Scan Your Way to Port Security

Did you leave your TCP/IP door unlocked?

by Vince LeVeque

There's a fine line between tools that can break or attack computer systems and those that are used to manage systems. Just as a knife that is sharp enough to cut steak is also sharp enough to kill, any tool that can manage or monitor a network can also be used to break into a network.System and network managers need to know about these attack tools to defend against their illicit use. It's also important to know which attack tools are solid and reliable enough to provide good information for managing your systems. An instrument designed to cause harm can also be used for productive purposes—(though using a switchblade to carve your steak may not be a good idea!).

Reconnaissance and Hacker Tools

One of a hacker's first steps in attacking a system is reconnaissance— gathering information about your network to find possible security holes. This is like a burglar casing a building to see where the doors and windows are located, which are unlocked, and which are hidden so their potential trespass can be concealed.

Hackers have their own tools for performing reconnaissance. These tools attempt to find unsecured systems that can be externally accessed. One tool, a wardialer, goes through an entire set of telephone numbers, identifying the ones with a modem tone and inventorying any login banners for later analysis. Wardialers first gained notoriety in the early 1980s movie War Games in which a youthful computer enthusiast gained access to a sensitive NORAD system via telephone link. A wardialer is a program that calls a given list or range of phone numbers and records those that answer with handshake tones (and so might be entry points to computer or telecommunications systems). We will not dwell on wardialers in any depth here, but suffice it to say that unsecured modems are a very big security exposure. Users with unauthorized remote control software like PC Anywhere place their entire network at risk for the convenience of dialing the office computer from home. If a network firewall is like a security guard at an office's front lobby, then PC Anywhere is like a door propped open at a back loading dock for someone's convenience.

A network scanner is another tool that can detect and analyze hosts on TCP/IP networks, such as those connected to the Internet. Network scanners can note the presence of hosts; the services running on them, and even make fairly accurate judgements about the type of host and the type and version of software on it.

TCP/IP Basics

To understand how a scanner works, we first need to review some basic TCP/IP.

TCP/IP is a layered protocol. Functions are defined modularly with higher level functions using services provided by lower level functions. The higher level functions only need to know the interface with the lower level function and not their internal functioning.

The top layer is the application. An application may be email, Web services, telnet, or even APPC applications such as Client Access/400. The protocols supporting the application provide system-to-system connectivity, so called session layer protocols. Applications communicate with the session layer via an abstraction called a port. A port is represented by a 16 bit number. Different applications use different ports, and certain applications have generally assigned ports. For example, telnet uses port 23, HTTP typically uses port 80, and DDM (when run over

```
//*****************************************************************
//
//
// To Compile:
//
// From a PC with a java compiler:
//
//    javac PortScanner.java
//
//
//  From the AS/400:
//
//   CRTJVAPGM CLSF('/YourDir/PortScanner.java')
//
//
//*****************************************************************
//
//
// This application will scan all TCP ports on the host system
// specified as a parameter to this class. It will print a list
// of the "listening" ports on the standard output line.
//
//
//  To Use:  java PortScanner YourSystemHostName(or IP address)
//
//*****************************************************************
import java.io.*;
import java.net.*;

public class PortScanner
{
  private static Socket sock = null;

  public static void main(String[] args)
  {
    if(args.length == 0)
    {
      System.out.println("Usage:  java PortScanner [hostname|ip address]");
      System.exit(0);
    }
    int count = 0;
    for(int port = 1; port <= 1024; port ++)
    {
      if(isListening(args[0], port))
      {
        System.out.println("Found a listener on host " +
sock.getInetAddress().getHostName() +
        " at " + sock.getInetAddress().getHostAddress() + ", port " + port);
        count ++;
      }
    }
    System.out.println("Found " + count + " ports listening at " + args[0]);
    System.exit(0);
  }

  private static boolean isListening(String hostorIP, int port)
  {
    boolean b = false;
    try
    {
      sock = new Socket(hostorIP, port);
      sock.close();
      b = true;
    }

    catch(UnknownHostException uhe)
    {
      System.out.println("Cannot connect to " + hostorIP);
      System.exit(0);
    }
    catch(Exception e)
    {
      //do nothing
    }
    return b;
  }
}
```

Figure 1: **This simple port scanning utility will identify open ports on the AS/400.**

TCP/IP) uses a number of ports, including 44, 447, and 448. Both the clients and the servers must allocate ports to a communication session. The client ports are variable,

Ports below 1024 are considered special and are called *well known ports*. These parts have specifically assigned services associated with them. On UNIX systems, services on well known ports can only be run the root or superuser. Ports above 1024 are called *registered ports*. Services are assigned port numbers by a designated Internet agency, so the service may stake some global claim to the port, avoiding conflicts with other, different services. Often, though, ports are simply claimed by the application vendor. The official list of ports is published by the Internet Engineering Task Force (IETF) as one of their Request for Comments (RFC) documents, specifically as RFC 1700. This document may be found on the IETF Web site at *www.ietf.org*.

