Global Information Assurance Certification Paper

Brenden Claypool – 2002
v1.4

Stealth Port Scanning Methods

Abstract

Know Thy System. This simple phrase is the basis for two things: attack and defense. In any attack, one of the first events is usually reconnaissance. Attackers want to know definitively what kind of system they are dealing with, and what services are running in order to make their job of attacking the computer easier. Administrators also want to know what ports are open so that they can be limited in some manner. Port scanning and stealth scanning technology in particular is in a constant race to stay ahead of scanning detection tools in order to remain undetected. There are many types of stealth scan techniques available to the attacker and administrator such as half-open, Xmas tree, UDP, Null, etc. These have been developed in part, to constantly stay ahead of scan detection technology. As one stealth method is discovered, another is invented and implemented.

What is a port?

A computer that is connected to a network will most likely have open ports. Ports "are used in the TCP [RFC793] to name the ends of logical connections which carry long term conversations." [1] For one computer to talk to another, one computer must have an open port that is listening for the other computer to connect to. The destination computer will have one or more open ports that the source computer knows about, in order to make a connection. There is a maximum of 65535 open TCP (Transmission Control Protocol) ports on a TCP/IP connected computer. There is also thousands of UDP (User Datagram Protocol) ports that are available. The Internet Assigned Numbers Authority (IANA) has broken down these port numbers into three categories:

- Well-Known Ports are those from 0 through 1023
- Registered Ports are those from 1024 through 49151
- Dynamic and/or Private Ports are those from 49152 through 65535

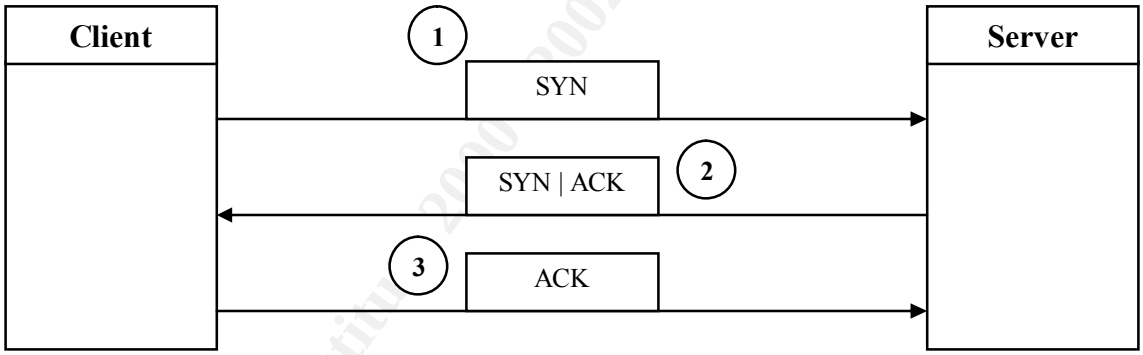
Well-Known ports are used by system processes, or those requiring special privileges These are the most common ports that network services run on, and as such are the some of the most popular targets for attack. The IANA is responsible for maintaining the Well-Known ports. The second group, the Registered Ports, is a large group of loosely based ports which have no enforced restrictions, which is also true for the Dynamic/Private ports. The Well-Known ports are also known as Standard ports, whereas any port above 1023 is known as a Non- Standard port. Opening a Standard port does require special system privileges, but any program can open an unallocated Non-Standard port. While it is true that the IANA is an attempt at standardization, almost any service on a computer can be run on any port.

What is port scanning?

Port scanning is a technique used by attackers, curious individuals, and administrators to collect information from computers connected to a network. System and network administrators use port scans to identify open ports to a system so that they may be able to limit access to those ports, or shut them off entirely. Attackers use port scanning in the same way that administrators do, but with malicious intent. There are many techniques used in stealth scanning, ranging from those that prevent being detected by logging systems, identity concealment, to confusing the server with invalid information. Each one of these techniques is interesting in their implementation and execution.

Non-Stealth Scanning

Non-stealth scanning will be discussed here as a comparison against stealth scanning and as a motivational factor for the rise of stealth scanning. In a TCP connection, a 3-way handshake occurs to make a connection.



