Global Information Assurance Certification Paper

Malicious Scanning

Kathryn Kerr
March 21, 2001

Introduction

Malicious scanning is a reconnaissance technique used to collect information about a target's machine or network to facilitate an attack against it. Scanning is used by attackers to discover what ports are open, what services are running and identify system software—all to enable an attacker to more easily detect and exploit known vulnerabilities within a target machine. This article explains how scanning works, describes different scanning techniques and the extent to which scanning can be detected and stopped. Although this article focuses on malicious scanning, it is important to note that many probes occur quite legitimately as part of normal TCP/IP activity.

Background

Before describing how malicious scanning works, it useful to understand a little about the Transmission Control Protocol/Internet Protocol (TCP/IP). Sometimes it is useful to think of the TCP/IP suite of protocols as being divided into four basic layers where each layer depends upon the layer beneath and where each layer plays a different role in moving data from source to destination on the Internet. (The Open Systems Interconnection (OSI) model identifies seven layers of network communication). However, the TCP/IP four-layer model and the OSI model are just conceptual models—the TCP/IP suite consists of more than 100 different protocols not layers per se, which are used to connect computers within a network and to transmit data.[1]

In the TCP/IP model, the **application layer (1)** prepares the messages and instructions to be sent and is where the programs and services that use the data reside. The **transport layer (2)** handles host to host data delivery services by converting messages into packets and taking responsibility for the sequencing and reliability of the delivery of those packets. The transport layer consists of two main protocols—TCP and the User Datagram Protocol (UDP). The **Internet layer (3)** converts packets to datagrams and controls the flow and routing of datagrams. The Internet Control Message Protocol (ICMP) operates within the Internet layer enabling routers to send error or control messages to other routers or hosts when required.[2] The **network access layer (4)** transmits the datagrams as individual bits.[3]

This background is relevant to understanding how malicious port scanning works. Specifically, when specially constructed probes are sent to a destination IP address, the TCP, UDP and ICMP protocols respond in particular ways. Intelligence, which is useful for an attacker, is then generated by the probes' interaction with these protocols.

Port scanning

Ofir Arkin, author of *Network Scanning Techniques*, describes scanning as the "art of detecting which systems are alive and reachable via the Internet, and what services they offer."[4] All machines connected to the Internet run numerous services that listen at well-known and not so well-known ports. A port is "the final portion of the destination address for any piece of Internet traffic."[5] Through port scanning, it is possible to find what TCP and UDP ports are available and being listened to by a service.[6] For example, if ports 21, 25, 80 and 110 are open this indicates the machine contains four servers for file transfer (FTP), inbound e-mail (SMTP), web (HTTP) and outbound e-mail (POP3). Each listening port represents a potential communication channel

with the target and a potential point of attack.[7]

At its most basic level, port scanning occurs when a data packet (or probe) is sent to a port and information about the state of the port is gleaned by the nature of the response (or lack of) sent back to the scanning computer. Port scanning works by harnessing the features of the TCP, UDP and ICMP protocols to provide data to an attacker about the state of target computer's ports. Some of these features are described below under 'techniques'.

Scanning tools

There is a vast array of automated scanning tools freely available on the Internet. These tools have different capabilities and tend to specialise in different types of scanning techniques. An overview of some of these tools can be found at http://www.insecure.org/nmap/press/infoworld-windows_scanners.txt. However, no article on port scanning would be complete without mention of nmap. Nmap[8] is probably the most popular, powerful and versatile of scanning tools, which is available for both Unix and Windows systems.[9] Within a single tool nmap combines most, if not all, of the variety of scanning techniques and can be used to scan large networks.[10]

Scanning techniques

There are many scanning techniques but this article will only describe a few of the more common ones to provide a basic understanding of how scanning works.

Usually before port scanning commences, an attacker will first test the network to see if it is alive using a ping.[11] If multiple hosts are queried at the same time, this is called a ping sweep. There are a variety of ways to conduct ping sweeps, each of which harnesses features of the ICMP, TCP or UDP protocols. A ping works by sending an ICMP ECHO request packet to the targeted system and waiting for an ICMP ECHO reply. If a reply is received the system is alive. No response means the target is down. Broadcast ICMP works in a similar way and can be used by an attacker to map all live hosts on a network. The request will be broadcast to all live hosts on the target network and they will send ICMP ECHO replies to the attacker. This technique, however, will only work on Unix machines.[12] Blocking ICMP traffic at the router will prevent this type of scan.