TCP/IP actually has two different session-layer protocols: User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). While both protocols provide ports to the application and both establish communication between endpoints, their underlying philosophy is very different . TCP provides reliable connection-oriented services, while UDP is a connectionless datagram service.

The session layer protocols in turn rely on networking protocols, which enable the network to find the route between the two endpoint systems. In TCP/IP, this is the Internet Protocol (IP). IP defines the source and destination address. Between the source and destination, devices called routers look at the destination address and decide which direction the packet should be sent. Both UDP and TCP rely on IP to carry them across networks, finding the route from source to destination.

One of a hacker's first steps in attacking a system is reconnaissance.

Finally, IP rests on top of various data link and physical protocols, including Ethernet, Token Ring, Frame Relay, and serial communication links. These layers are almost entirely invisible to IP. This fact, coupled with IP's extreme versatility in supporting almost every known data link and physical layer protocol, has helped make TCP/IP the de facto standard it is today.

Another protocol in the TCP/IP suite is harder to categorize. This is the Internet Control Message Protocol (ICMP). ICMP is often considered part of the IP layer, even though the ICMP information is contained within an IP packet. ICMP is used to transmit network control and status information between systems. When users receive a message that says "host unreachable" or "network unreachable", they are seeing messages generated via ICMP. ICMP is also behind the ever useful ping command.

This background helps us explain how a scan works. First, a scanner attempts to find all the systems on a network. The scanning software is given a range of IP addresses and for each address in that range, queries each one to find if it belongs to an active server. The most straightforward way of doing this is though ping. Other methods may use UDP or TCP packets.

When active servers are found, the scanner may then query each possible port on the server, to determine which services are provided. TCP scanning can be as simple as executing a TCP connect command, attempting to set up a connection with a whole series of ports, and then seeing if the connection request is accepted or refused. Writing a simple TCP scan program is not difficult for someone with basic familiarity with sockets

programming. An example Java program written by *Midrange Computing* contributor Jeff Markham is shown in Figure 1.

UDP is more difficult to scan because it does not generate a connection accepted/denied message. UDP scanning involves sending out UDP request packets and then waiting until a response is received. Rather than explicitly denying a request, UDP will simply time out with no response. This makes UDP scanning very slow. Compounding the slowness of the scan is a feature of UDP described in RFC1812. This feature slows down response to a series of UDP requests by progressively lengthening the time it takes to respond. To scan a full range of UDP ports under these conditions can try one's patience!

After identifying that a server exists at a specific IP address, and after identifying possibly vulnerable open ports, the next step is to gather information on the server itself and on the services. The two main methods for this are banner text gathering and fingerprinting using responses involving nonstandard packet formats.

Banner text gathering means simply grabbing the service's standard "log on" banner and looking for key words which give away the nature of the system and its services. For example, if you log in to an AS/400 suing FTP, chances are the initial message will say something like "QTCP at system x". The "QTCP" part of this gives away the fact that it is an AS/400 you are talking to. Knowing this, a hacker could then try accounts like QSECOFR, QSRV, QPGMR, etc., with their default passwords figuring that at least one of these will let them in.

Nonstandard packet fingerprinting involves portions of packets not explicitly defined in standards, responses to nonstandard inquiry packets, or cases where implementation is explicitly system-specific. Since these are not defined in the RFCs, there are subtle differences between different vendor's systems and between different versions of vendor's products. However, once identified, an attacker can focus on the vulnerabilities specific to the victim, whether it be default passwords, software flaws, or open file shares.

Useful Information

It should be clear how scanning tools can be used to break into systems. But these tools also have a benevolent side and can provide useful information to network administrators. Most obviously, a thorough port scan can show which systems are running unauthorized services, and which ones present security vulnerabilities. This sort of proactive security audit is important.

In addition to mimicking the behavior of an attacker, a scanner can assist a security audit in other ways. Many systems now come with built-in Web servers and FTP servers. If not included in the package, shareware versions of these servers can easily be downloaded and even installed on desktop PCs. Systems controlled by a responsible Information Systems organization hopefully have only authorized servers installed, however many systems are operated by end user departments or are otherwise not well controlled. It is often tempting to set up unauthorized servers on these systems, and in some cases users may believe they have a good business reason for doing so. Even where Information Systems staff are responsible for the system, unneeded services may be turned on. The best business reasons will not remove the inherent security flaws of unauthorized servers. Unauthorized server programs may be installed on desktop systems by users attempting to create some sort of useful feature. Maybe they want to start a departmental Web page, or they have just found some nifty network meeting software. These unauthorized programs can make network management more difficult by unexpectedly adding traffic and, in some cases, affecting network stability. Even with well controlled central systems such as an AS/400, servers may be started without full awareness of the risks. Scanners are valuable in ferreting out unauthorized servers. Once identified, these unauthorized servers can either be shut down or moved to a standard supported enterprise platform.

