Global Information Assurance Certification Paper

GIAC Firewall and Perimeter Protection Practical Assignment

Jerry Landry
Network Administrator
08/14/2000

Overview

The purpose of this document is to define the following for each service, protocol or vulnerability.

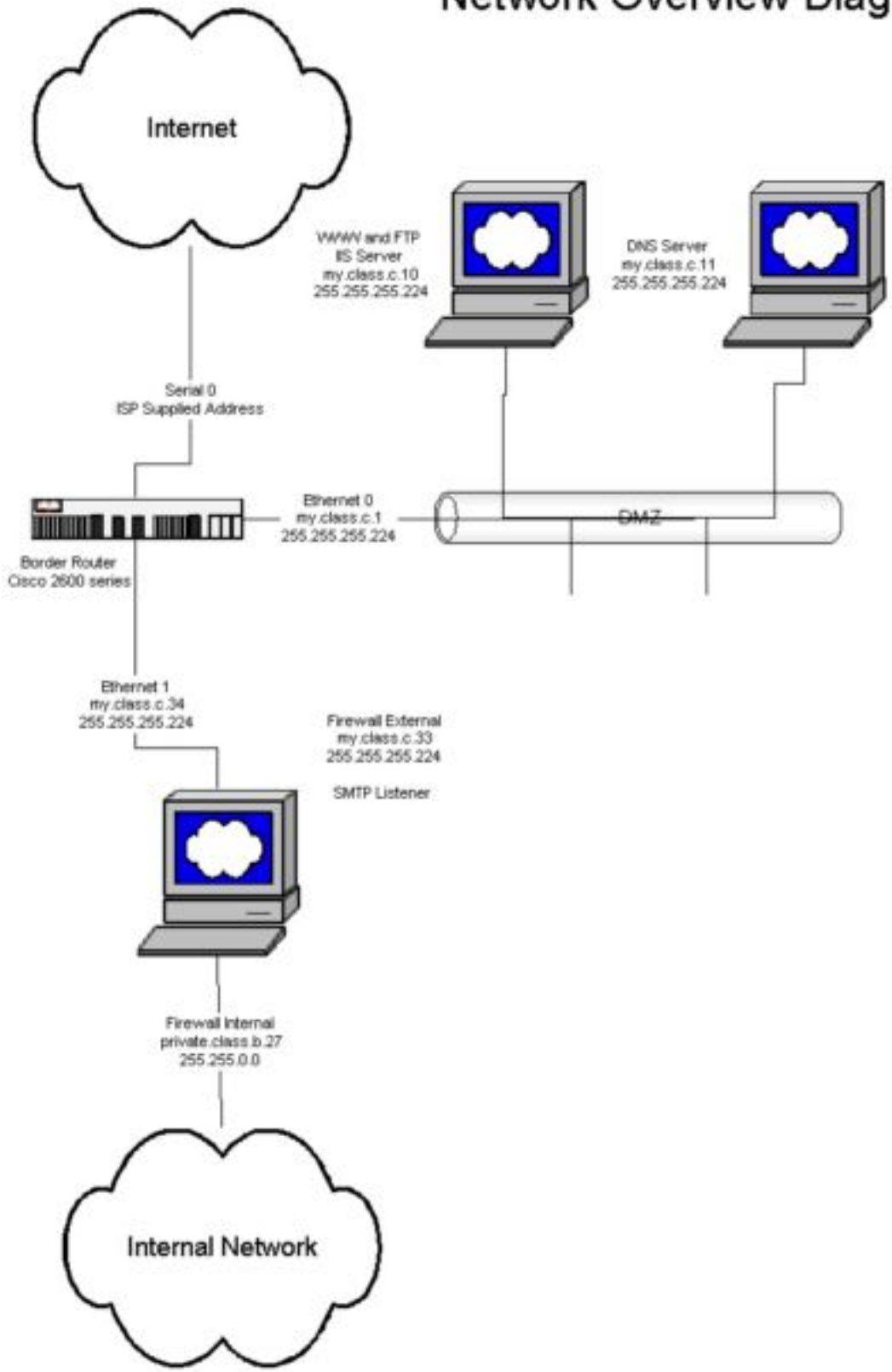
1. Define why each may be vulnerability.
2. Describe the behavior of the each on the network.
3. Syntax of the filter necessary to block / limit access to the protocol or service.

Our configuration was based on a Cisco 2621 with IOS version 12.1

At the end of this document, there will be a complete configuration for the border router.

As far as testing, we have decided to outsource the testing process. Given the number of vulnerabilities today, it would not be cost effective for us to keep up with all of the hacks and tools that are available on the internet. Instead of determining how to test each and every issue, we are going to contract with an auditing firm that specializes in network security. They will be contracted to test the top ten list items below, as well as any other known exploits. In addition, we will coordinate specific times where they will be allowed to attempt "true" denial of service exploits against our network.

