TCP/UDP Port Scanning Using Python/Scapy

1.0 Objective

TCP and UDP port scanning is a method of determining what applications a host is running. You will develop 2 Python programs using the SCAPY module to scan TCP and UDP ports on the external router **ext-rtr** in VLAB.

For TCP you will scan all the ports from 0 to 100. You will collect the responses and sort them by their status of OPEN, CLOSED, FILTERED. i.e. OPEN:1, 2, 3, ...; CLOSED:8, 9, ...;FILTERED: 11, 12, ...

For UDP you will also scan all the ports from 0 to 100 and collect the responses by their status of CLOSED and OPEN.

You should account for dropped packets in your solution. In other words, if you send out a TCP/SYN packet and get no response you should send out another as the packet may have been dropped. For UDP, no response means either the port is OPEN or the packet was dropped. You should send out additional UDP packets to verify.

For an open UDP port lookup the service name associated with the port number and **send a well-formed UDP packet** for that service to the port to verify it is running the service and what that service is. Look up service names and transport protocol numbers from the number registry in iana.org. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

2.0 Lab Setup

Turn on the router **ext-rtr** and the **Kali** machine. You will be scanning the **ext-rtr** computer. You will code your program on the Kali machine using Python and SCAPY. The first line of your program should be:

*from SCAPY.all import **

4.0 What to Submit

Write a lab report describing your results. Each Python program should be submitted as a Python program i.e. ***<name>.py***, followed by a screen shot showing the output of the Python program.

- [40 pts] Your Python TCP scanning program.
- [40 pts] Your Python UDP scanning program.
- [20 pts] Your Python service name discovery.