Once a live network has been detected the next basic test for an attacker is to determine whether a port is listening.[13] The **TCP connect()** scan on nmap achieves this by opening a full connection, using the TCP three-way handshake, on every listening port. Typical command line output from nmap for this type of scan would look something like the following, where –sT specifies the use of the TCP connect scan:

```
# nmap -sT xxx.xxx.x.xx

Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Interesting ports on (xxx.xxx.x.xx):
Port    State       Protocol  Service
7       open        tcp        echo
9       open        tcp        discard
13      open        tcp        daytime
19      open        tcp        chargen
21      open        tcp        ftp
...
Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
```
[14]

Connect() is the fastest scanning technique as it is able to scan ports in a parallel fashion. Its biggest disadvantage for attackers is that it is the easiest to detect and can be stopped at the firewall.[15]

TCP SYN (synchronise) scanning is often referred to as 'half-open' scanning because, unlike TCP connect(), a full TCP connection is never opened. The scan works by sending a SYN packet. If a SYN|ACK (synchronise|acknowledge) is received this indicates the port is listening. The scanner then breaks the connection by sending a RST (reset) packet. However, if a RST is received this indicates the port is closed. The advantage of this technique is that fewer sites log incomplete TCP connections. The disadvantage for attackers is that some packet filtering firewalls do look for, and stop, SYNs to restricted ports.[16]

The **TCP SYN|ACK** scan begins by sending a SYN|ACK packet. The TCP protocol requires that if a SYN|ACK is received without having sent a SYN (and because the port was closed) it assumes this was a mistake and sends a RST to tear down the connection. If however the port was open, the SYN|ACK packet will be ignored.[17]

As the name implies, **TCP FIN** (finish) scanning, initiates a scan by sending a FIN packet to a port. Generally, closed ports reply to FIN packets with a RST. Open ports, on the other hand, tend to ignore the FIN packet. In these situations, TCP FIN scans can be quite effective. However, because Microsoft ports respond to FIN packets differently—they send a RST regardless of the state of the port—this technique can only be used to find listening ports on non-Windows machines and to identify Windows machines.[18]

TCP vs UDP

As the previous examples demonstrate, many scanning techniques exploit features of TCP. TCP is responsible for creating reliable connections using a three-way handshake and ensuring that the data is passed to the correct application or service, which it determines on the basis of its port number.[19] Similarly, it not surprising that the most common scans are directed toward the 65,536 TCP connection-oriented ports because of the good feedback they provide.[20]

UDP is different from TCP in that no connection is established between the originating host and the target prior to the data being sent.[21] Scanning non-connection-oriented UDP ports is slower and more difficult because of the unreliability of the protocol and limited feedback generated.[22] In order to find UDP ports, the attacker generally sends empty UDP datagrams to the port. If the port is listening, the service will send back an error message or ignore the incoming datagram. If the port is closed, then the operating system will send back an "ICMP Port Unreachable" message.[23] Therefore (assuming the datagram did in fact arrive and the network is live), no response generally means that the UDP port is open.[24] Another exception, however, is if the machine has a packet filtering firewall it may not provide information either way. Despite its limitations, UDP scanning can still be worth doing since some of the more serious trojan horse programs like Back Orifice use UDP, specifically because UDP traffic is harder to detect and stop.[25]

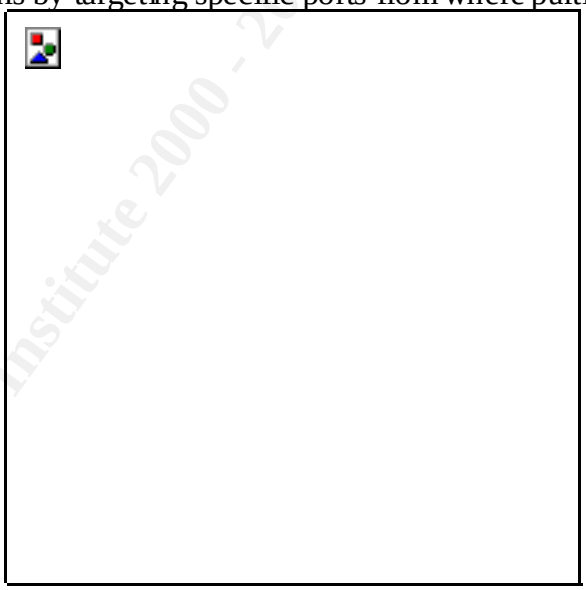
These scanning techniques have mostly described ways in which to detect listening ports on a target machine. Malicious scanning, however, also involves collecting other types of data which attackers need for reconnaissance purposes. These include detecting known vulnerabilities

within protocols and systems and application software; finding previously installed malicious code to exploit and reporting on ease of TCP sequence number prediction for the target. The latter can be used to target machines with a high potential for session hijacking,[26] ie masquerading as a trusted party.