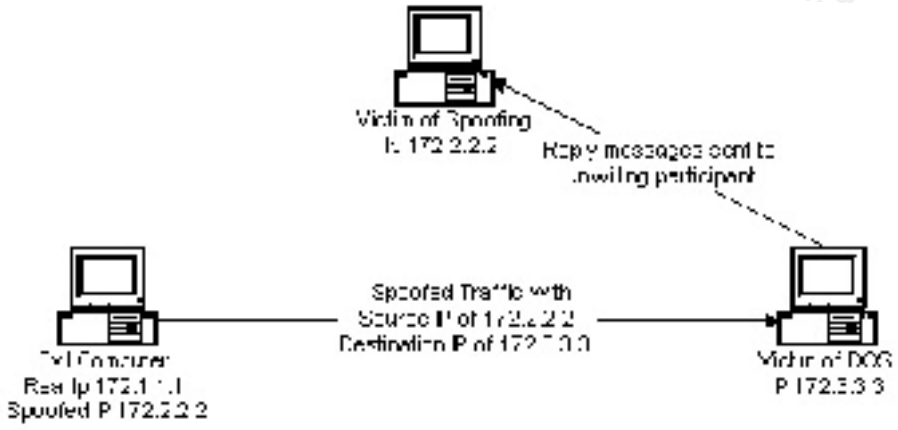
Network Overview Diagram



Spoofed Addresses / Source Routing

Spoofing

Spoofing is a process that inserts a bogus source IP address into an outbound IP packet. Doing this makes it extremely difficult to determine where the packet originated. This has become such a problem that the IETF has begun work on an "ICMP ITRACE(http://search.ietf.org/internet-drafts/draft-bellovin-itrace-00.txt)" which will assist in locating the source of spoof traffic.
A typical IP packet will consist of various fields but specifically a source address and a destination address. Under normal conditions, the source address will be the IP address of the machine initiating the network traffic and the destination address will be the IP address of the machine receiving the traffic. Typically, both are needed so that the appropriate communications handshaking and message transfers can take place. However, since spoofing is typically used for evil purposes, the person originating the traffic usually wants to stay anonymous. This is done by changing the source address, causing all replies from the victim machine to be sent to an unwilling participant.



We will allow outbound access using the following global access list on our Border Router.

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	ouboundfilter	Permit	Ip	Our.class.c.0	0.0.0.255	Any		
2	ouboundfilter	Deny	Ip	Any		Any		log

Rule #1 permits only IP traffic that has our private class c address as the source address from leaving our router.
Rule #2 denies all other traffic from leaving our network.

Access list 115 will be applied to the inbound side of interfaces Ethernet0 and Ethernet1.

We will also block inbound access to any source address that is in the Private & Reserved Address range using the following global access list on our Border Router. This list is based on recommendations from various sources including http://www.sans.org/dosstep/cisco_spoof.htm and http://search.ietf.org/internet-drafts/draft-manning-dsua-03.txt.

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	deny	icmp	any		any		redirect
2	inboundfilter	deny	ip	0.0.0.0	0.255.255.255	any		
3	inboundfilter	deny	ip	10.0.0.0	0.255.255.255	any		
4	inboundfilter	deny	ip	127.0.0.0	0.255.255.255	any		
5	inboundfilter	deny	ip	169.254.0.0	0.0.255.255	any		
6	inboundfilter	deny	ip	172.16.0.0	0.15.255.255	any		

7	inboundfilter	deny	ip	192.0.2.0	0.0.0.255	any		
8	inboundfilter	deny	ip	224.0.0.0	15.255.255.255	any		
9	inboundfilter	deny	ip	240.0.0.0	15.255.255.255	any		
10	inboundfilter	deny	ip	serial.interface.ip	0.0.0.0	serial.interface.ip	0.0.0.0	
11	inboundfilter	deny	ip	my.class.c.0	0.0.0.255	any		

Rule #1 prevents any ICMP redirect packets from entering our network.
Rule #2 denies access to any packet with a source address equal to the "special use" address space.
Rule #3 denies access to any packet with a source address equal to the private class "A" address space.
Rule #4 denies access to any packet with a source address equal to the loop-back address space.
Rule #5 denies access to any packet with a source address equal to end node auto-configuration address space.
Rule #6 denies access to any packet with a source address equal to the private class "B" address space.
Rule #7 denies access to any packet with a source address equal to the TEST-NET address space.
Rule #8 denies access to any packet with a source address equal to the multicast address space.
Rule #9 denies access to any packet with a source address equal to the IETF Reserved address space.
Rule #10 prevents a "Land Attack" which would basically could cause a router interface to establish a TCP connection with itself and get into an infinite loop
Rule #11 denies access to any packet with a source address equal to our private class "C" address space