When a client wants to connect with a server, it first sends a TCP packet with the SYN (Synchronize Sequence Number) flag set. The server then sends back a TCP packet with the SYN and ACK (Acknowledge) flags set if the port is open on the server. A RST (Reset) packet is sent to the client if the port is closed. If the port is open and the server sends back the SYN|ACK packet, the client computer then sends an ACK back to the server. [2] A port scan typically will have an IP address, or range of IP addresses and a list of ports to look for. A non-stealth port scanner will employ the use of the TCP connect() method of connecting to the destination host. The connect() is a system call provided by the operating system to open a connection to a remote host. [3] This is the most basic of port scan techniques. The TCP connect() uses the 3-way handshake and will succeed if the port being scanned is listening, otherwise it will fail. This is a common method of scanning and at the outset of a scan the user will have a list of ports that TCP connect() was able to reach.

The simplicity of this type of scan is also its downfall from an intrusion detection point of view. It is useful to note that the TCP connect() method does not require any special privileges on the computer it is running on, since the TCP packets do not need to be formed in any kind of special manner. While this type of scan will return some very useful results, the scan activity will be very visible to the administrator of the computer being scanned because there are many attempts to connect to the computer on different ports in a short timeline. Even the most basic logging setup will report these attempts to a diligent administrator. Services that get confused when a connection is made and then immediately disconnect will leave errors in log files. [4]

The Rise of Stealth Scanning

Given how easy it is for a system administrator to notice port scans, it is not surprising that stealth is becoming a major factor that administrators have to deal with. The less information that an attacker can give away to the remote computer, the more time he/she has to do reconnaissance without being noticed. Over the past few years, stealth scanning has dramatically increased in popularity with attackers. There have been many different techniques developed to perform stealth scans which all have a common thread: avoid detection by system administrators and intrusion detection systems.

System administrators typically use port scanning to find open ports, and a TCP connect() scan will suffice. It will find open ports as well as a stealth scan will, but it does it without the overhead of a stealth scan. Stealth scanning is a technique used mainly by attackers, and can be considered a major threat. However, a diligent system administrator should run stealth scans on his/her computers in order to test logging facilities and intrusion detection systems for their capability to detect these stealth scans.

Inverse Mapping

One of the first stealth scans to appear was the Inverse Mapping scan, which was reported in 1998 by the CERT® Coordination Center. The idea was that "intruders send packets that normally would go unnoticed or cause no unusual behaviour to a list of addresses" [5] Attackers use specially crafted packets with customized flags, which in this case included RST (Reset) and SYN-ACK packets and DNS response packets. This type of scan did not find out information about the ports specifically that were open, but rather tested the host to see if it would respond. A computer that exists and is connected to the network would respond to the request, while a non-existent computer would generate an ICMP host unreachable error message. In this manner, an attacker could stealthily map out a network.

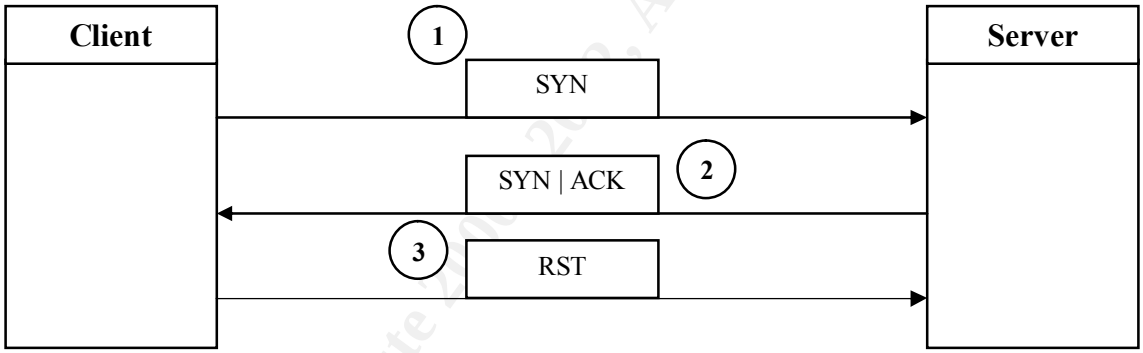
Slow Scan

Another approach to stealth scanning is the slow scan. This is a low-tech solution to the problem of being logged or noticed by the remote system. A "normal" scan will go through thousands of ports within a short time frame, usually under a minute. The nature

of TCP connect() scans allows them to be easily logged and discovered by the system administrator. While scanning very quickly can fill a log with information, slow scanning becomes a very viable option. By waiting for a given amount of time between scans for individual ports, logging programs can be defeated. The downside to this type of stealth is the time factor involved. To be stealthy enough to be undetected by an intrusion detection system or a system administrator can take a very long time. For example, a port scanner is set to scan one host with ports ranging from 1-1024 and given a time lapse of 5 minutes between each port, will take approximately 85 hours. There is nothing fancy about this method, but it does prove that unless a history is kept of all the attempts to each port, detection becomes very difficult. [5]

