = nmap.PortScanner()
    try:
        nm.scan(host, port_range, arguments='-sV')  # -sV = service/version detection
        for h in nm.all_hosts():
            print(f"Host: {h} ({nm[h].hostname()})")
            print(f"State: {nm[h].state()}")

            for proto in nm[h].all_protocols():
                print(f"Protocol: {proto}")

                ports = nm[h][proto].keys()
                for port in sorted(ports):
                    service = nm[h][proto][port]
                    print(
                        f"Port: {port}\t"
                        f"State: {service.get('state', 'unknown')}\t"
                        f"Service: {service.get('name', 'unknown')} "
                        f"{service.get('version', '')}"
                    )
    except Exception as e:
        print(f"Error: {e}")

# Example: Scan localhost
nmap_scan('127.0.0.1', '1-10')
```

Python

```
Host: 127.0.0.1 (localhost)
State: up
Protocol: tcp
Port: 1 State: closed   Service: tcpmux
Port: 2 State: closed   Service: compressnet
Port: 3 State: closed   Service: compressnet
Port: 4 State: closed   Service:
Port: 5 State: closed   Service: rje
Port: 6 State: closed   Service:
Port: 7 State: closed   Service: echo
```

NMAP Scanner