Source Routing

Source routing is a method of crafting a packet so that it goes through certain routers in an attempt to avoid other routers. This is typically done because a certain router may be blocking something that the initiator is trying to pass. In today's current internet environment, there is no legitimate reason that anyone would need to dictate what route a packet should take. Since we only route to and from our on private network, we should never get packets that route somewhere else via our router. Because of this, we will prevent all source routing on our router.

no ip source-route

Login Services

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	permit	tcp	any		my.class.c.10	0.0.0.0	eq 21
2	inboundfilter	permit	tcp	any		my.class.c.10	0.0.0.0	eq 20
3	inboundfilter	deny	tcp	any		any		range 20 23
4	inboundfilter	deny	tcp	any		any		range 512 514

Telnet (23/tcp)

Telnet is a terminal emulation program that is typically used for configuring remote hosts. Due to the security concerns with remote telnet over the Internet, we will disable all incoming telnet access. We will allow telnet access to routers and other devices from our private and public address space.

Rule #3 will prevent Internet access to telnet on our network.

SSH (22/tcp)

SSH (Secure Shell) uses encryption such as DES (Data Encryption Standard) and RSA to provide a secure method for accessing remote computers. It does require a SSH server running on the remote computer as well as a SSH client. Since

we don't have any SSH servers we will also disable this service from the internet. The other concern with SSH is that a hacker may hack into a box, install a SSH server and then have encrypted access to that box. That would make it impossible for an intrusion detection system to decipher the encrypted traffic.

Rule #3 prevents the Internet access to SSH on our network.

FTP (21/tcp)

FTP (File Transfer Protocol) is used to transfer information between machines. It has a number of uses but is used primarily to down load product support information and documentation and primarily for the maintenance and upkeep of web pages. It is also the odd ball of the TCP protocol. What makes it odd is that it uses 2 ports. It uses port 21 for control and port 20 for transferring data.

Rules #1 & #2 are used to allow Internet access to our public FTP server.
Rules #3 are used to prevent Internet ftp access on our network.

Rlogin et al (512/tcp – 514/tcp)

Rlogin is similar in nature to telnet, but was created by the Berkeley BSD UNIX community. Because of it's use of trusted hosts and users, rlogin is considered a vulnerable protocol. Again, remote login access to all systems are refused including rlogin.

Rules #4 is used to prevent Internet access to Rlogin on our network.

RPC and NFS

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	deny	tcp	any		any		eq 111
2	inboundfilter	deny	udp	any		any		eq 111
3	inboundfilter	deny	tcp	any		any		eq 2049
4	inboundfilter	deny	udp	any		any		eq 2049
5	inboundfilter	deny	tcp	any		any		eq 4045
6	inboundfilter	deny	udp	any		any		eq 4045

Portmap/rpcbind (111/tcp and 111/udp)

rpcbind (newer version of portmap) is a service that allows RPC services to register their program number, version and protocol. Once registered, that information is available via an rpcinfo –p command. Obviously no system administrator would want to readily give out what RPC services are running on a given machine. Because of that, the rpcbind ports are blocked from the internet.

Rules #1 - #2 will deny the Internet access to port 111 (tcp and udp)

NFS (2049/tcp and 2049/udp)

NFS (Network File System) allows computer to share resources over any TCP/IP network including the Internet. It works at the OSI application layer. It works in conjunction with XDR (External Data Representation) on the presentation layer and RPC (Remote Procedure Call) on the session layer to communicate with TCP/IP. Since the primary purpose of NFS is to share resources, it is obvious why it is a security concern. Since we don't allow NFS access to any server via the internet, we will completely disable this.

Rules #3 - #4 deny all Internet access to the NFS ports of 2049(udp and tcp).

Lockd (4045/tcp and 4045/udp)

LOCKD is used along with STATD (file locking daemon) to provide NFS with crash recovery capabilities. If NFS is not being used, it is highly recommended that LOCKD and STATD be disabled. One example of the vulnerablities of LOCKD can be found at http://www.securiteam.com/unixfocus/Linux_rpc_lockd_vulnerable_to_remote_DoS.html

Rules #5 - #6 deny all Internet access to the LOCKD ports of 4045 (udp and tcp).

