Global Information Assurance Certification Paper

Port Scanning Techniques and the Defense Against Them

Roger Christopher

October 5, 2001

Introduction

Port Scanning is one of the most popular techniques attackers use to discover services that they can exploit to break into systems. All systems that are connected to a LAN or the Internet via a modem run services that listen to well-known and not so well-known ports. By port scanning, the attacker can find the following information about the targeted systems: what services are running, what users own those services, whether anonymous logins are supported, and whether certain network services require authentication. Port scanning is accomplished by sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can be probed for further weaknesses. Port scanners are important to network security technicians because they can reveal possible security vulnerabilities on the targeted system.

Just as port scans can be ran against your systems, port scans can be detected and the amount of information about open services can be limited utilizing the proper tools. Every publicly available system has ports that are open and available for use. The object is to limit the exposure of open ports to authorized users and to deny access to the closed ports.

Port Scan Techniques

To defend against port scans, you have to understand how port scans are performed. There are various port scanning techniques available. Port scans have been made automated by popular port scanning tools such as Nmap and Nessus.

The following scans are available for standard for Nmap and Nessus.

1. Address Resolution Protocol (ARP) scans discover active devices on the local network segment by sending a series of ARP broadcasts and incrementing the value for the target IP address field in each broadcast

packet. This type of scan will have every IP device on the network respond with its own IP address in response. This scan will effectively map out an entire network.

2. The Vanilla TCP connect scan is the most basic scanning technique. The scan uses the connect system call of an operating system on a target system to open a connection to every port that is open. The scan is extremely noisy and easily detectable. The targeted system logs will show connection requests and error messages for the services that accepted the connections.

3. The TCP SYN (Half Open) scans are called half open because the attacking system doesn't close the open connections. The attacking scanner will send a SYN packet to the target and wait for a response. If the port is open, the target will send a SYN|ACK. If the port is closed, the target will send an RST. This type of scan is difficult to detect. The target system is in charge of closing the open connections and the target, most likely, will not have the proper logging set up to detect this type of scan.

4. The TCP FIN scan has the ability to pass undetected through most firewalls, packet filters, and scan detection programs. The attacking system sends FIN packets to the targeted system. The closed ports will respond with an RST. The open ports will ignore the packets. The attacking system will take note of which ports it received an RST on and report on the ports that did not respond with an RST.

5. The TCP Reverse Ident scan discovers the username of the owner of any TCP connected process on the targeted system. This type of scan allows the attacking system to connect to open ports and use the ident protocol to discover who owns the process.

6. The TCP XMAS scan is used to identify listening ports on the targeted system. The scan manipulates the URG, PSH and FIN flags of the TCP header. If the port is closed on the targeted system, the target will send an RST. If the port is open, the port will ignore the

packets.

7. The TCP NULL scan uses a series of uniquely configured TCP packets that contain a sequence number but no flags. It the target's TCP port is closed, the port will send an RST. If the port is open, the port will ignore the packet.

8. The TCP ACK scan is used to identify active web sites that may not respond to standard ICMP pings. The scan utilizes TCP packets with the ACK flag set to a probable port number. If the port is open, the target will send an RST in reply. If the port is closed, the target will ignore the packet.

9. The FTP Bounce Attack uses the ftp protocol support for proxy ftp connections. The beauty of this scan is that the attacker can hide behind an ftp server and scan another target without being detected. The downside is that most ftp servers have this service disabled.

10. The UDP ICMP port scan uses the UDP protocol. This port scan is successful in finding high number ports, especially in Solaris systems. The scan is slow and unreliable.

11. The ICMP ping-sweeping scan utilizes the ping command to do a sweep to see what systems are active. The downside is that most networks have the ICMP protocol either filtered or disabled.

(Fyodor)

Defending Against Port Scans

If you have a publicly accessible server, the system will be vulnerable to port scans. There is no certain way to defeat port scans. The court systems have determined that performing port scans is not illegal. Port scans are illegal only if the attacker uses information from a port scan to exploit a vulnerability or open port on the system. So the question is: How do we limit the information our systems will give out?

