Port Scans

Jalal Kawash
American University of Sharjah, UAE

INTRODUCTION

The hardest task for a hacker is to get a foothold into a computer network system. If the hacker manages to get inside, the rest of the network is easily conquered. This is often referred to as the 'eggshell principle'. It is hard to pierce in, but once inside, the whole network can be available for the hacker's grab.

Trying to map a blueprint of the target network is the first challenge to the hacker. It is important for hackers to find out as much information as possible concerning the target network. Such information may include host names and addresses, applications and operating systems running on these hosts (versions and patch states), and application organization (servers and their roles).

A famous hackers' operation is 'zone transfer'. The zone contains all the registered host names in a network. Zone transfer is a request to get a copy of the zone. The underlying Routing Information Protocol can be discovered and manipulated to redirect traffic and obtain unencrypted information. A decent firewall can disable zone transfers, but administrators do not always take necessary precautions. Yet if they do, piercing into the network is still possible utilizing *port scanning.*

In a computer network system, application processes communicate with each other by sending and receiving messages. Transport addresses, called *end-points,* are specified and associated with each application process. In the Internet, each end-point is a pair of values. The first value is a unique machine address, called the Internet Protocol (*IP*) address. The second value is a local *port* number, which uniquely identifies the address of a particular process per IP address. An *open* port is one that has a process listening for connections; otherwise, it is said to be *closed.*

Port scanning is the process of determining which ports are open. Open ports may expose the host machine to external attacks because they can be further examined by attackers to discover and exploit network service vulnerability. Hence, port scanning allows a hacker to determine which applications are exposed on a target host. Such exposed applications are typically essential to the operation of the system. For instance, HTTP Web servers (port 80) and FTP (port 21) are typical exposed applications.

Open ports are associated with services. So, if the hacker can find out information about the nature of the service, its version, and patching state, he or she can work on exploiting it. For instance, if the hacker performs operating system fingerprinting, to find out the underlying operating system version, he or she can figure out if there are any known security problems with the version at hand, especially if it is not properly patched. Exploiting these security holes, and with the help of other tools, a hacker can install a backdoor into a system through open ports. This can allow the hacker to control the system remotely.

Since port scanning is important to hackers, it is also equally important to network administrators, but with contradictory objectives. Since open ports may expose a machine to potential external attacks, it is important that administrators identify them to monitor or avoid any possible hacking attempts. This article exposes the subject of port scanning, outlining some of the technical details on how such scans work.

BACKGROUND

Shaping Factors

There are three important factors that shape the implementation of a port scanner. These are the underlying communication protocol, filtering and detection, and time and bandwidth (Schiffman, 2003; De Vivo, Carrasco, Isern, & De Vivo, 1999). Call the attacking host (where the port scanner is running) the *scanning host* and the attacked host the *scanned host.*

- **Protocol:** Port scanners obtain their clues about ports by analyzing the underlying protocol's behavior. Hence, a scanning method is typically

designed for some specified protocol. On the Internet, the most famous transport protocols are Transmission Control Protocol (TCP) and the User Data Protocol (UDP). In this article, we will limit the discussion to TCP scans, since UDP plays a lesser role in port scanning. The basic behavior of TCP is explained later in this section.

- **Detection:** Network administrators rely on an Intrusion Detection System (IDS) in order to monitor the network and detect any possible attacks. With an ever increasing number and effectiveness of attacks, IDSs have become indispensable for most organizations. Administrators cannot run detectable port scans while an IDS is running, since the latter can disable the scanner. Shutting down the IDS is obviously a bad idea. Hence, it is important to provide administrators with stealth port scanners that do not only hide the scanning behavior, but also conceal the scanning host IP address.
- **Time and Bandwidth:** Attackers have the luxury of spending an enormous amount of time performing port scanning, but administrators do not. Controlled port scanning incurs substantial overhead over the network's bandwidth and consequently its throughput. That is why administrators' controlled scans rarely span more than one day. Therefore in addition to providing administrators with stealth port scanners, it is also crucial to provide them with powerful scanners that can perform a comprehensive scan in a limited amount of time.

TCP Behavior

TCP is the most common target for port scanning. TCP is a connection-oriented protocol that any two processes need to establish a connection with two end-points in order to communicate with each other. To establish a TCP connection, a pair of processes follow the *three-way handshake* (Transmission Control Protocol RFC, 2006). This handshake consists of sending and receiving messages, called *segments,* whose purpose is to initialize the connection and its data streams. A TCP segment consists of a header and an optional body. The header includes six control bits, called flag bits, or simply *flags.* The flags that are set in a TCP segment header convey valuable information to the scanning host. Such information can reveal for instance whether a port is closed or open (Transmission Control Protocol RFC, 2006).There are six flag bits (Stevens, 1994, 1995):

- **SYN:** The SYN bit is set when a sender or a receiver is establishing a connection. The sender sets this bit in order to request a connection, and the receiver sets it in order to accept a connection.
- **FIN:** The FIN bit is set to signal the act of finishing sending data by the sender. That is, it releases the connection.
- **RST:** This bit is used to reset the connection. Typically, it signals a problem. The RST bit is set when confusion takes place due to a host crash, for example. It is also used to refuse a connection. This bit is typically set in response to a SYN or FIN segment arriving at a closed port.
- **URG:** This bit states that the urgent pointer is being used. The urgent pointer points to urgent information contained in the TPC segment and allows the receiver to jump directly to this information.
- **ACK:** The ACK bit, when set, states that the TCP segment is an acknowledgment.
- **PSH:** If this bit is set, the segment is asking the receiver to rush (push) delivery to the application process.

A TCP segment that has one of these six bits set, say bit *x*, is called an *x* segment. The three-way handshake starts with the sender sending a SYN segment to the receiver; the receiver acknowledges this SYN segment by a sending back an ACK+SYN segment (both the ACK and SYN bits are set). The sender finally acknowledges the receiver's SYN segment with an ACK segment and the connection is initialized. Figure 1 illustrates a successful three-way handshake.

The arrival of a TCP SYN segment at an open port starts the three-way handshake. However, if the port is closed, an RST segment is sent back to the sender. RST segments can be also sent as a result of the arrival of an ACK segment at an open port. A FIN segment can also detect if a port is open or closed since a closed port results in sending an RST reply, but for an open port the FIN segment is simply dropped. URG and PSH flags can also be used to detect open ports.