NetBIOS in Windows NT

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	deny	tcp	any		any		eq 135
2	inboundfilter	deny	udp	any		any		eq 135
3	inboundfilter	deny	udp	any		any		range 137 138
4	inboundfilter	deny	tcp	any		any		eq 139
5	inboundfilter	deny	tcp	any		any		eq 445
6	inboundfilter	deny	udp	any		any		eq 445

Windows 135(tcp/udp), 137(udp), 138(udp), 139(tcp) plus 445(tcp/udp) for Windows 2000

The ports listed above are the basis of the NETBIOS (Network Basic Input Output System) API (Application Programming Interface) in Windows. This API used to separate the application and transport layers in the Microsoft TCP/IP protocol stack. This allows a windows programmer to write a network-enabled program without knowing or understanding the TCP/IP protocol. While ease of use is makes life easy for a programmer, it makes a system quite vulnerable from a security administrators standpoint. For example, all windows systems come with a tool called NBTSTAT. This tool will allow another machine to determine the remote computers host name, whether or not it belongs to a domain, the domain name, and the IP address of any connected machines and their host name. All of this information can be quite useful to a hacker. In addition to NBTSTAT, windows comes with a NET utility. This utility allows a remote user to mount any available share on the remote computer, assuming you know the password for the share. A good example of exploiting these features was the "911 Virus". It would do scan thru various IP addresses using a nbtstat –a command. If it received a response, it would try using the net use command to mount the c drive with no password. If successful, it would install itself onto the remote computer.

Another recent vulnerability of the NETBIOS protocol is explained in detail in a Microsoft Technet document http://www.microsoft.com/technet/security/bulletin/fq00-047.asp. What it amounts to is that a virus could be crafted which uses the nbtstat –a command, just like "911", but instead of trying to mount the c drive, the virus would issue a name conflict datagram to the machine. Although the name is not in conflict, it would stop answering to that name.

Rule #1 and #2 denies Internet access to port 135
Rule #3 denies Internet access to port 137 and 138 on our network.
Rule #4 denies Internet access to port 139 on our network
Rule #5 denies Internet access to port 445 on our network.

Explanation of Microsoft port usage:

http://support.microsoft.com/support/kb/articles/q150/5/43.asp NT 4.0
http://www.microsoft.com/windows2000/library/resources/reskit/samplechapters/cnfc/cnfc_por_simw.asp Windows 2000

X Windows

Rule	Reflexive	Permit /	Protocol	Source Address	Source Mask	Destination	Destination Mask	Options

#	access-list	Deny				Address		
	inboundfilter	deny	tcp	any		any		range 6000 6255

6000/tcp thru 6255/tcp

Xwindows is a service, which allows remote graphical access to between UNIX and NT. One of the main problems with Xwindows is that several of the packages install with the x sessions available to anyone on the network by default. One web page, http://www.ducktank.net/tips/X.html, outlines how easy it is to exploit the problems with xwindows.

Rule #1 denies Internet access to all of the x windows ports.

Naming Services

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	permit	udp	any		my.class.c.11	0.0.0.0	eq 53
2	inboundfilter	permit	tcp	sec.dns.server.a	0.0.0.0	my.class.c.11	0.0.0.0	eq 53
3	inboundfilter	permit	tcp	sec.dns.server.b	0.0.0.0	my.class.c.11	0.0.0.0	eq 53
4	inboundfilter	deny	ip	any		my.class.c.11	0.0.0.0	
5	inboundfilter	deny	tcp	any		any		eq 53
6	inboundfilter	deny	udp	any		any		eq 53
7	inboundfilter	deny	tcp	any		any		eq 389
8	inboundfilter	deny	udp	any		any		eq 389

DNS (53/udp) to all machines which are not DNS servers

DNS (Domain Name System) is a service which allows computers to query it's database and convert host names into IP addresses. Since most communications require a destination IP address, this conversion is a requirement. For example, whenever someone tries to access www.sans.org from a browser, their machine does a DNS query to determine that 167.216.133.33 is the ip address of sans. Once the IP address is obtained, the browser can make a connection and download the page. With the number of domains currently in use, it would be impossible for any single organization to manage all of the Name to IP entries in the DNS database. Because of this, each domain has a machine designated as the primary. A single machine can be primary for multiple domains, but multiple machines cannot be primary for a single domain. In the lookup described earlier, the browser typically only knows about a single DNS server and tries it first. If that server did not know about the sans.org domain, it would contact a higher level server on behalf of the browser. The higher level server (probably a root server) would respond with the ip address of the DNS server who is primary for the domain in question. The local DNS server would then contact the primary DNS server for the IP address. What this means is that if someone is going to use one of the machines that is listed in your DNS tables, it needs to be accessible by everyone on the internet. In the DNS query explanation above, the UDP protocol was used. Under rare circumstances, the TCP protocol would be needed if the answer to the query were too large. Since that would never be the case on our DNS server, we have decided to prevent TCP access to the DNS server.

Rule #1 allows everyone on the Internet to access our DNS server
Rule #4 prevents any further access to the DNS server.

