Port Scanning Tools For Windows And Linux

If you look after servers, you need to understand port scanning. If you don't, rest assured that the hackers who scan your systems will certainly do so.

By Pan Pantziarka
Technical Writer

Port scanning is the network equivalent of a burglar casing the premises before attempting entry. It is used by hackers to sniff out points of entry into a network, to establish vulnerabilities before proceeding any further. And, just as with the burglary analogy, making sure that your premises are secure can often be enough to deter the opportunist. The aim of this article is to explain what a port scan is, why it is performed, and to examine a number of tools used in scanning. In this sense it is attempting to arm network support personnel with the same weapons likely to be used by hackers and other malicious users.

Some of the tools described run under Windows 9x or NT/2000. All are easily available as freeware or shareware, and thus well within the technical capability and budget of small companies or departments of larger ones. In every case a source for the tool will be provided so that you can try it for yourself. However, think carefully before trying a port scan on a live system that you do not own or have official responsibility for. You may be breaking the law, or breaching your employment contract, if you do not have permission to do so. Also, if the system has some form of intrusion detection software installed, initiating a port scan might trigger the system into automatically shutting down or some other equally unwanted pre-programmed action.

In addition to Windows, many of the better known scanning tools are available for the Linux platform. Download sites for these packages will also be included, even where the software may be commonly available as part of a wider Linux distribution such as Red Hat, SuSe, Mandrake or Debian.

What Is A Port?

In the most general sense, a computer port is any device which can be used to provide data I/O. The most common examples are the hardware serial and parallel ports that most computers still come equipped with. With these devices the flow of data can be bi-directional and takes place at a very low level in the operating system. Data to and from a serial port, for example, is handled by device drivers rather than an end-user program such as Word or Excel.

The same idea of low-level bi-directional data-flow applies to the ports we are discussing in this article. The TCP/IP and UDP ports, which is what port scanning tools are looking for, can be viewed as providing a simple bytestream connection between two machines operating across a network (or the Internet). Application programs such as Internet Explorer can use these ports to communicate without having to be concerned with the various network layers included in the TCP/IP stack.

TCP/IP and UDP ports are described by a 16-bit integer, which makes for 65535 possible ports (or 65535 entry points into your machine, if you want to be really paranoid). These port numbers are not assigned at random. Ports from 0 to 1023 are often called the "well-known port numbers", and are assigned by the Internet Corporation for Assigned Names and Numbers (ICANN). These ports include port 80, used by the http protocol for delivering Web pages, port 21 for ftp, port 110 for POP3 email and so on. Because these ports have standard numbers a Web browser, for example, knows that it simply has to log into port 80 on a machine in order to request pages from any http server that is running.

Ports 1024 to 49151 are called the registered ports, and these too are assigned by

ICANN. These are used by particular application programs, such as port 1433 for Microsoft SQL Server. Finally, the ports from 49152 through to 65535 are called dynamic port numbers or private ports, and can be used by any program to communicate with any other program. A full list of assigned ports is available from **http://www.iana.org/assignments/port-numbers**. [*This file is also on your CD - Ed.]*

How Does Port Scanning Work?

There are a number of different types of port scan, but all of them involve sending a message to the port on the computer being scanned (which is identified by IP address) and then waiting for a response. The type of message sent and the information returned by the machine are what define the different type of scans. The reasons for varying the messages sent are driven partly by the need to evade detection and partly by watching for a particular response. The simplest example of a scan is a TCP Connection scan, in which a full connection to the target computer is initiated. This involves a three-way handshake (SYN -> SYN / ACK -> ACK), and is easily detected on the target machine (which immediately makes it unpopular with would-be hackers). A TCP SYN scan, on the other hand, does not make a complete connection to the target - the scanning machine refuses the connection at the last stage, by which point it has received information from the target and yet may not have been logged because a connection was not fully established.

The responses to a scan can tell the scanner which ports are open, but can also give a clue as to which operating system the target machine is running. For example a TCP RPC scan only works for Unix and can return information about which remote procedure call (RPC) ports are open and which programs are using the ports.

Detecting Port Scans

Before launching into the survey of different scanning tools, it is perhaps appropriate to look at how it is possible to detect whether your machines are being scanned. One of the simplest methods of detecting scans is to install a firewall package. One obvious candidate is ZoneAlarm, a well-known personal firewall, which flags up a dialog box whenever a machine is online and subject to a port scan.