Operating system identification

Sometimes a target's operating system details can be found very simply through self-revelation, for example, by examining its telnet banners or from its File Transfer Protocol (FTP) servers, after connecting to these services. However, where this information is not available stack finger printing provides an effective approach. **TCP/IP stack finger printing** is another nmap scanning technique to identify the particular version of an operating system that is running on a target machine. The 'stack' is a reference to the layer analogy of the conceptual models and stack finger printing gets its name by examining the vendors' (often unique) implementation of TCP/IP protocols for their operating systems.[27] Stack finger printing works by being aware of the differences between operating systems and writing probes which test for these differences.[28] Rarely will identification come from one probe—usually a variety of probe tests will need to be performed to make an accurate assessment or best guess. An example of one type of probe is TCP initial sequence number sampling. After responding to a connection request, information about the operating system can be gleaned from the pattern of the sequence numbers of which there are four types.[29]

Scanning for exploits

Scanning for exploits occurs by targeting specific ports from where particular exploits can commonly be accessed.[30]  For example, SubSeven is a common trojan, some versions of which (eg, 2.1) have a default installation on TCP port 27374.[31] However, the port numbers commonly used by trojans are not iron bound. Any program can use any port number—it depends what port the attacker loading the program chose.[32]

If a trojan program was installed on a system it generally opens a high-numbered port. Then, while an Internet connection exists, that open trojan port could be scanned and located enabling an attacker to fully compromise the system. Firewalls, however, can detect this type of activity

by proactively blocking unauthorised network traffic so if a trojan resides within a machine it could never be found or used to further compromise the system.[33] For a list of commonly exploited ports also see http://www.netice.com/Advice/Exploits/Ports/default.htm.

Scanning for vulnerabilities

Scanning for vulnerabilities occurs by targeting specific ports where vulnerabilities are known to exist.[34] While vulnerabilities can take various forms, there are generally three types of vulnerabilities associated with the services detected at listening ports. These include software errors (including those for which exploits exists) which allow an attacker to perform an unauthorised action that causes harm to the targeted system; misconfiguration errors[35], for example, normal systems services, which, when they aren't required, have not been disabled, such as the Windows file and printer sharing on NetBIOS ports 137 – 139; and procedural vulnerabilities such as failure to provide adequate password protection.[36] Software vulnerabilities can be detected by scanning networks looking for particular software with known vulnerabilities and includes all software used within each of the TCP/IP transport layers. For example, recently, Berkeley Internet Name Domain (BIND) vulnerabilities have been receiving particular attention by attackers. (BIND is a popular implementation of the Domain Name System protocols). One BIND vulnerability, known as the tsig bug (transaction signature), affects a number of versions of BIND and will allow an attacker to gain access to a system using a buffer overflow. In this case, once an appropriate version of BIND has been located, the attacker is able to use an exploit to attack the system. Fortunately in many cases such as this, the vulnerability can be corrected by simple installation of the vendor patches or by upgrading to the most current version of the software.[37]

Stealth and spoofed scans

There are a group of scans which are loosely called "stealth" scans because, as the name implies, they try to evade or minimise their chances of detection. Some stealth scans will pass through firewalls, undetected by the filtering rules. For example, fragmenting the IP datagrams within the TCP header will bypass some firewalls acting as packet filters because they cannot see a complete TCP header that matches their filter rules.[38] Another form of stealth scan occurs at a pace below the threshold (slower) at which the IDS are generally set.[39] If the IDS threshold is lowered to detect slow scans, this results in too many false positives and the scan gets "lost" amongst the data overload, thus hiding in normal network traffic. Other scan signatures may not be logged[40] such as the TCP SYN or half-open scan mentioned previously.

Spoofing refers to methods used to conceal the true identity of an attacker. Spoofed scans occur in a few different ways. For example, the FTP bounce method takes advantage of a vulnerability in FTP servers. By compromising a third party's FTP server, an attacker is able to use it to look for connections on a targeted system. This allows an attacker to probe the ports and get the results without revealing the attacker's own IP address.[41] The hping port scanner uses a similar technique to hide the source of its scans by finding an "idle silent host" through which to probe the target site on the attacker's behalf.[42]

Scanning from the network security manager's view

In the same way there are tools that facilitate malicious scanning activity, there are tools designed to detect and stop malicious scans. In addition to packet filtering firewalls and

intrusion detection systems (IDS), there are a range of other tools which specialise in detecting various types of scans. However, it is important to recognise the limitations of these tools. In describing some of the scanning techniques available I have commented on their ability to be detected using packet filtering firewalls, IDS and routers but none of these tools can stop or detect scans completely.