DNS (53/tcp) except from any external secondary DNS Server

The only acceptation to the "no TCP access to our DNS server" would be for Zone Transfer access to our Secondary DNS servers. Under a normal DNS configuration you would have at least one primary and at least one secondary. Whenever you have a secondary DNS server, it must be allowed to do Zone Transfers with the primary. The Zone Transfer moves all records from the primary DNS server to the secondary DNS server. Since a Zone Transfer provides a complete dump of a

DNS database, we want to limit capability to known, trusted machines. Ultimately, a persistent hacker will determine which machines are on a given network, but you don't want to make it a simple as allowing the hacker to do a Zone Transfer.

Rules #2 - #3 are used to limit TCP access to our DNS to our 2 secondary DNS servers.
Rule #4 prevents any further access to the DNS server.
Rules #5 - #6 prevent access to any other addresses on port 53

LDAP (389/tcp and 389/udp)

LDAP (Lightweight Directory Access Protocol) is a directory service protocol that standardizes the method of storing extracting data in a hierarchical structure. The hierarchical structure stores objects, which could include items such as people, servers, services, etc. Windows 2000, Exchange and Firewall-1 are just a few of the numerous packages that LDAP as part of their base package. Because of the vast amount of information available via LDAP, ensuring that only trusted systems have access to it is of the utmost importance.

Rules #7 - #8 will deny Internet traffic going to any LDAP services.

Mail

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	permit	tcp	any		my.class.c.33	0.0.0.0	eq 25
2	inboundfilter	deny	tcp	any		any		eq 25
3	inboundfilter	deny	tcp	any		any		eq 109
4	inboundfilter	deny	tcp	any		any		eq 110
5	inboundfilter	deny	tcp	any		any		eq 143

SMTP(25/tcp) to all machines, which are not external mail relays

SMTP (Simple Mail Transport Protocol) is the protocol that is used for transferring email over the internet.

Rule #1 permits access SMTP access to our external mail relay.
Rule #2 denies access to SMTP on any other machine on our external network.

Our firewall further refines which domains are accepted. This is accomplished by adding the following lines to the configuration file. If the domain is not defined as a local domain and permit-relay is not setup for that domain, the mail is rejected by the smap

smap,smapd: permit-hosts *
smap: local-domain domain1.com domain2.com domain3.com domain4.com
smap: permit-relay *.domain1.com
smap: permit-relay *.domain2.com
smap: permit-relay *.domain3.com
smap: permit-relay *.domain4.com

POP(109/tcp and 110/tcp)

POP (Post Office Protocol) allows a client application to retrieve mail from the MTA (Messages Transfer Agent or email server). The current version of POP is POP3. Typically, POP3 is a receive only protocol, which means that it must download the email from the server and then delete the email on the server. There are however, versions that allow the client to leave the messages on the server. Whenever messages are sent to the MTA, SMTP is used. In addition, POP3 only allows support for one mailbox.

Rule #3 will deny Internet traffic going to port 109 (tcp)
Rule #4 will deny Internet traffic going to port 110 (tcp)

Since we do not support this service on any machine in our DMZ, it is completely blocked.

IMAP(143/tcp)

IMAP (Internet Mail Access Protocol) was created due to the limitations of POP. It increased the authentication, supports multiple mailboxes, and supports various modes of operation such as online, offline and disconnected. One example of an IMAP exploit and fix can be found at http://www.microsoft.com/technet/security/bulletin/ms00-001.asp.

Rule #5 will deny Internet traffic going to port 143(tcp).

Since we do not support this service on any machine in our DMZ, it is completely blocked.

Web

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	permit	tcp	any		my.class.c.10	0.0.0.0	eq 80
2	inboundfilter	permit	tcp	any		my.class.c.10	0.0.0.0	eq 443
3	inboundfilter	deny	ip	any		my.class.c.10	0.0.0.0	
4	inboundfilter	deny	tcp	any		any		eq 80
5	inboundfilter	deny	tcp	any		any		eq 443
6	inboundfilter	deny	tcp	any		any		eq 8000
7	inboundfilter	deny	tcp	any		any		eq 8080
8	inboundfilter	deny	tcp	any		any		eq 8888

HTTP (80/tcp) and SSL (443/tcp) except to external web servers, also block common high order ports (8000/tcp, 8080/tcp, 8888/tcp, etc.)

The default port for HTTP, Hypertext Transfer Protocol is port 80 and the default port for SSL, Secure Sockets Layer, is port 443. HTTP is typically used for web browsing, and SSL is used a secure and encrypted connection is needed. Unless you have a web server setup and running, you typically don't want to allow HTTP and SSL traffic. This is becoming more and more important since most application providers making there administrative tools accessible via a web browser. This means that the boxing is "listening" for traffic on port 80. By disallowing port 80 traffic to all servers except for the web server, we are ensuring that any unknown features that may have been accidentally left on, will not be accessible from the internet.

