Global Information Assurance Certification Paper

Nmap - The tool, it's author, and it's implications

By: Brent Deterding - July 13, 2000

Nmap (available at http://www.insecure.org) is the commonly accepted authority in information gathering tools. It is the first tool that both an attacker and a defender reach for, for a reason. It is an extremely versatile and useful information gathering tool that yields much of the necessary information about a machine and it's possible weaknesses. Care must be taken when using Nmap.

What is Nmap ? - Features of Nmap

Nmap started off as a port scanner, intended to aid in the mapping out of possible security holes on a system or set of systems. According to www.insecure.org:

> "Nmap is a utility for port scanning large networks, although it works fine for single hosts. . . . Sometimes you need speed, other times you may need stealth. In some cases, bypassing firewalls may be required. Not to mention the fact that you may want to scan different protocols (UDP, TCP, ICMP, etc.)."

Nmap can specify a machine or set of machines via IP address, DNS name, or network number and can connect to those machine(s) via TCP connect(), TCP SYN, TCP FIN, TCP Xmas, TCP NULL, TCP bounce, TCP ACK, UDP ICMP, ICMP, TCP, and direct RPC. Nmap can fragment packets in several timing schemes to further thwart firewalls that may count the number of packets over a given time period.

One of Nmap's signature features is Operating System detection. OS detection is accomplished through TCP/IP fingerprinting. TCP/IP fingerprinting works by sending a series of non-standard packets to an operating system and seeing what is returned. This "fingerprint" is matched to a database that ships with Nmap containing several hundred operating system "fingerprints." Nmap also supports features such as dynamic ttl times, parallel scanning and pinging, flexible target and port specification, decoy scanning, and output to text or machine-readable formats.

Nmap is an open-source project, meaning that it's source code is freely available to look at, modify, and use at one's will. Nmap is available on many OS's simply because users of those OS's took the code and ported it to the OS of their choice. Being open-source, Nmap is, of course, free. Although free, it offers more functionality and better performance than it's pricey brethren. Nmap is in wide use by anyone needing information on a system, be they the system administrators or someone attempting to attack or otherwise compromise a system.

Who wrote Nmap ? - Fyodor

Fyodor wrote Nmap as a port scanning tool, as was mentioned above. Fyodor recently gave an interview to SecurityFocus.com (http://www.securityfocus.com Audio/Visual->Interviews->Fyodor). In this interview Fyodor stated as one use of Nmap that he found most gratifying was the use of Nmap to find proxies to bypass censorship, as experienced in China. Nmap has many other uses - which include both (arguably) "good" and "bad" uses. Fyodor also mentioned that Nmap may develop into a full-blown commercially-available vulnerability scanning utility, while keeping the existing features in the free tool. Also discussed in the interview was a version for Windows NT, which recently became available from http://www.eeye.com/html/Databases/Software/nmapnt.html. Fyodor will gladly accept an NT port for Nmap, although he will not develop it himself. From insecure.org:

> "All my programming work is for UNIX as I believe Windows98/NT is still too primitive for power users. I also believe that Microsoft intentionally engages in anti-competitive and ethically challenged actions which can have devastating results for the industry and consumers."

SecurityFocus.com asked Fyodor if he was concerned about theory stating that Nmap could be banned by a government. The possibility has been discussed in mailing lists and open forums. However, Fyodor sees no need for worry over this perceived threat, as he does not believe that it could be banned due to it's large dissipation and the fact that it is open-sourced.

Fyodor is a self-proclaimed hacker. He thinks of a hacker as a person with a passion for technology and exploring it's limits. This would apply to those who find and publish security holes, (gray-hats) as well as those hackers who commit crimes using the information (black-hats), although they are not hackers because they committed crimes, but because of the skills they used to commit them. Fyodor has a distinct dislike of "script-kiddies." It has been argued that tool authors - such as Fyodor, are responsible for script kiddies having the success that they have. However. system administrators (white-hats) need security scanners, despite the fact that attackers can use those same tools to uncover security holes. Is it better to expose security holes until they get fixed or to not have tools such as Nmap available, thus making holes harder to find? It is commonly accepted that security through obscurity is really no security at all. Fyodor agrees with this assessment, and goes on to state that he thinks script-kiddies have actually helped the security field, as they are constantly scanning networks and finding holes. This constant scanning reveals holes that are occasionally exploited; typically with the consequence of a company's public embarrassment. However, these holes are often quickly fixed, leaving it closed for other potentially more dangerous attackers such as corporate espionage or malicious intruders. This also increases the likelihood of other holes being discovered by both the victim company and other companies/organizations before they can be exploited. Although security through obscurity is no security at all, there is argument that the large volume of exploits made known does more to harm security then it does to help it. Fyodor argues that it is this large volume of exploits that has brought a higher understanding and concern of security issues. He points out the state of affairs several years ago; when compared to today, we are much more secure. That is, companies are taking more responsibility to make their products more secure. For example, consider Microsoft. Compare the security offered in Windows 95 versus the security offered in Windows NT. Clearly, steps have been taken in the right direction in the past several years. In future versions of Nmap, expect to see socks proxy bounce scanning, IP ID scanning, traceroute support, more input and output forms, faster scans, Solaris binaries (ala RPMs), an improved front-end, a default configuration file (.nmaprc), and making Nmap a shared library.