Half Open Scan

The evolution of the TCP connect() method to include stealth is through a technique called SYN scanning, or half-open scanning. The title originates from the method used to connect to ports on a host. While the TCP connect() method uses the full 3-way handshake to connect to a port on a host, the SYN scan uses a modified handshake which only includes a 2-way communication channel.



The SYN scan begins exactly the same way that the TCP connect() method does by having the client send a packet with the SYN flag set. Similarly, the server then sends back a SYN|ACK packet to the client if the port is open. If the port is not open, a RST (Reset) packet is sent to the client. This is where the SYN scan and the TCP connect() method differ: a final ACK packet is never sent back to the server acknowledging that the client has received the SYN|ACK packet from the server. Instead, a RST packet is sent to the server in order to destroy the connection. In this manner, a full TCP connection is never established between the client and server. Although this method is fairly stealthy, there are a number of drawbacks in using it. Firstly, a logger that is configured to log all SYN connection attempts will capture port scan attempts employing the half-open scan technique. [5] This is due to the fact that the half-open scan still uses a SYN packet as the first step in the communication process. Also, because the packets involved in this type of scan must be custom built, most operating systems require root or administrator access to the underlying system calls. This is a protection measure put in place to prevent invalid packets from being created by any user on the system. Since this scan is fairly well known, most firewalls are configured to detect and deter this type of scan.

FIN Scan

As techniques and software advance in the struggle to maintain security against stealth scans, so do the scanner techniques and strategies. The security community is in almost perpetual catch-up mode to raise their level of security to meet the level of insecurity that the attacker puts forth in the advancement of attacking and probing technologies. The FIN (Finish) scan is an answer to the possible logging capabilities of the SYN scan. Some packet loggers and firewalls are configured to detect SYN packets to restricted ports. In the FIN scan, a packet is sent with just the FIN flag set. If the port is closed, the host sends back a RST flag, whereas an open port simply ignores the packet and nothing is returned to the client. This is required behaviour as set out in the RFC for Transmission Control Protocol. [6] It is through exploiting the requirement that TCP has for ensuring packets arrive at their destination that attackers can probe open ports and possibly evade detection. Because a firewall or packet logger may be setup to detect SYN packets, a FIN packet would slip through unnoticed.

Xmas Tree Scan

Like the FIN scan, the Xmas tree scan employs the use of invalid packet header flags to elicit a response from a host regarding open ports. There are a few different methods that have been applied that all use the Xmas tree scan name. Nmap executes the Xmas tree scan using 3 packet header flags, which are the FIN, URG (Urgent), and PSH (Push) flags. [3] This type of scan is very similar to the FIN scan, with 2 extra flags set. Other Xmas tree scanners set all TCP header flags to be on, which is most likely where the name is from. Like FIN scan, a closed port will return a RST packet, whereas an open port will ignore the packet.

Null Scan

The Null scan produces a reaction to the FIN and Xmas tree scans, but differs in packet header flags. Instead of turning on flags in the header that would cause the packet to be received by the host as an invalid packet, the Null scan turns off all header flags. [5] This again causes a RST packet to be sent to the client if a port is closed, but is ignored if the port is open. Microsoft operating systems in addition to a number of others have ignored the RFC for TCP and have implemented it somewhat differently than the standard. This causes the FIN, Xmas tree, and Null scans to fail on Windows based operating systems, and others such as Cisco, HP/UX, and IRIX. These operating systems all send a RST if the port is open instead of ignoring the packet. [5]