Nonetheless, using tools which provide some level of protection and which can provide data about the level of activity directed against a machine or network is better than no protection and remaining ignorant about the level of hostile interest that machine or network faces. Without the use of security tools to act as a radar, scanning will continue to occur invisibly in the background but the lack of visibility does not mean it is not occurring. Given the way in which scanning tools can be directed to scan broad ranges of IP addresses,[43] no single machine or network connected to a public network is immune from malicious and indiscriminate scanning.

As a potential pre-cursor to a network attack, logs of scanning traffic provide an indication of the level of threat a network faces by revealing both direct and opportunistic interest in targeting a particular machine or network and provide information about potential sources, even from spoofed IP addresses and access points. Therefore, detection and blocking of scans should be accompanied by monitoring and, if possible, analysis of the logs for clues of potential attacks. Also, as a standard preventative measure, system and network administrators, using tools such as SATAN or SScan2K,[44] should scan their own systems to identify and then treat and minimise vulnerabilities on their networks.[45]

Conclusion

In recent years scanning has become associated more often than not with malicious attacker reconnaissance activity rather than normal network activity. This has come about because of the availability of powerful and easy to use scanning tools such as nmap, which are capable of providing intelligence about large numbers of networks and machines within a relatively small amount of time. Consequently levels of malicious scanning have steadily increased over the past few years. A network's scanning activity provides insight into the potential level of interest which exists to attack that network. The threat is ongoing and, in many cases, increasing as new vulnerabilities continue to emerge yet network security managers have finite resources with which to manage these risks. The challenge for network security managers is not to become complacent about, or overwhelmed, by the volume of scans experienced on a daily basis but to continue to maintain a strong security posture while recognising the limitations of these protective security measures.

References

[1] Dr-K. *A complete hacker's handbook – everything you need to know about hacking in the age of the web*, 2000, Viking Penguin Books Australia, p 53
Phleeger, Charles. *Security in Computing*, 2nd edition, 1997, Prentice-Hall, page 384

[2] Comer, Douglas E. *Internetworking with TCP/IP – Principles, Protocols and Architectures*, 4th Edition, Prentice-Hall, 2000, page 130

[3] Dr-K. *A complete hacker's handbook – everything you need to know about hacking in the age of the web*, 2000, Viking Penguin Books Australia, p 53
Phleeger, Charles. *Security in Computing*, 2nd edition, 1997, Prentice-Hall, page 384