One way to limit the information gained from port scans is to close unnecessary services on the targeted systems, i.e. if you are running a web server, http should be the only service offered. On UNIX systems, the easiest way to limit the information given to port scanners is to edit the /etc/inetd.conf and remark out any unnecessary services. You will also want to edit the /etc/init.d and the runlevel file that your system is utilizing. Remove any unneeded services. Also, ensure that your system is not running in the X11 mode. If you are running in the X11 mode, your system will broadcast the 6000 service whether you are logged in or not.

Another way to limit the information given to port scanners is to employ TCP Wrappers, where applicable. TCP Wrappers give the administrator the flexibility to permit or deny access to the services based upon IP addresses or domain names. TCP Wrappers work in conjunction with the /etc/inetd.conf file. TCP Wrappers works by invoking the tcpd daemon prior to providing the specified service. When an incoming request is detected coming in from an authorized port, TCP Wrappers will first check the /etc/hosts.allow file to see if the IP address or domain name has permissions to access the service. If no entry is found, TCP Wrappers will check the /etc/hosts.deny file. If no entry is found there, or if the statement ALL : ALL is found, TCP Wrappers will ignore the request and not permit the requested service to be utilized. When the system is port scanned, TCP Wrappers will still allow the service to be advertised, however, the scanner will not receive any additional information from the port unless the scan is coming from a host or domain specified in the /etc/hosts.allow file. When scanned, the system will list the service as being open. When the attacker tries to exploit the open port, TCP Wrappers will reject the incoming connection if it is not originated from an approved host or domain. The drawback of TCP Wrappers is that not all services are covered. Services such as http and smtp are not covered, and if improperly configured, will be susceptible to exploit. TCP Wrappers is not susceptible to IP spoofing. When an incoming request is detected, TCP Wrappers will perform a reverse DNS lookup on the requesting IP address. If the reverse lookup matches the requesting IP, TCP Wrappers will permit the connection. If the reverse lookup fails, TCP Wrappers will assume that it is dealing with an unauthorized host and will not permit

the connection.

Finally, another way to limit the amount of information given to port scans is to utilize PortSentry offered by Psionic. PortSentry detects connection requests on a number of selected ports. PortSentry is customizable and can be configured to ignore a certain number of attempts. The administrator can select what ports PortSentry will listen to for connection requests and the amount of invalid requests. The administrator will list ports that their system is not supporting. Upon detection, PortSentry will employ TCP Wrappers and make an entry into the /etc/hosts.deny file for the suspected intruder. PortSentry will also setup a default route statement for the offending system. The default route statement will route all packets from the offending system to either another system or a dead system. The result is that the targeted system will appear as being non-existent. On Linux systems, PortSentry is able to detect all TCP and UDP scans, whereas on Solaris systems are only able to detect the TCP Vanilla and UDP scans.

Conclusion

Every system is vulnerable to port scanning. The best offense is a good defense. Never accept the default installation of operating systems. Most default installations have numerous ports open to allow a greater flexibility. Before a system is placed online, a port scan should be performed against the system. If more ports are open then required, close those ports. The greater number of services that your system offers, the more vulnerable the system. Periodically check the /etc/inetd.conf, the /etc/init.d file and the run control files on your system for unneeded services. If a system has been compromised, attackers may try to open up more ports on the system so they can more easily exploit the ports weaknesses. The more vigilant the system administrator is, the more resistant their system will be to penetration and the least likely they will be exploited.

References

1. Anonymous. Maximum Security, Second Edition. Indianapolis: SAMS, 1998. 177 – 180.

2. McClure, Stuart; Scambray, Joel; Kurtz, George. Hacking Exposed, Network Security Secrets & Solutions. Berekley: Osborne/McGraw Hill, 1999. 38 – 51.

3. Fyodor. "The Art of Port Scanning." September 01, 1997. URL: http://www.insecure.org/nmap/nmap_doc.html. (September 25, 2001).

4. Psionic Software, Inc. URL: http://www.psionic.com/abacus/portsentry. (September 25, 2001).

5. Venema, Wietse. "TCP Wrappers 7.6 BLURB." March 21, 1997. URL: ftp://ftp.porcupine.org/pub/security/tcp_wrappers_7.6.BLURB. (September 25, 2001).

Last Updated: May 30th, 2017

Upcoming Training

SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event