UDP Scan

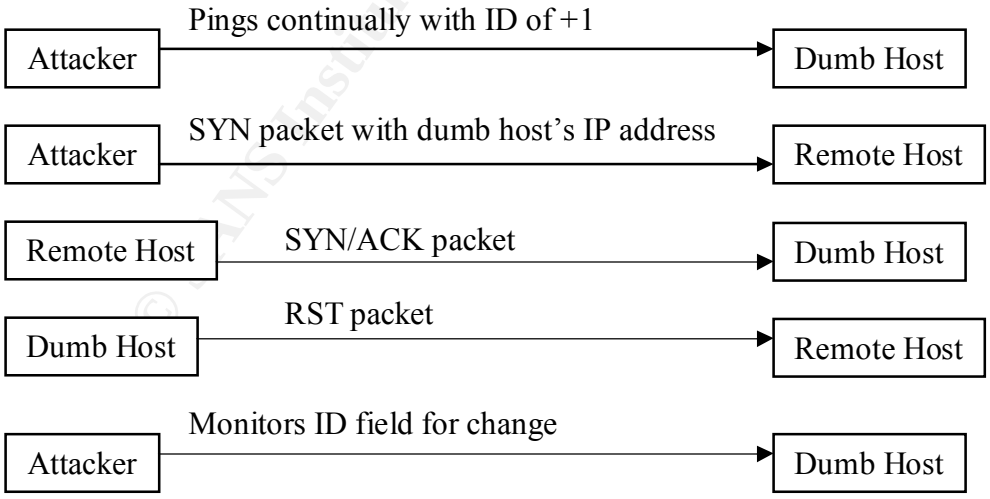
There is some debate over UDP scanning as to the relative usefulness it provides. On one hand, UDP is a much simpler of a protocol than TCP is, given that it is not connection oriented, that is UDP does not concern itself with ensuring that packets arrive at their destination successfully. On the other hand, UDP scanning can be a very useful

tool for finding open, undocumented UDP ports that any given service may be running on. [5] There are a number of vulnerabilities that can be scanned for by using UDP scanning. Programs can easily open high UDP ports without the user's knowledge. These are mainly undocumented, thus tracking them down is made much easier with the application of UDP scanning. Currently there is only one known method for UDP scanning, which entails sending a 0 byte UDP packet to each port on the host machine. If a port is closed, an ICMP port unreachable error will be returned, otherwise it can be inferred that the port is open. [2]

Dumb Scan

The dumb scan method of stealth scanning involves the use of a third party computer that receives very little or no network traffic. This third party is also known as a dumb host. Typically, attackers search for these computers on cable modem subnets looking for Windows-based computers that have been left on at night. The dumb host method of stealth scanning requires a utility to generate customized TCP packets, and a ping utility. The following procedure for dumb scanning is fairly basic, but extremely effective. Firstly, the attacker sends a repetitive ICMP ping to the dumb host with an ID number of +1. Secondly, the attacker sends a spoofed SYN packet to the host with the dumb host's IP address in place of his/her own. The destination port is set to the port that the attacker wishes to scan. Because the host receives the TCP packet with the IP of the dumb host, any reply to a connection request will be sent back to the dumb host. The continuous pinging of the dumb host reveals whether the port is open on the host or not. Typically, if the port is open, the ID number will increase, whereas if the port is closed the ID will most likely remain at +1. [7]

Procedure:



The dumb scan is very effective, and very stealthy. By utilizing a third party, connection attempts are concealed and most logging capabilities by an intrusion detection

system are thwarted. This is due to the fact that no information is communicated directly from the remote host to the attacker's computer.

Fragmentation, Decoying, and Spoofing

There are a number of methods for scanning a computer and making it very difficult for that system's administrator to track and record the reconnaissance activities of the attacker. There are a number of techniques that, instead of being concerned with hiding from being detected by loggers or administrators, are more concerned with protecting the identity and location of an attacker.

Fragmentation

The main idea behind fragmentation is to use very small, broken up IP packets. If the TCP header is broken up into many smaller pieces, it is much more difficult for packet filters, intrusion detection systems, and system administrators to detect what the attacker is doing. [3] Firewalls and packet filters that queue up all IP packets will most likely to detect this kind of a probe, since all of the fragmented packets would be collected and analyzed before letting them pass. These packets are very small and usually of different size. Many programs cannot cope with packets of such small size and shape. The point of this type of protection is not so much to protect the identity of the attacker, but more to conceal the intention of the packets being transmitted to the server.

Spoofing

The main technique that attackers use to conceal their identity and location is called spoofing. Spoofing is "a technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host." [8] Spoofing today can include an attacker who uses the IP address of a trusted host, but can also refer to a person who uses a spoofed IP address which might be picked at random. In either case, the intent of an attacker using a spoofed IP is to conceal his/her identity, and make it appear to the computer that the incoming port scans are coming from somewhere else. Knowing the IP of a trusted host would almost ensure that a port scan would be less noticeable to a system administrator of intrusion detection system, which would in turn allow the attacker more time to do reconnaissance before being detected. Spoofing IPs at random gives the attacker an almost unlimited range of IPs to choose from, and thus makes tracking down the attacker that much more difficult. This type of spoofing is much more obvious to the intrusion detection system or system administrator in that the IP that shows up on the host is not a "trusted" or known IP, which would raise an alarm.

