**Assignment 03**

**Host Discovery Techniques**

**CS4061**

**Ethical Hacking Concepts and Practices**

**Submitted by:** Momenah Saif

**Roll number:** 21i-1909

**Date:** 16-03-2024

**Table of Contents**

[ **Introduction** 3](#_Toc161485101)

[ **Code Explanation** 3](#_Toc161485102)

[ **Summary** 6](#_Toc161485103)

[ **References** 6](#_Toc161485104)

# **Introduction**

In this assignment we were given to write a python code that shows options to run number of scans on a target machine. My code runs the following scans.   
1. ARP Ping Scan

2. ICMP Ping Scan

o ICMP Echo Ping

o ICMP Echo Ping Sweep

o ICMP Timestamp Ping

o ICP Address Mask Ping

3. UDP Ping Scan

o Port Scan

4. TCP Ping Scan

o TCP SYN Scan

o TCP Ack Scan

o TCP Null Scan

o TCP XMAS Scan

o TCP FIN Scan

5. IP Protocol Ping Scan

o ICMP, TCP, IP

I mostly used Scapy for scan but in some areas scapy wasn’t working write so I used other methods like nmap and ping

# **Code Explanation**

**Imports:**

* os: It is used for interacting with the operating system, such as getting the terminal size.
* scapy.all: It provides functionalities for crafting and sending network packets.
* subprocess: It allows to execute shell commands from your Python script.
* re: It is used for regular expressions, which are helpful for parsing the output of the ping command.
* nmap: It is a library for using the Nmap port scanner within the script.
* shutil: It provides utility functions for interacting with the file system. It was used in code to create a better output

**Functions:**

* generate\_ip\_range(start\_ip, end\_ip): This function takes a starting and ending IP address and generates a list of all IP addresses within that range, used in ICMP ping sweep.
* arp\_ping\_scan(target): This function performs an ARP ping scan on the target IP address. It creates an ARP request packet and sends it to the target. If a device responds, its IP and MAC address are displayed.
* icmp\_echo\_ping(target): This function performs an ICMP echo ping on the target IP address. It sends an ICMP echo request packet and waits for a response. If a device responds, its IP address is displayed.
* icmp\_echo\_ping\_sweep(target): This function uses the generate\_ip\_range function to iterate through a range of IP addresses and performs an ICMP echo ping on each one of the IPs in the range which are the next 10 IPs to target IP.This function is called in a loop in main to run it for every IP.
* icmp\_timestamp\_ping(target): This function performs an ICMP timestamp ping on the target IP address. It sends an ICMP timestamp request packet and analyses the response to extract the round-trip time.
* icmp\_address\_mask\_ping(target\_ip): This function attempts to perform an ICMP address mask ping on the target IP address. This type of scan is rarely used in modern networks and is not usually successful as most nowadays OS don’t respond to it.
* udp\_ping\_scan(target, start\_port=130, end\_port=150, retries=1): This function performs a UDP port scan on the target IP address, scanning a range of ports (default set to 130 to 150) and sending UDP packets to those ports. It analyzes the responses to identify open, closed, or filtered ports.
* tcp\_syn\_scan(target, start\_port=100, end\_port=150, timeout=2): This function performs a TCP SYN scan on the target IP address, sending TCP packets with the SYN flag set to a range of ports (default set to 100 to 150). It analyzes the responses to identify open and closed ports.
* tcp\_ack\_scan(target): This function performs a TCP ACK scan on the target IP address, sending TCP packets with the ACK flag set to a range of ports. It analyzes the responses to determine the state of the ports which can be open, closed, or filtered.
* tcp\_null\_scan(target): This function performs a TCP null scan on the target IP address, sending TCP packets with no flags set to a range of ports. It analyzes the responses to identify open or filtered ports.
* check\_os(target\_ip, system): This function uses Nmap to perform a OS detection on the target IP address and checks if the OS belongs to a specified system.
* tcp\_xmas\_scan(target): This function performs a TCP Xmas scan on the target IP address,first checks the OS using check\_os function. If the target OS is not Windows it sends packet to perform xmas to know the open and closed ports.
* tcp\_fin\_scan(target): This function performs a TCP FIN scan on the target IP address, which is usd to identifying open ports on UNIX systems. It checks the OS using check\_os before doing with the scan.
* icmp\_ping\_scan(target): This function sends an ICMP echo request packet (ping) to the target IP address and displays a message if a response is received.
* ip\_ping\_scan(target): This function sends a raw IP packet to the target IP address and displays a message if a response is received.
* tcp\_ping\_scan(target, port=80): This function sends a raw TCP packet to the target IP address and displays a message if a response is received.
* ip\_protocol\_ping\_scan(target): Its calls the above three IP protocol ping scans
* print\_centered(text, color\_code): This function is used for printing text with a centered alignment and a specific color code. It retrieves the terminal width and calculates padding to center the text visually.
* print\_menu(): This function creates the menu displayed to the user of the tool. It uses the print\_centered function for the title options.
* main(): This function first calls the print\_menu() function and then asks the user for target IP. Next it makes a IP range from the target IP. First it converts the entered target IP address into a list of integers, separating each octet (number between 0 and 255). Then it creates a copy of the start\_ip\_parts list to use for generating a range of IPs for scanning. After that it increases the last element (representing the final octet) by 10, creating an ending IP address 10 IPs above the target. Then it ensures the last octet doesn't exceed the valid IP range (0-255).Then the code converts the modified end\_ip\_parts list back into a string representation of the ending IP address. After that a while loop creates an infinite loop that continues until the user exits the program. The while calls the scans according to user input and menu displayed before. If user presses 0 the code exits.

# **Summary**

We were given to make a scanning tool using Python code. The code uses with libraries like Scapy and Nmap. It allows users to scan a target network and discover devices on it. The tool offers various functionalities through a menu-driven interface. Users can perform different types of scans, including ARP scans, ICMP pings (echo, timestamp, address mask), UDP port scans, TCP scans (SYN, ACK, null, XMAS, FIN) and IP scan (ICMP, IP, TCP). The tool also identifies the operating system running on the target device in some cases. By analyzing the responses to these scans, the script can help identify open, closed, or filtered ports and devices on the network.

# **References**

Used chatgpt multiple times to get functions that works properly for each of the scan.