# Port Scanning

**Objectives**

1. Introduce the techniques of port scanning.
2. Use port scanning audit tools such as nmap.

**Introduction:**

- All machines connected to a LAN or connected to Internet via a modem run many services that listen at certain ports.
- A service is a program that waits inside a loop for a request message from a client, and acts on the request.
- By port scanning, one discovers which ports are available (i.e., being listened to by a service). Essentially, a port scan consists of sending a message to each port, one at a time and examining the response received. If the port is in use, it can then be probed further for weakness.
- Port Scanning is one of the most popular among the reconnaissance techniques attackers use.

**Port Scanning Terms:**

**Port Numbers:** Both UDP and TCP use source and destination port numbers in their packets; the source and destination IP addresses are provided by the underlying IP.

Port numbers are an abstraction manufactured by the network layer of the operating system in accordance with the TCP/IP standards

These are 16-bit unsigned numbers. The port numbers are divided into three ranges:

1- Well Known Ports (from 0 through 1023)

2- Registered Ports (from 1024 through 49151)

3- Dynamic and/or Private Ports (from 49152 through 65535

**Sockets:**

• A socket is an abstraction, similar to a file descriptor, constructed by socket().

• A socket so constructed is bound to an IP address and port number via the bind() call.

• A server program then waits for a connection via the listen(), and accept()s a connection.

A socket is said to be active after the server has accepted a connection. It is connected to a remote active socket via an open data connection. Closing the connection destroys the active sockets at both endpoints. A passive socket is not connected, but rather awaits an incoming connection in the listen(), which will spawn a new active socket. A socket is not a port, though there is a close relationship between them. Each port can have a single passive socket, awaiting incoming connections, and multiple active sockets, each corresponding to an open connection on the port. Servers use bind(), listen(), and accept(). A client uses connect(). The read(), write() are used by both clients and servers. The process of connection is show in figures 1 ,2 and 3.