Another spoofing technique involves the Address Resolution Protocol. ARP deals with the interaction between the IP and Ethernet protocols. Any computer that is connected to an Ethernet network running IP (TCP/IP most frequently) has two addresses that it uses for communication. The MAC address (Media Access Control) is, in theory, an unchangeable and unique address in all space and time stored on the Ethernet card.

The other address is the IP address, which is used by applications regardless of what network hardware is operates on top of. ARP basically deals with negotiating between these two protocols; allowing the Ethernet protocol to find the MAC address of the destination, given the destination's IP address. [9] A table is kept of IP addresses and MAC addresses to keep network traffic down. It is through construction false ARP request and reply packets, an attacker can scan a network undetected. By spoofing ARP information, an attacker can appear as a different computer, especially on a local area network environment. Although the MAC address is supposed to be static and different on every Ethernet interface, there are programs for Linux and a number of other operating systems that allow the MAC address to be easily changed at will. This ability to manipulate MAC information can easily make port scanning activity appear that it is coming from a different computer than it actually is.

Decoying

Another technique involved in identity protection is using decoys. Decoys primarily consist of a set of spoofed IPs which are sent to the server during a port scan. This technique allows an attacker to either completely mask off their actual IP address from a host, or bury it within a group of spoofed IP addresses. When an attacker uses solely spoofed IP addresses in a scan, it creates a major problem in that any one of the IP addresses could have been the attacker, and thus can take a lot of time and effort to track down the actual originator. Leaving the attacker's IP address in a group of IP address also makes for a difficult task of weeding out decoy IP addresses to find the real originator of the port scan. This is especially true when the attacker chooses decoy IP addresses that are valid and operating, since the host being attacked would see all valid addresses. The more decoys that are used in a stealth scan, the higher degree of difficulty is involved in tracing the scans back to the actual IP address that was used in the scanning. This is especially true of hosts that are not protected in any way via a firewall or packet filter that might possibly filter out the offender's IP address and alert the system administrator to the problem.

Bouncing

Bounce scanning is another technique, which allows an attacker to camouflage his/her scanning activities. Essentially, attackers "bounce" their scans through services running on other computers that allow commands to pass through, in effect covering their tracks. There are a number of services such as FTP and Finger, which allow a person to execute commands that are directed towards a service on one computer, but bounce off and go to another computer.

FTP

Bouncing off of an FTP server "takes advantage of a vulnerability in the FTP protocol itself" [10] in that the original specification for the FTP protocol included support for proxy ftp connections. While ftp proxy support may have been a desired

feature in 1985 when the RFC for FTP was written, but this is a serious vulnerability in today's Internet. [3] The built-in support for proxy FTP connections allows a person to connect to an FTP server, and then ask that server to send a file to another server. This "feature" can be exploited by an attacker by connecting to an FTP server behind a firewall, and then scan ports that would normally be blocked by a firewall or packet filter from outside the internal network. If there is a read/write directory on an FTP server (and there almost always is), an attacker can send arbitrary data to ports that are found to be open. [10] A technique that is used to perform such actions is to use the PORT command of FTP to declare that a passive user data transfer process (DTP) is listening on the target computer on a specified port number. Then if the attacker uses the LIST command to show the contents of the current directory and the response is then sent over the server DTP. If the port is listening on the host, the transfer process will be successful indicating an open port, but if the port is closed an error will be generated. This method is useful for scanning behind firewalls and concealing the identity of the attacker, but it is also slow and somewhat tedious.

Finger

Finger scanning is most useful for scanning for computers that are running a finger server. Because most finger servers support recursive queries, an attacker can run a query such as "brenden@foo@bar", which will ask "bar" to resolve "brenden@foo", which in turn causes "bar" to query "foo". [3] This type of bouncing also allows the attacker to hide the original source of the finger request.