Implications of using Nmap to scan networks

Nmap is, without dispute, a very powerful tool that has many uses for the white, gray, and black hats of the security community. There are, however, several important factors to consider before using Nmap to scan a single machine or a large network. Although port-scanning is legal (in the U.S.), it is not "nice." Scans from unauthorized sources can be deemed as an invasion with unpredictable results. As an example I will use my own experience with Nmap. I asked to have the task of scanning a class B network to see what I could find. I was not very experienced, but had taken several precautions. I had obtained permission from my supervisor (networking), as well as from the chief security officer, whose purview included network scanning. In my inexperience, I chose to perform a full scan with TCP SYN packets on the default timing. I ran the scan three times per week. I used the results from the scan to gather any information I might find useful. Several illegal ftp sites were shut down and some OS accounting information was gathered. However, people noticed my scan, which had come unannounced to them. My supervisor knew, the security officer knew, but no one else did. Better care should have been taken to publish the scan to those who would notice it (such as "we will be scanning you sometime this month"). Although useful information was gained, it was a pie-in-the-face for the organization and for me. A well documented, narrow scan should have been performed, as opposed to my "scan everything" approach. I mentioned unpredictable results; there are more. Many people had personal firewalls in place and saw my scans as an attack. Not only did they flood the security officer with demands for my head, but they retaliated. As I mentioned, I was inexperienced and my box was easily exploited. No one did anything

malicious, but they let me know that they got in and didn't appreciate my scans. As another nasty side-effect, some older machines crashed as a direct result of my scanning them. They were later fixed, but my scan interrupted batch jobs on several servers. My seemingly innocuous scans had wide-ranging implications that impacted many people and did much bad for the image of security at my organization. Overall, some information was gleaned from my scans, but no more that could be gained from a simple scan that is much less invasive. I am now also a security officer and perform network scans routinely in my normal duties. I gather much more information now then I ever did. My scans runs faster, give me more useful information, have a high signal to noise ratio (lots of useful info and little-to-no fluff), and adversely affect no one. Scanning is a job that any security officer should perform on a routine basis. However, care should be taken not to adversely affect anyone and to acquire prior written authorization. In this manner, scanning can yield many benefits without having any adverse effects.

Previously, it was mentioned that Nmap arguably has both good and bad uses. The good uses of Nmap are apparent; Nmap is a very powerful information-gathering tool which helps many security professionals daily. However, it can be argued that Nmap has features that provide for bad uses. In this context, a bad use is a use which holds benefit for attackers and nothing but detriment for defenders. One such feature that could be argued to provide a bad use is the ability of Nmap to fragment packets, coupled with timings that send very few packets spread over a long period of time. When packets are fragmented and come through very seldomly, a firewall or intrusion detection device is unlikely to notice anything suspicious. The slow timing insures that no alarms go off by having a large number of packets, a large amount of data, or connections to many ports of a single device or port in a given period of time. Fragmenting the packets allows an attacker to possibly sneak packets through a firewall by confusing/bypassing packet filters. Because packet filters only look at the first fragment of a packet attackers can conceal data with overlapping fragments and headers. This can be useful for a DDoS attack. Another feature that can provide for a bad use is decoy scanning, where Nmap is used to generate many false scans which divert attention from the true scan. These bad uses certainly benefit the attacker community, but do they help the good guys? Isn't this security through obscurity? To a point, yes, it is. Yet, if fragmentation wasn't incorporated into Nmap would it leave a security hole open? Perhaps it would. Of what use is decoy scanning to a defender? There is no good use of decoy scanning to a defender. So maybe fragmentation is a good thing, it makes us prepare for the real-world possibility of dealing with fragmentation and seal a possible security hole. However, I would argue that decoy scanning is a feature that Nmap does not need, as it only serves to benefit the attacker and be a detriment to the defender. There are other such features in Nmap that aren't mentioned here, these two were chosen to illustrate that some features that may appear to be harmful may be a good feature after all, such as fragmentation, while others may continue to be regarded as bad features to have, such as decoy scanning.

1. Fyodor, "Nmap: Stealth Port Scanner for Network Security Auditing, General Internet Exploration and Hacking"
URL: http://www.insecure.org/nmap/index.html#intro (13 July 2000)

2. SecurityFocus.com "Interview with Fyodor"
URL: http://www.securityfocus.com Audio/Visual->Interviews->Fyodor (13 July 2000)

3. Rich Jankowski, "Scanning and Defending Networks with Nmap" 20 February 2000
URL: http://www.linuxsecurity.com/feature_stories/feature_story-4.html (13 July 2000)

4. Elizabeth D. Zwicky, Simon Cooper, & Brent Chapman, "Buiilding Internet Firewalls"
O'Reily and Associates, June 2000(13 July 2000)

5. SANS, "IP Behavior"