COMPUTER NETWORKS

ASSIGNMENT 2

**NAME: ADHESH DHEERAJ**

**SRN: PES1UG21CS029**

**4TH SEM SECTION A**

**QUESTION:**

* Development of OS Fingerprinting Tool

**SOLUTION:**

* #!/usr/bin/python

from scapy.all import \*

import logging

logging.getLogger("scapy.runtime").setLevel(logging.ERROR)

# Define the OS fingerprinting function

def os\_fingerprint(ip):

# Define the fingerprints for various operating systems

fingerprints = [

{"os": "Windows", "ttl": range(113, 129)},

{"os": "Linux", "ttl": range(63, 65)},

{"os": "FreeBSD", "ttl": range(61, 63)},

{"os": "OpenBSD", "ttl": range(48, 50)},

{"os": "MacOS", "ttl": range(37, 39)},

{"os": "Cisco IOS", "ttl": range(253, 256)},

{"os": "HP-UX", "ttl": range(1, 2)},

{"os": "Solaris", "ttl": range(64, 69)}

]

# Send an ICMP echo request packet and capture the response

pkt = sr1(IP(dst=str(ip))/ICMP(), timeout=1, verbose=0)

#print (pkt)

# Check if the response is not None

if pkt is not None:

# Extract the ICMP sequence number from the response

if pkt.haslayer(IP):

ttl = int(pkt[IP].ttl)

print("Time to live:", ttl)

else:

print("No IP layer found in the packet")

return

# Loop through the fingerprints

for fp in fingerprints:

# Check if the Time to live matches the fingerprint

if ttl in fp["ttl"]:

# Print the operating system and return

print("The operating system of", ip, "is", fp["os"])

return

# If no fingerprints match, print an error message

print("Could not determine the operating system of", ip)

# Test the function with an example IP address

ip = input("Enter the IP address to fingerprint: ")

os\_fingerprint(ip)

**OUTPUT SCREENSHOTS:**