[4] Arkin, Ofir. *Network Scanning Techniques – Understanding How it is Done*, November 1999, http://www.sys-security.com (16 March 2001)
[5] Gibson, Steve. *TCP/IP "Ports"*, Gibson Research Corporation, http://grc.com/su-ports.htm, (12 March 2001)
[6] Mateti, Prabhaker. *Port Scanning* , Wright State University, www.cs.wright.edu/~pmateti/courses/Probing
[7] Fyodor. *The Art of Port Scanning*, www.insecure.org/nmap/nmap_doc.html, (12 March 2001)
[8] Fyodor. *The Art of Port Scanning*, www.insecure.org/nmap/nmap_doc.html
[9] http://www.eeye.com/html/Databases/Software/nmapnt.html, (12 March 2001)
[10] Fyodor. *Remote OS Detection via TCP/IP Stack FingerPrinting*, www.insecure.org/nmap/nmap-fingerprinting-article.html, (12 March 2001)
[11] Jankowski, Richard C. *Scanning and Defending Networks with Nmap*, http://www.saturnlink.com/articles/21700nmap.html, (12 March 2001)
[12] Arkin, Ofir. *Network Scanning Techniques – Understanding How it is Done*, November 1999, page 4, http://www.sys-security.com (16 March 2001)
[13] Mateti, Prabhaker. *Port Scanning*, Wright State University, www.cs.wright.edu/~pmateti/courses/Probing
[14] Jankowski, Richard C. *Scanning and Defending Networks with Nmap*, http://www.saturnlink.com/articles/21700nmap.html, (12 March 2001)
[15] Fyodor. *The Art of Port Scanning*, www.insecure.org/nmap/nmap_doc.html
[16] Fyodor. *The Art of Port Scanning*, www.insecure.org/nmap/nmap_doc.html
[17] Arkin, Ofir. *Network Scanning Techniques – Understanding How it is Done*, November 1999, page 8, http://www.sys-security.com (16 March 2001)
[18] Fyodor. *The Art of Port Scanning*, www.insecure.org/nmap/nmap_doc.html
[19] Dr-K. *A complete hacker's handbook – everything you need to know about hacking in the age of the web*, 2000, Viking Penguin Books Australia, p 56
[20] Mateti, Prabhaker. *Port Scanning* , Wright State University, www.cs.wright.edu/~pmateti/courses/Probing
[21] Dr-K. *A complete hacker's handbook – everything you need to know about hacking in the age of the web*, 2000, Viking Penguin Books Australia, p 57
[22] Mateti, Prabhaker. *Port Scanning* , Wright State University, www.cs.wright.edu/~pmateti/courses/Probing
[23] http://www.netice.com/Advice/Underground/Hacking/Methods/Technical/Port_Scan/UDP/default.htm
[24] Fyodor. *The Art of Port Scanning*, www.insecure.org/nmap/nmap_doc.html
[25] http://hackyourself.com/sample.dyn, (12 March 2001)
http://sans.org/infosecFAQ/homeoffice/scanned.htm, (12 March 2001)
[26] http://sans.org/newlook/resources/IDFAQ/What_is_nmap.htm, (12 March 2001)
[27] Arkin, Ofir. *Network Scanning Techniques – Understanding How it is Done*, November 1999, page 13, http://www.sys-security.com (16 March 2001)
[28] Fyodor. *Remote OS Detection via TCP/IP Stack FingerPrinting*, www.insecure.org/nmap/nmap-fingerprinting-article.html
[29] Arkin, Ofir. *Network Scanning Techniques – Understanding How it is Done*, November 1999, page 14 http://www.sys-security.com (16 March 2001)
[30] http://www.sans.org/infosecFAQ/homeoffice/scanned.htm
[31] http://www.sys-security.com/html/papers/trojan_list.html (16 March 2001)
[32] http://hackyourself.com/sample.dyn (21 March 2001)
[33] http://grc.com/faq-shieldsup.htm#IDENT, (12 March 2001)
[34] http://www.sans.org/infosecFAQ/homeoffice/scanned.htm
[35] Arkin, Ofir. *Network Scanning Techniques – Understanding How it is Done*, November 1999, page 6 http://www.sys-security.com (16 March 2001)
[36] http://grc.com/su-danger.htm, (12 March 2001)
[37] http://www.isc.org/products/BIND/bind-security.html, (12 March 2001)

[38] http://www.netice.com/Advice/Underground/Hacking/Methods/Technical/Port_Scan/fragmented/default.htm (18 March 2001)
[39] Hartje, Roger. "*NFR appliance nips, anlyszes attack*", eWeek - Building the .com enterprise, Ziff Davis Publishing Holdings Inc, 2001
[40] Arkin, Ofir. *Network Scanning Techniques – Understanding How it is Done*, November 1999, page 7 http://www.sys-security.com (16 March 2001)
[41] http://advice.networkice.com/advice/Underground/Hacking/Methods/Technical/Port_Scan/default.htm (19 March 2001)
[42] http://www.securiteam.com/securitynews/A_new_stealth_port_scanning_method.html (19 March 2001)
[43] Jankowski, Richard C. *Scanning and Defending Networks with Nmap*, http://www.saturnlink.com/articles/21700nmap.html, (12 March 2001)
[44] http://neworder.box.sk/search.php3?srch=detect+and+scans (22 March 2001)
[45] http://www.saturnlink.com/articles/21700nmap.html

Bibliography

Arkin, Ofir. *ICMP Usage in Scanning or Understanding some of the ICMP's Protocol's Hazards*, The Sys-Security Group, December 2000, http://www.sys-security.com (16 March 2001)
http://www.sans.org/infosecFAQ/securitybasics/netsec_scanning.htm
http://packetstorm.securify.com/groups/synnergy/portscan.txt
http://209.143.242.119/cgi-bin/search/search.cgi?searchvalue=port+scanning&type=archives
http://www.wiretrip.net/rfp/p/doc.asp?id=13&iface=2
http://grc.com/su-ports.htm
http://hackyourself.com/ports.html
http://www.isi.edu/in-notes/iana/assignments/port-numbers