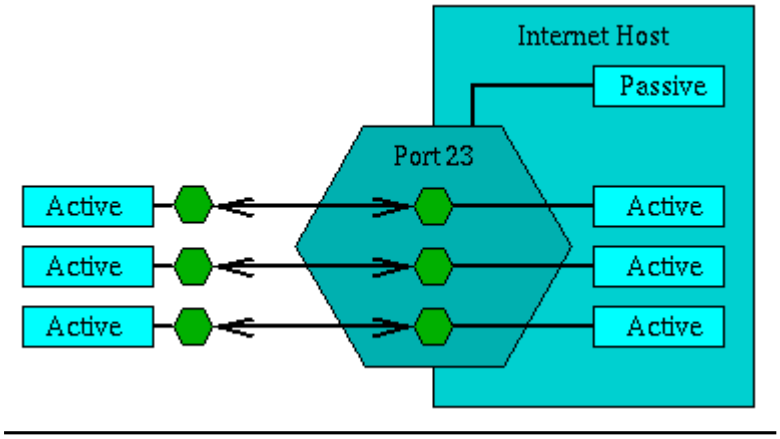


**Figure 1**

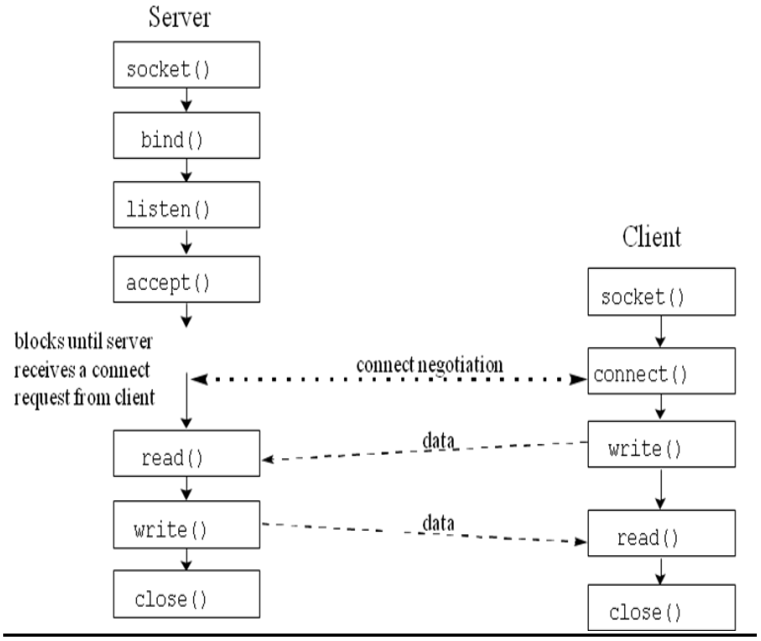


**Figure 2** Socket calls for connection-oriented communication

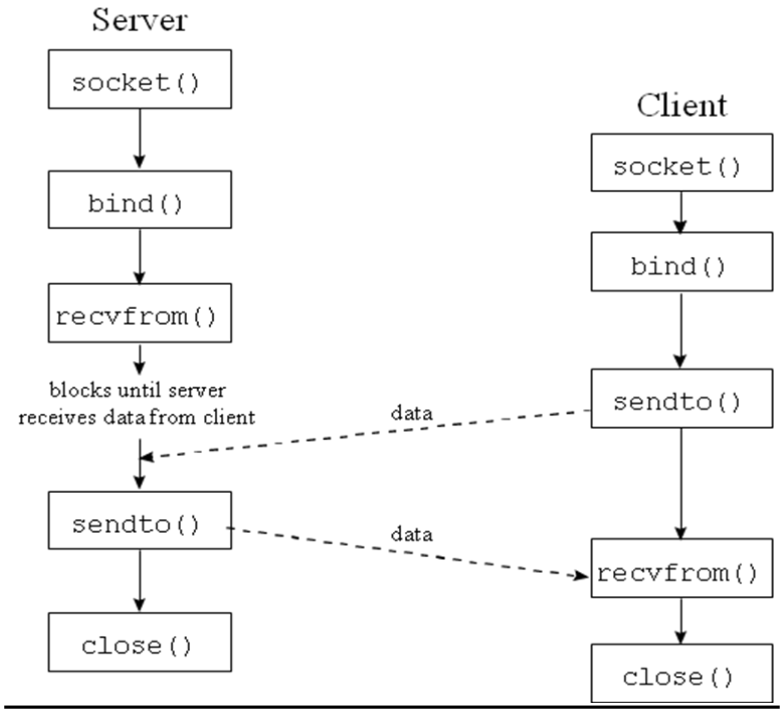


**Figure 3** Socket calls for connectionless communication

**Open Port:** A service process is listening at the port. A port is opened by the OS at the request of a specific process. The OS receives packets arriving at this port and gives the messages to the service process. If the OS receives a SYN at an open port, this is the first packet of the three way handshake.

**Closed Port:** No process is listening at the port. If the OS receives a SYN at a closed port, an RST is sent.

**Filtered Port:** A packet filter is listening at the port.

**UDP scan:** Finds open UDP ports. Note that TCP and UDP both have the same port numbers, even though the OS distinguishes them as completely separate; see the file named /etc/services. The source port of UDP is an optional field. When meaningful, it indicates the port of the sending process. If it is not meaningful, a value of zero is used. UDP responds in a different manner from a TCP scan. In order to find UDP ports, the attacker generally sends empty UDP datagrams at the port. If the port is listening, the service process will send back an error message or ignore the incoming datagram. If the port is closed, then the operating system sends back an "ICMP Port Unreachable" message.

**Fingerprinting an OS**

Fingerprinting is the technique of interpreting the responses of a system in order to figure out what it is. To make this more effective, unexpected but well-chosen combinations of data are sent to the system in order to trigger unique-enough responses. This is because while most systems respond alike with correct data, they rarely respond the same way when sent unusual data.

**Port Scanning Tools**

1. SAINT
2. nmap (zenmap) , which we discuss below.
3. nessus.

**Nmap:**

The nmap port scanner (www.nmap.org) is widely known. According to its author Foydor, nmap is a utility for port scanning large networks, although it works fine for single hosts.

Sometimes you need speed, other times you may need stealth. In some cases, bypassing firewalls may be required. Not to mention the fact that you may want to scan different protocols (UDP, TCP, ICMP, etc.). You just can't do all this with one scanning mode. And you don't want to have 10 different scanners around, all with different interfaces and capabilities. All these scanning technique you will found in nmap.

Nmap ("Network Mapper") is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.

**Uses of Nmap :**

While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime

**Nmap output :**

The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. Key among that information is the "interesting ports table".

•That table lists the port number and protocol, service name, and state. The state is either open, filtered, closed, or unfiltered. Open means that an application on the target machine is listening for connections/packets on that port. Filtered means that a firewall, filter, or other network obstacle is blocking the port so that Nmap cannot tell whether it is open or closed. Closed ports have no application listening on them, though they could open up at any time. Ports are classified as unfiltered when they are responsive to Nmap's probes, but Nmap cannot determine whether they are open or closed. Nmap reports the state combinations open|filtered and closed|filtered when it cannot determine which of the two states describe a port.

•The port table may also include software version details when version detection has been requested.

•In addition to the interesting ports table, Nmap can provide further information on targets, including reverse DNS names, operating system guesses, device types, and MAC addresses.