Alternatively there are a number of dedicated intrusion detection systems (IDS) which will flag port scans as well as attempts at network intrusion (hacking). Many of these tools are for Unix and Linux, but one of the most well-known of these - delightfully called “snort” - is now available for Windows 2000. Snort is available for free under the GNU Public Licence. The Web page for snort - (covering versions for all operating systems) - is **http://www.snort.org**.

Hackers go to all sorts of lengths to cover their tracks when port-scanning a remote computer. For example, it’s not difficult for an IDS to notice when all its ports are scanned sequentially in the space of a few minutes from the same address. But if the scan is done over a period of weeks, from half a dozen different addresses, detection is made much harder.

IPEye

IPEye is a command-line driven port scanner for Windows 2000 by Arne Vidstrom. Weighing in at a mere 32 KB download, this is a slimmed down but powerful tool that bears many similarities to the kind of command-shell utilities familiar to Unix and Linux systems. The basic usage for IPEye is:

```
ipEye <target IP> <scantype> -p <from port> <to port> [op-
tional parameters]
```

The scantype parameter can take values of:

-syn = SYN scan
-fin = FIN scan
-null = Null scan
-xmas = Xmas scan

Of these scan types, only the SYN SCAN is valid when scanning a Windows system.

IPEye will scan the requested ports, given a valid IP address, and return a list of

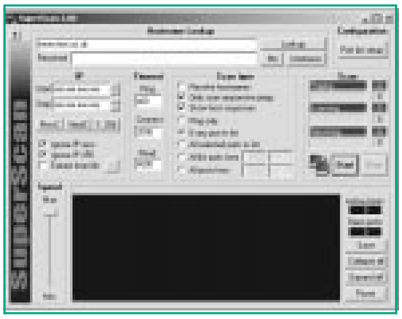


Figure 1.

ports which are open, closed or rejected. The optional parameters include a delay between scanned ports, a source port and a source IP address for the scan. The latter can be used as means of hiding the source of scan.

The latest version of IPEye can be downloaded from Arne Vidstrom's site at **ntsecurity.nu/toolbox/ipeye/**, which also includes a useful FAQ. The site is also home to a large number of other security and networking utilities which are certainly worth deeper investigation.

While IPEye is very useful, it does suffer from a number of drawbacks (in addition to its command-line nature). Firstly it is specific to Windows 2000, and so it is not suitable for NT or Windows 9x. Secondly it can only scan one IP address at a time. Finally, you need to have the IP address of the machine you are scanning - host names are not accepted.

SuperScan

SuperScan 3.0 is a fully graphical port scanner for Windows (see Figure 1), written by Rob Keir. SuperScan is able to scan ports from a range of IP addresses, ping remote machines to check for signs of life, scan for different ports and is also able to perform host lookup, by-passing the need to manually look-up the IP address of a system you need to scan.

The list of ports that it scans is fully configurable, and the default list is both comprehensive and informative. Three lists are supplied: the default list is supplemented by a list of common trojans and application ports from the excellent book "Hacking Exposed" and the third is a list of ports used by trojans. Ports can be manually selected, and additional parameters can be entered for each of them. By using host look-up the program is primed to port scan the IP address that the host name resolves to. A tree display is used to provide feedback on all the hosts / ports that it finds open, which is a useful mechanism for navigating through large numbers of IP addresses and ports.

Despite the usefulness of SuperScan, it too is a rather slim download of around 250 KB. The on-line help file provided with the program provides much additional useful information. SuperScan works on all versions of Windows which support the TCP protocol. In practice this will be all versions of Windows 9x, NT and Windows 2000. Early releases of Windows 95 did not include this, but if a machine is able to connect to the internet than you can be assured that it can run SuperScan.

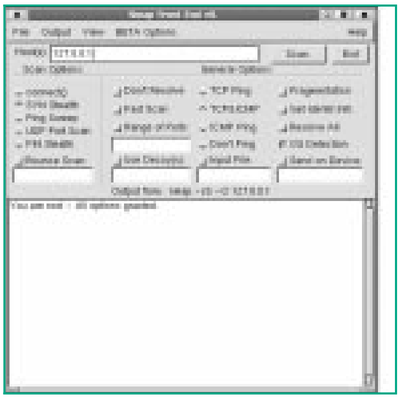
The latest version of SuperScan, and a number of other network and security tools, can be downloaded from **www.foundstone.com**.