Tools of the Trade

There are many port scanning tools available for almost every operating system that can be connected to a TCP/IP network. The biggest and most feature-rich one is probably NMap. While there are other scanning tools that exploit specific problems with TCP/IP implementations such as Hping2, the vast majority do not have any stealth technologies built-in at all. This is good news for the system administrator, since these will be very easily picked up by logging programs and even the most basic intrusion detection system.

NMap

Nmap is considered to be one of the best all-round scanning, and stealth scanning tools available today. "Nmap is designed to allow system administrators and curious individuals to scan large networks to determine which hosts are up and what services they are offering." [3] Nmap has support for all of the major stealth scanning techniques, and adds new options as they are discovered. It also supports IP spoofing, decoying, fragmenting, and a number of very useful features. Nmap was written with the security auditor in mind to perform intrusion detection and port scanning detection tests. What is available to the system administrator is also available to the attacker, so a diligent administrator must be as aware of what Nmap can do for him/her, as aware of what it can

be used for by attackers. Beyond the usual scanning technology, Nmap can also be used to remotely identify a host's operating system. This is done through a technique called TCP/IP fingerprinting. The premise of this is that TCP/IP is a specification, but the implementation of it into a functioning system has differed slightly between software companies. When the data is correct, each implementation reacts the same, but when invalid data is sent to a TCP/IP stack, each implementation reacts a little differently. These small differences can be compared to other operating systems and a fingerprint is created. This is useful when scanning for specific services.

Hping2

Hping2 is the second version of a tool designed to take advantage of poor TCP sequence number generation in a number of TCP/IP implementations. There is a myriad of options that hping2 provides, but the main function is to scan for ports while concealing the identity of the attacker. The process involves pinging a 3rd party computer and monitoring the initial sequence number that was generated. A sequence in the TCP protocol will be maintained and lost packets are discovered. [11] Essentially this means that an attacker can scan a host and have it think that the scanning is originating from the 3rd party's computer.

Summary

There are a great many number of stealth scanning techniques available to the attacker and system administrator today. The number continues to rise as the security community continually aims to catch up to the latest method of stealth scanning and this trend shows no signs of stopping. The lesson learned is that security professionals must be as well equipped as possible to thwart attempted port scanning activities. They must be as current with emerging technologies as the people trying to attack their computers. Only through due diligence and constant learning can a security professional detect such complex activities such as port scanning. Security professionals must be prepared to combat stealth scanning attempts using any combination of techniques available to the attacker. It is through the combination of stealth technologies that detection of this activity becomes more difficult than one method alone. While software programs may be able to detect and stop stealth scanning activities to some degree, a knowledgeable and experienced security professional will be the biggest asset in dealing with stealth scanning activities the originate from both the internet, and within the local network itself.

Resources

[1] "Port Numbers." May 28, 2002.
URL: http://www.iana.org/assignments/port-numbers (May 30, 2002)

[2] Maimon, Uriel. "Port Scanning without the SYN flag." November 8, 1996.
URL: http://www.phrack.org/show.php?p=49&a=15 (May 30, 2002)

[3] Fyodor. "NMAP Network Security Scanner Man Page."
URL: http://www.insecure.org/nmap/nmap_manpage.html (May 29, 2002)

[4] "OSDEM Presentation - Network Reconnaissance Techniques"
URL: http://www.insecure.org/nmap/OSDEM_Presentation/ (May 30, 2002)

[5] "CERT® Incident Note IN-98.04" September 29, 1998.
URL: http://www.cert.org/incident_notes/IN-98.04.html (May 30, 2002)

[6] "RFC 793 – Transmission Control Protocol." September 1981.
URL: http://www.faqs.org/rfcs/rfc793.html (May 29, 2002)

[7] Natas. "In Regards to Secured Hosts."
http://www.8op.com/sec666/inregards.txt (June 10, 2002)

[8] "IP Spoofing." (2002)
URL: http://www.webopedia.com/TERM/I/IP_spoofing.html (May 30, 2002)

[9] Whalen, Sean. "An Introduction to Arp Spoofing." April 2001.
URL: http://packetstormsecurity.org/papers/protocols/intro_to_arp_spoofing.pdf (May 30, 2002)

[10] Mateti, Prabhaker. "Port Scanning." 2001.
URL: http://www.cs.wright.edu/~pmateti/Courses/499/Probing/ (June 3, 2002)

[11] "A New Stealth Port Scanning Method" December, 1998
URL:
http://www.securiteam.com/securitynews/A_new_stealth_port_scanning_method.html
(June 3, 2002)