Scanning the Network: Supplement

Exercises

1. Start a packet capture and run nslookup from a Windows host and from a Linux host. What is the TTL in the IPv4 header from the packet sent from the Windows system? From the Linux system? This is just one component of the method NMap uses to identify a remote operating system. See <http://nmap.org/book/osdetect-methods.html> for more details.
2. Run an NMap stealth scan against a target, specifying two TCP ports, one known open and one known closed. Capture the traffic between scanner and target. Identify the sequence of packets for the open and closed port.
3. Run an NMap connect scan against a target, specifying two TCP ports, one known open and one known closed. Capture the traffic between scanner and target. Identify the sequence of packets for the open and closed ports.
4. Read the file /usr/share/nmap/nmap-services (from a Kali system). Sort the result to determine the top 100 TCP ports. Run a default stealth scan against a target using the fast (-F) option. Verify that the TCP ports in the top 100 are scanned, but those outside are not. Repeat the process with a UDP scan (-sU).
5. Run a Metasploit scan using auxiliary/scanner/portscan/tcp specifying two TCP ports, one known open and one known closed. Capture the traffic between scanner and target. Identify the sequence of packets for the open and closed ports. Compare the results to an NMap stealth scan and an NMap connect scan.
6. Compare the Metasploit module auxiliary/scanner/portscan/syn with auxiliary/scanner/portscan/tcp. Which is more reliable? A network packet capture is helpful.
7. Compare an NMap ARP scan (-PR) to the Metasploit module auxiliary/scanner/discovery/arp\_sweep to the Kali tool arping.
8. Run the Metasploit module auxiliary/server/fakedns. How might it be useful?
9. Modify the Metasploit module auxiliary/scanner/dns/bind\_ver so that it reports the service to the database. Save the version of BIND in the info field.