Atelier Web Security Port Scanner

AWSPS is a much more complex beast than either of the other two Windows-based scanners. It features an idiosyncratic user interface which somewhat obscures the usefulness of the tools that it includes. In addition to providing TCP scanning functionality, it also features UDP port scanning, local network enumeration and a high-level of detail on the local network set-up.

For a machine on a local area network, AWSPS can provide extremely useful information about other networked machines, users and so on. Just as importantly, it provides a first-rate listing of the port setup on the local machine, detailing which ports are open, what their status is etc. Additionally it can provide traffic details for TCP and UDP traffic, as well as for control packets (ICMP), including ping. It is this extra set of functionality devoted to the local machine and its networked environment that differentiates AWSPS from the likes of IPEye and SuperScan. Where the other tools are more outward looking, AWSPS turns its gaze inwards. The user interface is not as clear and as simple as SuperScan's, but the level of detail it can provide makes this an important utility to include in any network security toolbox.

AWSPS is a shareware product, and the latest version can be downloaded from the Atelier Web site at **www.atelierweb.com/pscan**.



Host(s): 127.0.0.1

- ○ connect()
- ● SYN Stealth
- ○ Ping Sweep
- ○ UDP Port Scan
- ○ FIN Stealth
- □ Bounce Scan
- □ Don't Resolve
- □ Fast Scan
- □ Range of Ports
- □ Use Decoy(s)
- ○ TCP Ping
- ● TCP&ICMP
- ○ ICMP Ping
- ○ Don't Ping
- □ Input File
- □ Fragmentation
- □ Get Identd Info
- □ Resolve All
- ☑ OS Detection
- □ Send on Device

Output from: nmap -sS -O 127.0.0.1

You are root - all options granted.

Figure 2.

Other Windows Port Scanners

The three Windows port scanners listed above are just a sample of the numerous tools and utilities which exist for this platform. Other well-regarded tools include: NetScan Tools Pro 2000, NTOScanner, WinScan and AW Security Scanner. Some of these tools are commercial products, though evaluation copies are often available for download.

Nmap

Created by someone who calls himself Fyodor, nmap (short for Network Mapper) is rightly considered to be the scanning tool by which all others are measured. Currently available for Linux, Solaris and a number of other BSD-based flavours of Unix, it can be downloaded from Fyodor's extremely informative site at **www.insecure.org/nmap**.

Although it is primarily a text-based utility, run from a command shell, recently a graphical front-end has also been produced, though currently this is still in beta (see Figure 2). To get a feel for the power in nmap, it is still best to run it from a terminal session from Gnome or KDE or other X Windows system. For best results it is recommended that a user has root privileges. Typing nmap -h at the shell prompt produces a listing of the numerous options and parameters that the program can take.

The man pages for nmap are highly recommended. Not only do they detail the various options and parameters, but they contain useful insights in the vulnerabilities and flaws in different operating systems. The hacker roots of nmap are clear to see, and it includes the ability to spoof IP addresses, fragment packets, use ftp relay hosts and more. It is a tool that is in daily use by malicious users, and it makes sense for network security personnel to be just as familiar with it.

Other Linux/Unix Tools

Whilst it is the most well-known of the Linux / Unix port scanners, nmap is not the only such tool. There are many tool kits and utilities available for these platforms, nearly all of them available as open source. Most of these utilities are command prompt driven, and require a good degree of technical expertise to yield the best results. Worthy of mention are two quite venerable utilities which still provide useful capabilities: strobe and netcat. Both of these are available for download from numerous mirror sites, both as source code and as binary files (in both RPM and DEB formats).

Summary

Port scanners are vital pieces of any network security toolkit, and the range of such utilities is quite extensive for all major platforms. In all cases good research and an understanding of the underlying technology is essential to get the most use from the software. Hackers are possibly already pointing such programs at your network, so it makes sense that you do the same before they do. But it is also important to note that these powerful programs can easily be misused, either accidentally or with malicious intent.

Network security is not a task that can ever be closed off and deemed as having been completed. Even if a comprehensive port scan shows that your system is properly configured today, it may not remain that way for long. Port scanning needs to be a regular maintenance task as new devices are added to the network. Closing off ports, and securing those that remain, is not enough to deter the most determined hacker, but it may be enough to deter the less determined opportunist. As in other areas of security, there should be no room for complacency.

> *"Port scanners are vital pieces of any network security toolkit, and the range of such utilities is quite extensive for all major platforms."*

PCNA