The best business reasons will not remove the inherent security flaws of unauthorized servers.

Denial of Service

Recently a number of Web sites were put offline due to so-called distributed denial of service attacks. These attacks involved compromising many systems with high speed Internet connections, and then setting them up so they would respond to a single remote command to initiate a coordinated attack on a specific target. This is just one example of an attacker loading stealth software to remotely manage and monitor the victim's systems. Backdoor software, including many so-called Trojan horse programs, has become a common tool in attackers' kits. Network scanning can also detect backdoor software, by noting the characteristic ports used by the attacker and then using those open ports to remotely control the victim hosts. Ports often used by malicious software includes 12345, 373343, 6711 and 6776 (see http://www.commodon.com/threat for more information). These ports are the ones defined in the default configurations of the common attack software. Port scans should be combined with the most current antivirus software to provide the best protection.

Lastly, a scanner can provide information regarding what is actually on your network versus what has been documented. While I'd love to believe that all users have the time to keep their network diagrams up to date, there are those rare occasions when they get behind. A scan is an excellent way to reveal new systems, disclose systems that are no longer online, and show which servers are providing which services.

Be Careful!

If users decide to use scanning as a proactive tool for network management, they should be careful. Use only reputable tools and only use tools for which source code is provided. Attack tool authors are not always the most careful or ethical individuals, and users do not want to run a tool that leaves their security worse than it was before. If possible, review the source code to ensure that no hidden functions and no obvious errors are in the tools. Some less-than-reputable tool authors have been caught including features such as having the software "secretly" e-mail the list of vulnerable servers to the tool author themselves, for example. The very nature of scanning is itself somewhat risky. Machines with less than robust TCP/IP software may crash or otherwise behave oddly when faced with a scan. Using host fingerprinting can be particularly dangerous. Fingerprinting attempts to identify the target system through subtle differences in how different types of systems respond to different communication inquiries. Some of these inquiries involve sending packets which do not follow TCP/IP standards. These packets may have flags and various options set in ways not supported by standards, for example. When faced with non-standard packets, systems often respond in vendor-specific ways. Since there are no standards, each TCP/IP stack is free to implement their own response. Unfortunately, sending nonstandard packets to a machine may cause undesirable results, such as an unwanted reboot. You certainly don't want your port scan causing an entire department of Windows machines to give the "Blue Screen of Death" all at once!

Software that specializes in scan detection is available. These are especially useful for

personal computers connected directly to high speed Internet lines , such as popular cable modem and DSL connections used by many home users. Users' machines with one of these connections are being scanned more often than they'd like to think. For PCs running Microsoft Windows, consider installing Network Ice's BlackIce Defender software, which can be downloaded from http://www.networkice.com for $40. For Linux and other personal UNIX systems, PortSentry by Psionics is worth a look. These products perform a number of functions, chief of which is detecting a pattern of repeated inquiries against different ports on your machine, all coming from the same source IP address.

Use It or Lose It

Network scanning can be useful to network administrators. These tools are important for security assessments and information gathering, though they should be used with caution. Scanning tools can also be very dangerous when used to attack a network, by finding systems with vulnerable services. A scan from the global Internet is almost always trouble, and unfortunately it occurs all too often. Firewalls can protect against scans by locking out unauthorized ports, by obscuring the nature of internal systems, and by preventing many stealth scans. A reputable firewall product always provides scan detection features. Scan detection should be enabled and should generate an immediate alert to the firewall administrator. On the AS/400, IP Filtering and Network Address Translation (NAT) can be used to help protect systems against scanning. For more information on IP Filtering and NAT, check out *Secure your AS/400 with IP Filtering*, MC, August 2000 and *Not an Insect but a Powerful TCP/IP Tool*, MC, September 2000 . In the right hands, port scanning can be a great tool to help secure an AS/400, NT, UNIX or even Linux system. But in the wrong hands, scanning can expose a system to danger. Make sure you take advantage of this ability, before some hacker does.

Vince LeVeque is a senior security engineer for Science Applications International Corp. (SAIC). He can be reached at *vleveque@earthlink.net.*

REFERENCES AND RELATED MATERIALS

- *Hacking Exposed: Network Security Secrets and Solutions.* Stuart McClure, Joel Scambray, and George Kurtz. MacGraw Hill, 1999
- Insecure.org Web site: *www.insecure.org*