Rule #1 permits the Internet to access to port 80 (HTTP) on our web server.
Rule #2 permits the Internet to access to port 443 (SSL) on our web server.
Rule #3 denies all other traffic from the Internet to our web server.
Rules #4 - #8 will deny Internet traffic going to ports 80, 443, 8000, 8080, 8888 anywhere on our class "C" address space.

As a note, having the eq (port #) after the destination causes the rule to filter only on the destination address.

Small Services

Ports below 20/tcp and 20/udp

Ports below 20 are typically used for services such as Echo, Discard, Chargen and Daytime. These services all provide bits of information that could be of use to a hacker, but are rarely used for anything anymore.

no service tcp-small-servers
no service udp-small-servers

Time (37/tcp and 37/udp)

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
	inboundfilter	deny	tcp	any		any		eq 37
	inboundfilter	deny	udp	any		any		eq 37

Miscellaneous

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	deny	udp	any		any		eq 69
2	inboundfilter	deny	tcp	any		any		eq 79
3	inboundfilter	deny	tcp	any		any		eq 119
4	inboundfilter	deny	tcp	any		any		eq 123
5	inboundfilter	deny	tcp	any		any		eq 515
6	inboundfilter	deny	udp	any		any		eq 514
7	inboundfilter	deny	tcp	any		any		range 161 162
8	inboundfilter	deny	udp	any		any		range 161 162
9	inboundfilter	deny	tcp	any		any		eq 179
10	inboundfilter	deny	tcp	any		any		eq 1080

TFTP (69/udp)

TFTP (Trivial File Transfer Protocol) performs the same function as FTP, but lacks any means of authentication. TFTP is also the method of transferring files to and from routers. Since it is unauthenticated and is used for routers, it should be restricted as much as possible.

Rule #1 will deny Internet traffic going to port 69 (udp)

Finger (79/tcp)

Finger is both a protocol and a program. The program is used to determine the status of a host, such as who is logged on and to determine information about a user, such as email address. Since this is not the type of information that should be given out, access to this port should be disabled.

Rule #2 will deny Internet traffic going to port 79 (tcp)

NNTP (119/tcp)

NNTP (Network News Transfer Protocol) is the most common method of transferring Usenet article data. Because of the current breed of viruses such as the Kak virus and the fact that common virus scanners do not work with news readers, we will disable all incoming news traffic. The other issue with NNTP is the shear volume of traffic. Our network provider states:
"*Usenet:*
If you are interested in a Usenet news-feed, inform your engineer and you will receive a copy of the current NNTP News Guide (or UUCP News Guide if applicable). If you have ordered a 56K or 64K circuit, you will not be able to receive a "full" newsfeed because it would exceed your available bandwidth. If you have ordered a T1 or E1 circuit, a "full" newsfeed may exceed your available bandwidth, depending on what other traffic is on your line. Please select specific newsgroups that you wish to receive."

Rule #3 will deny Internet traffic going to port 119 (tcp).

NTP (123/tcp)

NTP (Network Time Protocol) is used to synchronize machines to a common time. This is especially important when trying to investigate network security issues. However, as with any other type of IP traffic, NTP traffic can be spoofed. Spoofed NTP traffic could be used to change the time on a system prior to attempting to break in so that it's logs could not be compared to other logs.

Rule #4 will deny Internet traffic going to port 123 (tcp)

LPD (515/tcp)

LPD is used to provide network printing services. While sounding relatively harmless, it has suffered buffer overflow problems similar to just about every other service that uses the TCP/IP stack. One example of this can be found at http://www.microsoft.com/technet/security/bulletin/ms00-021.asp.

Rule #4 will deny Internet traffic going to port 515 (tcp)

Syslog (514/udp)

Syslog is a service that allows multiple machines to log system messages to a central host. While this makes it extremely convenient for a system administrator to have a single location to review logs. One concern is that since most syslog servers accept unauthenticated traffic, it would be easy for a hacker to add numerous bogus entries. This could potentially cause a system administrator to try and track down something that is not real in the first place.

Rule #4 will deny Internet traffic going to port 123 (tcp)

SNMP (161/tcp and 161/udp, 162/tcp and 162/udp)

SNMP (Simple Network Management Protocol) is used to read and write to a network device's MIB (Management Information Base. SNMP is primarily used for monitoring of a network. Once one of these packages are installed, you tell it what subnets to monitor and the package will go out and discover the devices that have SNMP enabled. The discovery process works by trying a default community string such as public. The community string is the primary means of authentication used by the SNMP service. Once it performs the discovery process, the software can display and alarm on any information that is contained in that device's MIB.

The big security concern with SNMP is the community string, which is the primary means of authentication. Since it is typically left with the default setting of public, it is an easy way for hackers to investigate and change settings on an SNMP enabled device.

Rules #7 and #8 will deny Internet traffic going to the SNMP ports 161 and 162 (udp and tcp).

BGP (179/tcp)

BGP (Border Gateway Protocol) is used by the backbone of the internet to aggregate route information. Since our perimeter router is not connected to the backbone of the Internet, it should not need to receive BGP packets.

Rule #9 will deny Internet traffic going to port 179(tcp)

Socks (1080/tcp)

Rule #4 will deny Internet traffic going to port 1080 (tcp)

ICMP

Rule #	Reflexive access-list	Permit / Deny	Protocol	Source Address	Source Mask	Destination Address	Destination Mask	Options
1	inboundfilter	deny	icmp	any		any		echo
2	ouboundfilter	deny	icmp	any		any		echo-reply
3	ouboundfilter	deny	icmp	any		any		time-exceed
4	ouboundfilter	deny	icmp	any		any		dest-unrch

Incoming Echo Request (ping and windows tracert)

Rule #1 will deny access to any ICMP packet that is trying to perform an echo reply. An echo / echo-reply is the easiest way to map a network.

The ping command uses the ICMP packet to determine if the host is listening. It also calculates the amount of time it takes to receive the reply. Since you have the ability to determine the time it takes for a packet to get to a host, it is a common tool used in troubleshooting network problems. Because of that, you must compare the risk of allowing the traffic against the loss of troubleshooting ability before disabling this traffic.

Windows tracert uses the ICMP echo, but in a slightly different way. It sends the ICMP packet with to the destination with a time to live equal to 1. Since each router decrements this value by 1 when it receives the packet, it will time out at the first router. The first router will reply that it timed out. The windows client will send this same the packet 2 more times with a time to live equal to 1 and record the amount of time each packet takes. The windows client will then increment the time to live by 1 and repeat the process. The incrementing by one will get the packet past the first router, but will time out at the second router. This process is repeated until the final destination is reached.

Outgoing Echo Replies, Time Exceeded, and Unreachable Messages

In addition to adding an access list to the router interface to disable unreachable ICMP traffic, we will add "no ip unreachables" entry to the router, which is another way of preventing unreachable messages.

Because of the uncertainty involved with today's applications we will make sure that ICMP traffic does not leave our network. This is especially true with the UDP protocol. Since it does not have a reply mechanism, some systems will reply to UDP traffic with ICMP if problems are encountered. Because of these two unknowns, we will disable any ICMP traffic

that could be used to map our network. Echo replies, time exceeded and unreachable messages are all types that could be used to do this mapping.

Rule #2 will deny outbound echo-replies.
Rule #3 will deny outbound time exceeded.
Rule #3 will deny outbound unreachable messages.

Authenticated Return Traffic

All of the rules above filter out specific ports that are known to be of a hostile nature. Any remaining traffic is either additional traffic that was not covered in this list or traffic is a reply to traffic that actually originated from within our network. The configuration below uses reflexive access-lists to track the outbound traffic for authentication in the inbound direction. We could have simplified the inbound access-list by entering only the permits and then the reflexive access-list evaluate, but we wanted to ensure that the ports listed above as deny got rejected **no matter what**.

The outboundfilter reflexive list uses a permit line with a reflect option. The reflect option builds a pseudo state table (iptraffic) for all outbound traffic. Then the inboundfilter drops all unwanted traffic, allows all necessary inbound traffic and then validates what is left against the iptraffic table. If there is a match, the traffic is allowed to pass, otherwise, it is rejeted.

Sample Configuration

Sample Cisco Configuration - The specifics of the interfaces would still need to be setup and all routing entries need to be setup

```
!
! *** You must be in global configuration mode to begin entering these configurations.  ***
! *** do this by entering config term in enable mode
!
no cdp run
no service finger
no service udp-small-servers
no service tcp-small-servers
no ip source-route
no ip bootp server
no ip http server
no ntp master
no logging console
ip refective-list timeout 120
!
! ##### Start Serial 0 outbound Access List
!
!  *** After entering the next line you will be in access-list mode
!
ip access-list extended outboundfilter
deny icmp any  any  echo-reply
deny icmp any  any  time-exceed
deny icmp any  any  dest-unrch
permit tcp our.class.c.0 0.0.0.255 any  reflect  iptraffic
permit udp our.class.c.0 0.0.0.255 any  reflect  iptraffic
permit icmp our.class.c.0 0.0.0.255 any  reflect  iptraffic
deny ip any   any  log
!
!  *** Enter “end” to get out of access-list mode
!
! ##### Start Serial 0 inbound Access List
!
```

```
! !  *** After entering the next line you will be in access-list mode
!
ip access-list extended inboundfilter
!
!   *****Prevent ICMP Echo requests******
!
deny icmp any  any  echo
deny icmp any   any  redirect
!
!   *****Prevent Spoofing******
!
deny ip 0.0.0.0 0.255.255.255 any
deny ip 10.0.0.0 0.255.255.255 any
deny ip 127.0.0.0 0.255.255.255 any
deny ip 169.254.0.0 0.0.255.255 any
deny ip 172.16.0.0 0.15.255.255 any
deny ip 192.0.2.0 0.0.0.255  any
deny ip 224.0.0.0 15.255.255.255 any
deny ip 240.0.0.0 15.255.255.255 any
!
!   *****Prevent "Land Attack" Another type of spoofing******
!
deny ip serial.interface.ip 0.0.0.0 serial.interface.ip 0.0.0.0
!
!   *****Prevent inbound RLOGIN******
!
deny tcp any  any  range 512 514
!
!   *****Prevent inbound portmap/rpcbind******
!
deny tcp any  any  eq 111
deny udp any  any  eq 111
!
!   *****Prevent inbound NFS******
!
deny tcp any  any  eq 2049
deny udp any  any  eq 2049
!
!   *****Prevent inbound lockd******
!
deny tcp any  any  eq 4045
deny udp any  any  eq 4045
!
!   *****Prevent all inbound windows sharing services******
!
deny tcp any  any  eq 135
deny udp any  any  eq 135
deny udp any  any  range 137 138
deny tcp any  any  eq 139
deny tcp any  any  eq 445
deny udp any  any  eq 445
!
!   *****Prevent all inbound Xwindows******
!
deny tcp any  any  range 6000 6255
!
!   *****Allow DNS queries to our DNS server and Zone transfers with our secondary server******
!
permit udp any  my.class.c.11 0.0.0.0 eq 53
permit tcp sec.dns.server.a 0.0.0.0 my.class.c.11 0.0.0.0 eq 53
permit tcp sec.dns.server.b 0.0.0.0 my.class.c.11 0.0.0.0 eq 53
!
```

```
!   *****Prevent all other access to our DNS Server******
!
deny ip any  my.class.c.11 0.0.0.0
!
!   *****Prevent all other inbound DNS******
!
deny tcp any  any  eq 53
deny udp any  any  eq 53
!
!   *****Prevent Inbound LDAP******
!
deny tcp any  any  eq 389
deny udp any  any  eq 389
!
!   *****Allow inbound mail connections to our external mail relay ******
!
permit tcp any  my.class.c.33 0.0.0.0 eq 25
!
!   *****Prevent all other inbound mail services******
!
deny tcp any  any  eq 25
deny tcp any  any  eq 109
deny tcp any  any  eq 110
deny tcp any  any  eq 143
!
!Setup allowed ports on WWW and FTP Server
!
permit tcp any  my.class.c.10 0.0.0.0 eq 21
permit tcp any  my.class.c.10 0.0.0.0 eq 20
permit tcp any  my.class.c.10 0.0.0.0 eq 80
permit tcp any  my.class.c.10 0.0.0.0 eq 443
deny ip any  my.class.c.10 0.0.0.0
!
!   *****Prevent all other inbound telnet SSH and FTP******
!
deny tcp any  any  range 20 23
!
!   *****Prevent all other inbound web protocols******
!
deny tcp any  any  eq 80
deny tcp any  any  eq 443
deny tcp any  any  eq 8000
deny tcp any  any  eq 8080
deny tcp any  any  eq 8888
!
!   *****Prevent TIME, TFTP and FINGER******
!
deny tcp any  any  eq 37
deny udp any  any  eq 37
deny udp any  any  eq 69
deny tcp any  any  eq 79
!
!   *****Prevent NNTP and NTP******
!
deny tcp any  any  eq 119
deny tcp any  any  eq 123
!
!   *****Prevent LPD and SYSLOG******
!
deny tcp any  any  eq 515
deny udp any  any  eq 514
!
```

```
!   *****Prevent Inbound SNMP ******
!
deny tcp any  any  range 161 162
deny udp any  any  range 161 162
!
!   *****Prevent BGP and SOCKS******
!
deny tcp any  any  eq 179
deny tcp any  any  eq 1080
!
!   *****Determine if traffic the return packet for of previous authenticated outbound traffic******
!
evaluate iptraffic
!
!
!  *** Enter “end” to get out of access-list mode
!
!
!  *** Enter “interface serial 0” to enter config mode for serial 0 interface
!
interface serial 0
ip address isp.assigned isp.mask.assigned
ip access-group inboundfilter in
ip access-group outboundfilter out
!
!
!
!  *** Enter “end” to get out of interface mode
!
!
!  *** Enter “interface ethernet 0” to enter config mode for ethernet 0 interface
!
interface ethernet 0
ip address my.class.c.1 255.255.255.224
!
!
!  *** Enter “end” to get out of interface mode
!
!
!  *** Enter “interface ethernet 1” to enter config mode for ethernet 1 interface
!
interface ethernet 1
ip address my.class.c.34 255.255.255.224
```